



2021 Edition

Michał & Diana Malewicz

Designing User Interfaces

Learn User Interface Design, get your dream job and become a high-valued, successful designer



Table of Contents

INTRO

1. Introduction	3	4. Design and Art	21
2. Who is this book for?	7	5. What makes UI good?	26
3. UI vs UX	14	6. Perception	34

DESIGN BASICS

7. Screens	47	16. Tables	213
8. Layout and grid	55	17. Forms	229
9. Objects	77	18. Modals, popups	261
10. Colors	94	19. Navigation	280
11. Gradients	125	20. Animation	299
12. Typography	139	21. Photos	311
13. Icons	167	22. Illustration	323
14. Buttons	178	23. Language	329
15. Cards	203		

BONUS CHAPTERS

24. Design styles	337	29. Prototyping	404
25. Design process	351	30. Show your work	415
26. Design systems	358	31. Starting a design	426
27. UI Audit	373	32. Success in design	447
28. Developer handoff	394	33. Summary	465

Introduction

1.

Introduction



The license for this book covers one person only.

DO NOT redistribute or share this PDF with anyone, as it will lead to your license being void (and updates suspended).

Introduction

Hello! Michał and Diana here.

Thank you so much for purchasing “Designing User Interfaces”!

The idea for this book came to life in December of 2018. While we were teaching UI design classes, some of our students have asked us, what's the book with all those PDF materials we've been sending them one by one after class. Of course, at that time, they were simply things we put together to make it easier for the students - nothing more.

We looked around and noticed that a lot of the books on UI are out of date, short, vague, and often contradictory. We decided to change this and write this book. Nearly all of the knowledge within these pages is based on over 500 design projects we completed in the last ten years.

Now we are sharing all that experience with you!



Michał Malewicz



Diana Malewicz

HOW DO WE KNOW ALL THIS?



Starting in 1999, we have designed hundreds of digital products for all kinds of devices. That doesn't mean we're all-knowing. We're continually learning, teaching (which is also a great learning exercise), and testing new methods for building beautiful, usable digital products.

This book is a summary of our knowledge and our experiences. We don't assume that what we say here is the only right way to do things. It's the way we do it.

We want this book to grow and expand over time. That's why we're open to your suggestions, feedback, and ideas. If you think something is missing, needs an update or an additional explanation - reach out! If we use your feedback, we'll add you to the collaborator list at the end of this volume.

Knowledge is there to share it. While writing this ebook, and talking to our friends, co-workers, clients, students, and subscribers, we found out what are the most common problems people encounter while trying to design digital products. Our main goal was to give you all the answers to the most common questions, so you don't have to repeat the same mistakes we have.

Enjoy!



We build design-driven software that delivers results.

In 2013 we co-founded HYPE4 - a design-driven software house. Since then, we've been working for both small startups and Fortune-500 companies from all over the world. We were lucky enough to be able to work on nearly every product category (from banking, finance, advertising, and media to pregnancy tracking).

We started as a small, design-only studio with three employees. We did the designs, and then external software houses handled the development. We quickly noticed a huge problem with that approach. The execution quality was so low that we were rarely satisfied. In some cases, the designs were butchered beyond recognition. We decided to take action by starting the development branch of the company but strictly following our design-driven approach.

All of our developers understand the design principles, and our designers know a bit how to code. That allows us to achieve a much higher front-end development precision and overall quality. That has since become our defining feature.

At HYPE4, we build web and mobile products of all sizes. After hours we also work on our internal startups and have a lot of fun doing it. If you'd like to learn more, go to:

www.hype4.com



2.

**Who is this
book for?**

Who is this book for?

HOW TO DESIGN GREAT THINGS?

Each day we use a countless array of digital products. Some are obvious - like mobile apps and websites. But there are interfaces everywhere around us now. You interact with ticket machines, payment terminals, parking meters, alarm clocks, your TV, or your gaming console.

Most of these products are a result of the work of large groups of people. Multiple skillsets merge and intersect to create amazing experiences: designers, marketing ninjas, project managers, developers, and an occasional genius CEO.

This book will show you the building blocks and good practices, so you can spot them in the products you use, and improve the products you make.



“

**15 years of experience,
and over 500 projects
distilled into the best
design practices.”**

DESIGNERS

Designers will gain knowledge on how to make better looking and more usable products.

It took us years to learn some of these truths the hard way. We've condensed and simplified all that knowledge, so you don't have to repeat our mistakes. That will let you grow faster and be better. Good practices, quick-and-dirty safe hacks, and intentional rule-bending will allow you to bump up readability and aesthetics of what you design.

Beginners will gain solid ground to improve from, but even experts can find this book useful.

We worked on over 500 different products from almost every possible category. From fintech portals, all the way to pregnancy trackers. The problems we've encountered and the solutions are all within these pages.

Read on!



FRONT-END DEVELOPERS

Front-end developers will learn the basics of design and best practices on working with designers to speed up and raise the quality of their implementation work. We believe that every designer should understand a bit of coding, and every developer should know what design is all about.

A conscious developer gains another level of understanding of what he's building, which can quickly bring his work to a whole other level.

This design-fluency is true, especially for junior developers, as they can grow without repeating the mistakes of others.

margin-left: 44px;
color: #223;

Details My Cart **12**

Shifty Boa White 2019

\$169,55

ThirtyTwo Women

Size **37** Color **White**

Focused around comfort and flexibility, ThirtyTwo's signature Shifty Boa White Snowboard Boots are back this season. The Shifty comes complete with a Boa Lacing System providing a quick and efficient closure that enables easy on-the-fly adjustability.

background: #EF3030;
border-radius: 25px;

letter-spacing: 0;
line-height: 19px;

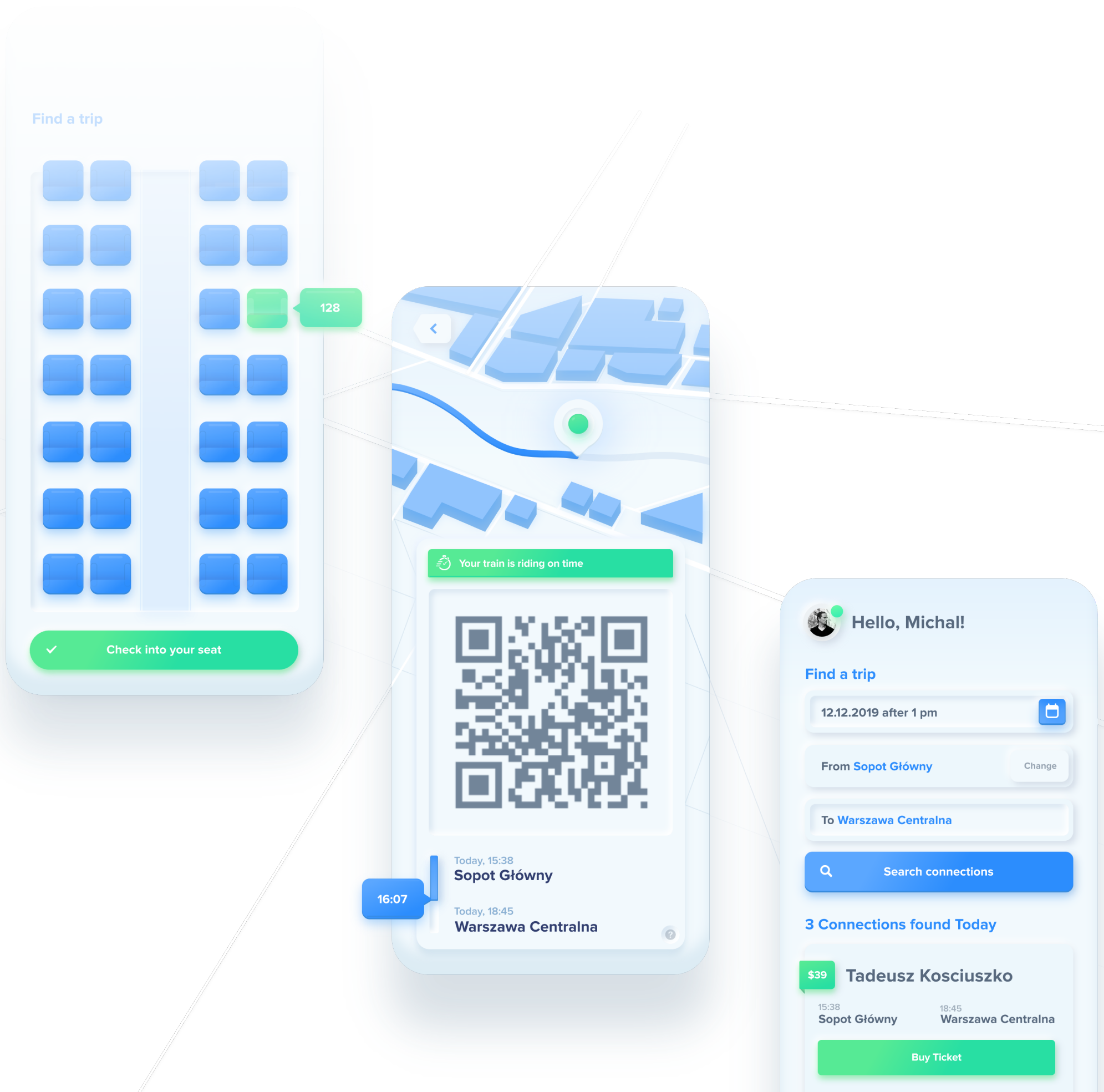
color: #EF1818;
letter-spacing: 0;
text-align: center;

ADD TO CART >

STARTUPS

Startup founders will understand how even with smaller budgets, you can make your MVP shine. After all, a professionally built prototype will give you better insights about your Product Market Fit, than something resembling a half-done wireframe. Maximize your chances to gain user's trust as that leads directly to success.

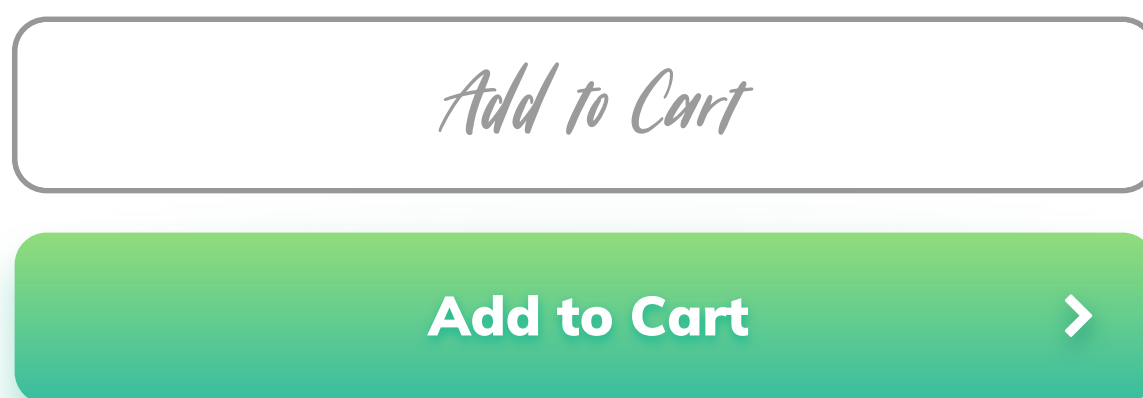
Remember - people "buy" into a product or service with their eyes first. Make a great first impression, mix it with an excellent value proposition, and you've got yourself a hit!



MARKETING, SOCIAL MEDIA NINJAS

Marketing and social media ninjas will learn how even the smallest, easy to do tweaks can drive conversion and readability up. We all want happy, engaged users interacting with what we do. Eliminating the roadblocks will help with achieving that.

We'll show you how to wrap your content visually, so it's easy to understand and has that "premium" look & feel people trust.



Which button do you think will have more conversion?

CURIOUS TYPES

All digital product users will learn how to see the difference between a good and great product. You'll gain critical thinking and knowledge of what to look for in an app or a website. That, in turn, will let you get better at all the other aspects of what you do. Even if it has nothing to do with pixels displayed on a screen. Precision and aesthetics can go beyond that, right into our RealWorld™.

DESIGN?

The main ideas behind great products are universal. Planned action, precision, and care should go into everything we do in life.

Let's make GREAT things.



3.

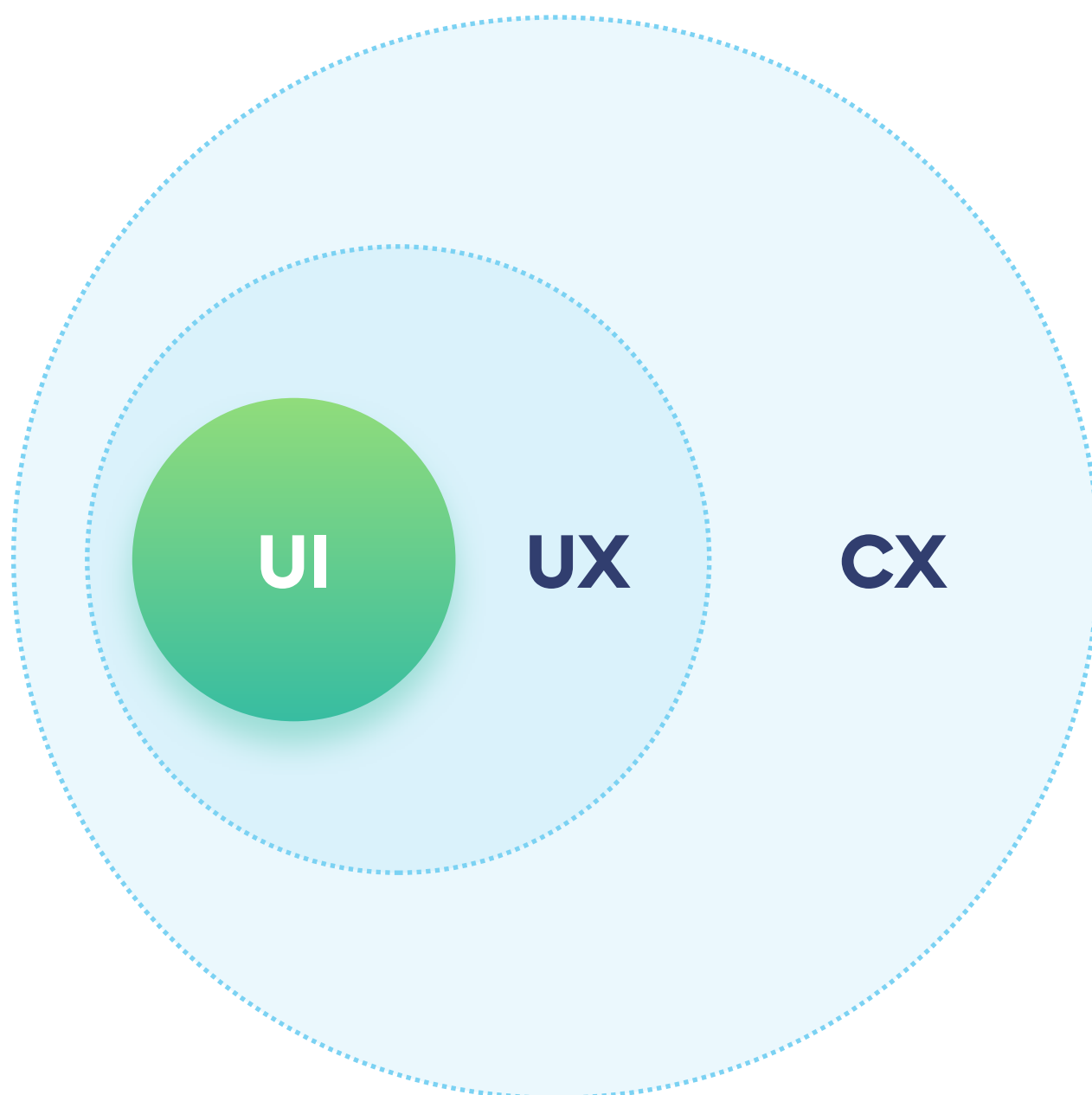
UI vs UX

UI vs UX

DEFINITIONS

Where does the humble interface fit inside the trendy industry terminology? It's often looked down upon by "real user experience specialists" as "painting with crayons." We believe this is the wrong approach. What we see is just as important as information architecture and business goals.

Customer experience is a high-level category with UX being a part of it, while UI being a part of the UX process. The diagram below shows the relation between these elements. To build a great product, you need to take care of every little detail with great precision.

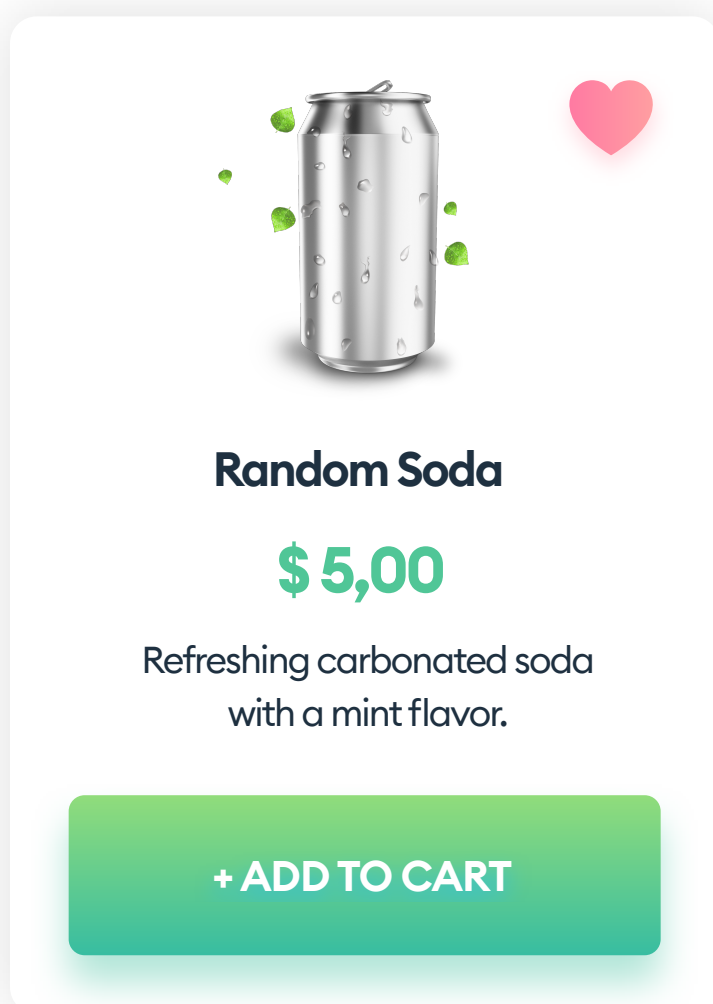


USER INTERFACE

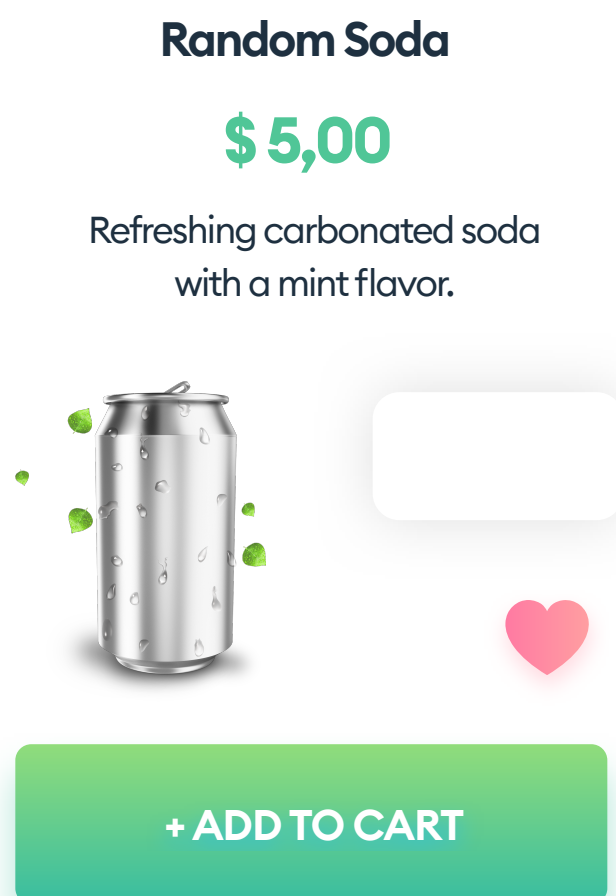
UI or User Interface is the visual representation of a digital product. Its obvious use cases are apps and websites. UI is the link between the user and the functionality of a product. It helps to achieve the desired outcome through a series of human-machine interactions. It's a series of text, shapes, graphics, and photographs combined in a way that allows for a fluid, natural interaction.

An interface is a mix of grid, layout, typography, colors, animations, and microinteractions. In other words, UI is all that we experience - mostly with our eyes. UX defined information architecture is a blueprint for UI's refined final look.

A UI designer is responsible for the final look & feel of the product. The job is to define its unique style and make it fit the target market. That visual representation should be readable, usable, and devoid of any unnecessary, distracting elements.



Product Card



Component set

“

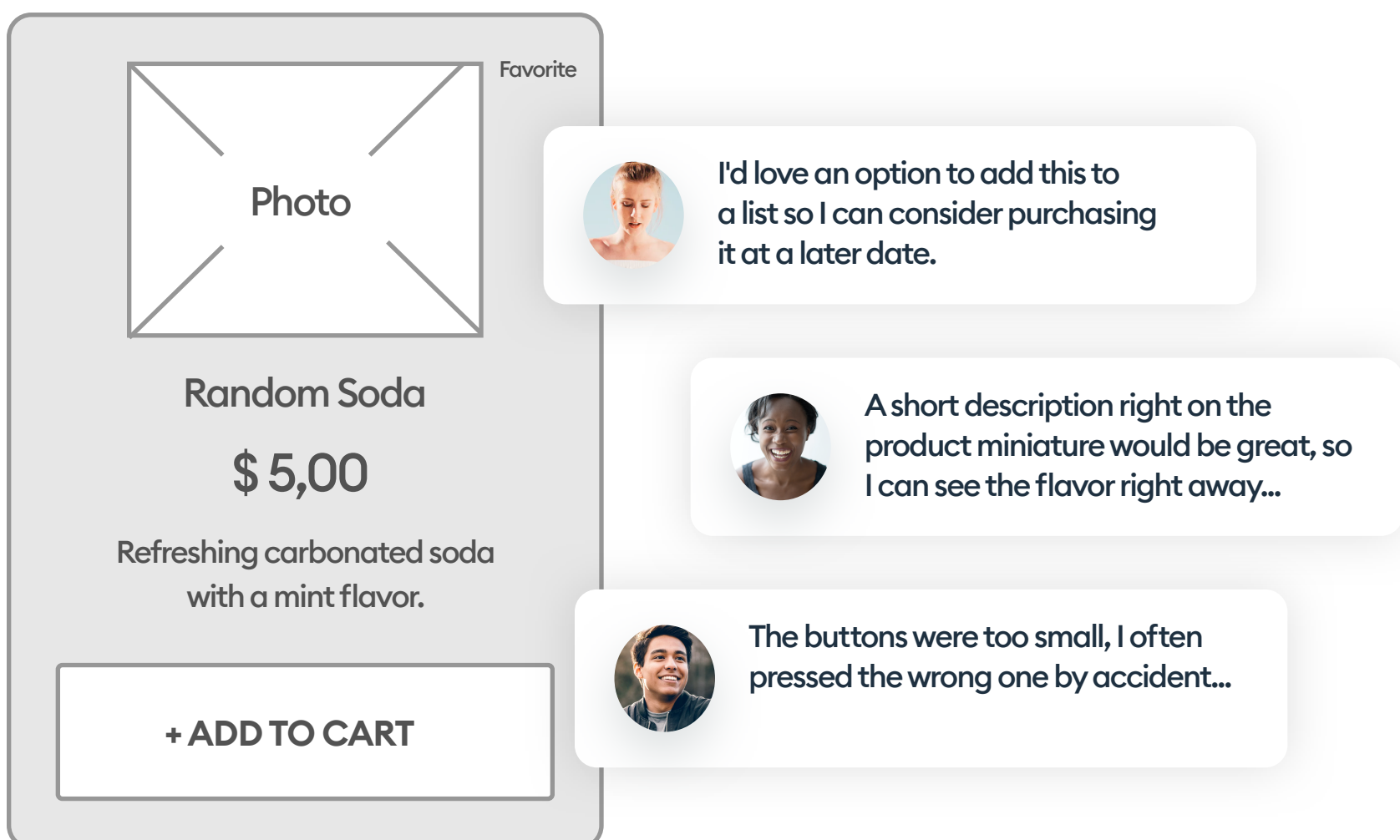
**An Interface is like a joke.
If you have to explain
it, it's not that good.”**

USER EXPERIENCE

UX or User Experience defines and studies how easy it is to use the product. With digital products, that means the interface, navigation patterns, and communications. The goal of UX is to allow the largest possible group of users the ability to understand and use a product. If the role of a UI designer is the look & feel, the role of a UX designer is defining how it will work.

Aside from information architecture planning, UX also covers various kinds of research (surveys, A/B tests, focus groups, interviews, workshops, and more).

UI is a part of User Experience because readability and looks are also influencing the ease of use and shape our feelings towards a product. This book covers the most basic UX principles but focuses heavily on the UI perspective of design.



A basic information architecture (wireframe) and user comments to implement.

CUSTOMER EXPERIENCE

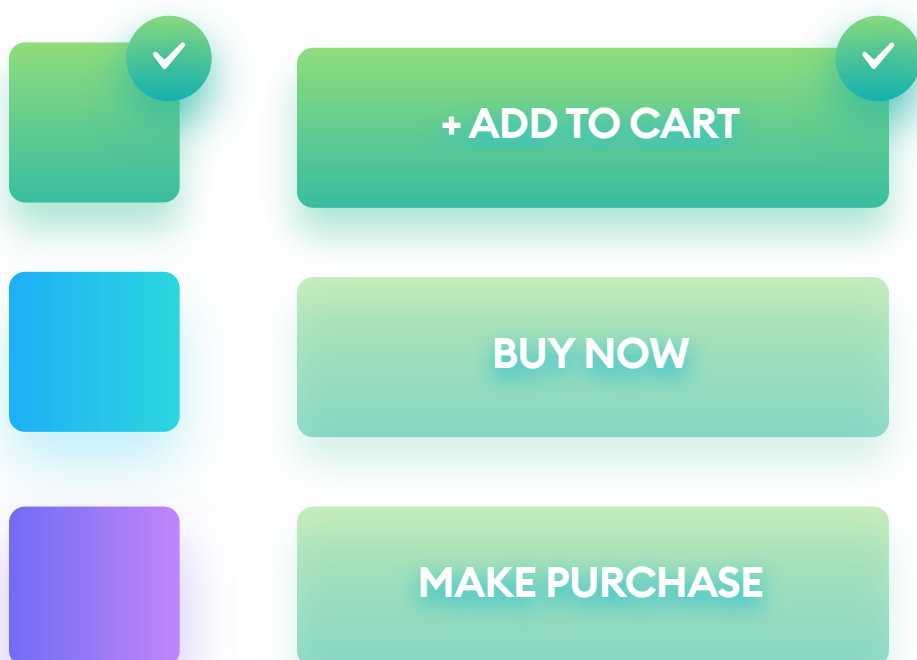
CX or Customer Experience is often confused as being another name for UX. The truth is that CX is a top-level process that defines not only how your product works, but how your entire company operates.

The CX process is the UX of your product, but also your branding, marketing, or friendly customer support on the hotline.

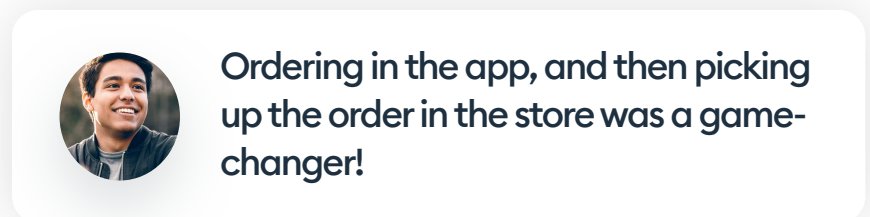
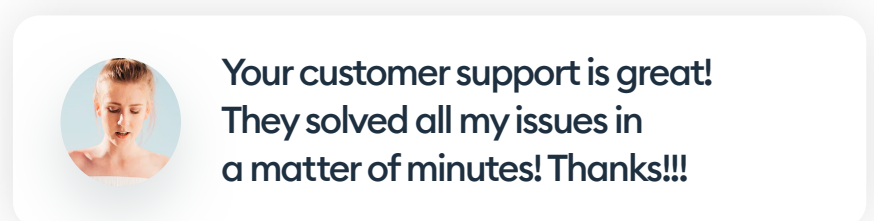
You can design Customer Experience by defining clear, consistent rules for your entire brand. Those rules apply to all the real-world experiences, as well as UX and UI. They even influence the way you write your copy.

A good example is one of the popular fitness apps. They changed the boring "save your workout" message to an "I'm awesome" button. Customer engagement went up.

All of those elements combine for a consistent, coherent vision with which our potential customer can quickly identify.



Changing the brand colors to a more positive, green gradient yielded an improvement in user perception of our brand. We also tested various types of label text to see what works best.



Merging the digital world ideas with the real world.

PRODUCT DESIGN

In the last few years, many of the UX designers started calling themselves Product Designers instead. The term merges most UX competences with UI and basic research, allowing you to solve a wide range of potential problems.

This new naming convention comes from the fact that UX designer, as a title, doesn't show which skills the person has. UX designers are often product owners, researchers, wireframe makers, or all of those combined.

A product designer can help with both the business processes, choosing the right building approach, and final interface designs.

It's often the most versatile person in the company that can guide the rest of the team, including the developers.

Some say Product Design is User Experience design 2.0.





4.

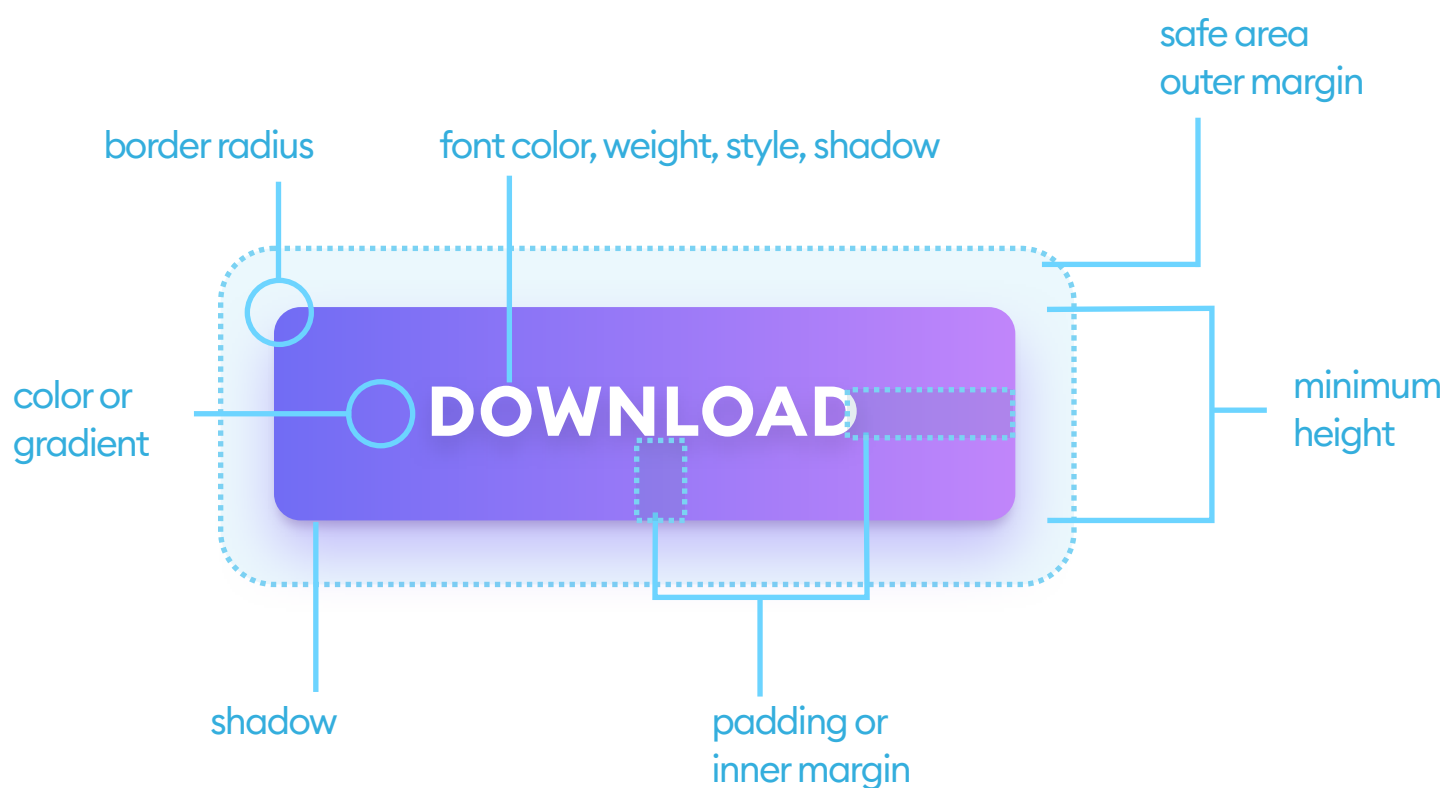
Design and Art

Design and Art

ARE UI DESIGNERS ARTISTS?

There often is a perceived equal sign between UI designers and graphic designers. It's quite an oversimplification that needs clearing. A graphic designer is commonly associated with being an artist, while UI design is more of a craft, that adheres to a set of strict rules.

User interfaces are not "painted with pixels," as many people think about them. They're a precise, visual blueprint for a working product. They're not artworks on a gallery wall, but rather an architectural plan - accurate and with very little artistic chaos.



Great interface design is pretty rigid in terms of its rules, constraints, and regulations. Given too much artistic freedom, it can deviate highly from the functional specifications, making the resulting product very hard to use.

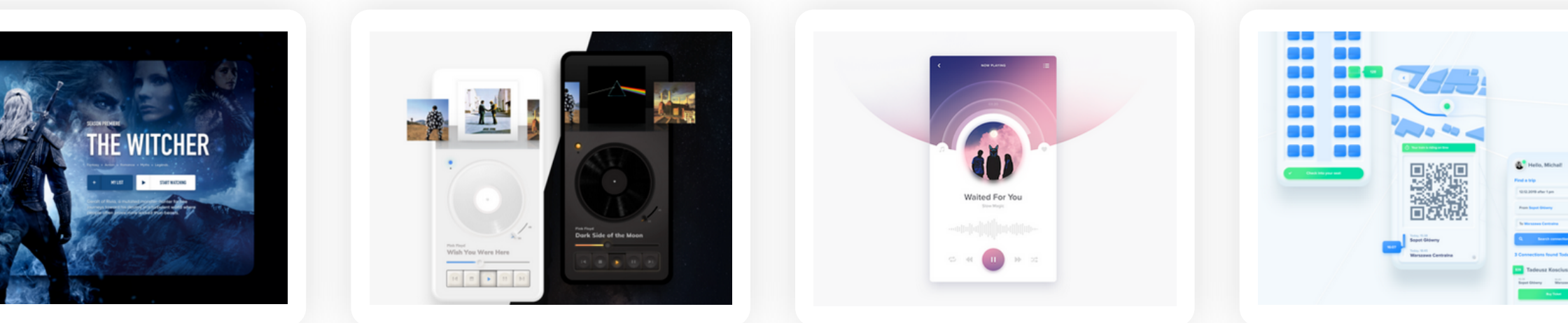
“

**UI design
is definitely
NOT art.”**

ART AND DESIGN

While it may come in handy, being an illustrator or a painter is not necessary for UI design. The best interfaces are often a result of manually entering the right numbers into boxes, instead of artistic strokes on a graphic tablet. You can be a successful UI designer with just using a couple of basic, good practices without ever getting out of the comfort zone.

Readable, useful, and aesthetically pleasing result is the culmination of a successful UI design.



A couple examples of us testing various visual styles, as seen on our Dribbble profile. Most of them require additional work to become a truly usable digital product.

DRIBBBLE

Dribbble.com designs are probably the closest intersection of art and UI design. The main goal of posting a project to Dribbble is to achieve the highest possible aesthetic effect - often at the cost of functionality. Many of those designs are either impossible to code, unusable, or not accessible. Bringing them back towards something that works would often require a lot of changes.

That's why it's good to use them as inspiration for visual ideas while being critical and judging the usability of everything you see.

“

**Dribbble designs
often have poor
usability, but they're
a great source of
visual inspiration.”**



5.

**What makes
UI good?**

Good UI?

WHAT TO LOOK FOR?

Great UI is a sum of many parts. It is the interplay of colors, typography, interactions, layout, and style that requires a lot of work, experience, and imagination. If you're starting as a junior designer, try to use the tested good practices first. Innovation and your very own style will come naturally, with experience.

A great UI is rarely fully done. In most cases, it's an ongoing process with changes and updates coming from user feedback and innovative ideas.

A good place to start is by following the **10 Rules of Good Design by Dieter Rams**. The original set is often called 10 commandments of great design and is framed on walls of many design studios. Those rules, however, were meant to initially cover designing real-world objects, so we adapted them slightly to fit the UI world.

The original set of rules was established back in the 70's, but they're just as relevant today. Precision, attention to detail and careful planning are all necessary for a great product to take shape.

Safe design methods will allow you to learn these things by following rigid guidelines. Then, as you get more experience, you can start to bend these rules or flat-out break them. Always start with the established good practices, however.

Remember - good design is as little design as possible!

“

**Good design
makes a product
easy to use.”**

GOOD DESIGN IS INNOVATIVE

Innovation is great, but it can't come at the expense of usefulness.

A good example is the Twitter "Pull to refresh" interaction pattern. It was both completely new, yet strangely natural, fun to use way to refresh your content. It's nearly invisible, yet it made its way into countless other products. Don't innovate for the sake of being different - triangular buttons won't replace the classic rectangular ones anytime in the near future.

The most innovative you can get nowadays is with microinteractions. Defining how familiar elements affect each other with user input, is still open to new, creative ideas.

GOOD DESIGN MAKES A PRODUCT USEFUL

Usability is the base of User Experience. UI Usability, however, is mostly about clarity. Decorative fonts, lack of grid or eye-watering colors decrease the UI's readability, thus decreasing its usability.

GOOD DESIGN IS AESTHETIC

Aesthetic is quite subjective and a lot has changed with how we perceive it since the 70's. One thing that hasn't changed is that aesthetic comes from internal harmony, made by grid, typography, whitespace and a well planned layout. All designs that are based on good principles (like the Golden Ratio) will be more aesthetically pleasing, than random ones.

GOLDEN RATIO

1.618 is a number that allows to set hierarchy in a natural way.

GOOD DESIGN IS LONG-LASTING

If you get your grid and whitespace right, the design will still have merit, even after a huge shift in design trends. Maybe the gradients or illustrations won't hold up as well, but design precision and clarity don't age.

GOOD DESIGN MAKES A PRODUCT UNDERSTANDABLE

Understandable UI is a mix of good readability, proper hierarchy and the right type of content. Most of these things come from the right approach to grids, layouts and overall precision. Once the design part is done, you can work on the language part to match.

GOOD DESIGN IS UNOBTRUSIVE

With modern design tools it's especially easy to go all the way with colors, animations and sugar-coating. Those kinds of designs look well on Dribbble portfolios, but often forget that the role of UI is to deliver content.

How the actual Instagram app looks, is less important than the photos or videos you can see inside it. It's especially relevant for products based on showing content.

Try not to fall into the trap of form above function. A great interface fades away after a while - we stop noticing it and using it becomes natural.

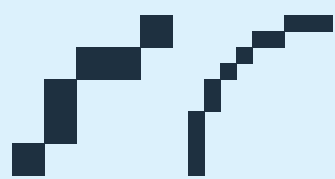
GOOD DESIGN IS HONEST

Honesty - or the lack of overpromising - is easy to determine especially in more complicated, content-heavy designs like charts and graphs.

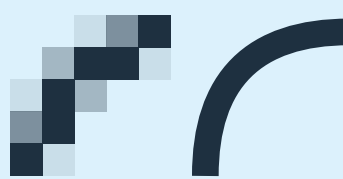
It's a known fact, that a fluid, sinusoidal, anti-aliased graph looks better than a jagged line. More often than not, however, real data doesn't look that way. It's one of the white lies we try and feed our users to like our product more.

ANTI-ALIASING

It's a way of adding transparent pixels on diagonal and rounded edges so they look smoother from a distance.



Without anti-aliasing
two zoom levels



With anti-aliasing
two zoom levels

GOOD DESIGN IS THOROUGH DOWN TO THE LAST DETAIL

Precise design can be seen especially with low-importance elements still being created with care and attention to detail. Microinteraction while clicking on a checkbox, or a small X used to close a window should not be treated as an afterthought. Make even your terms and conditions screens beautiful, and the users will notice.

GOOD DESIGN IS ENVIRONMENTALLY FRIENDLY

Even virtual products (like most UIs) affect the environment in some way. Sure, it's not as big of an effect as with physical manufacturing, but we should consider the server power consumption (optimizing the app for faster loads can also be done in UI) and systemic ways of building the products which in turn allow to make them faster.

GOOD DESIGN IS AS LITTLE DESIGN AS POSSIBLE

Less is more. This rule goes well with the unobtrusiveness of a product. UI is not supposed to be the star of the show. It exists solely to deliver content (text, audio, video, images etc.) and it can't overpower them visually.

So a great UI is using the brand guidelines, to create best possible content delivery channel. Ornamentation and unnecessary elements can lead to an overdesigned project that negatively impacts the actual experience.

Design is a path, not a goal.

Design smart.

“

**With design,
less is always
more.”**

Perception

6.

Perception

Perception

Perception

INTRODUCTION

Our brain uses a set of methods and shortcuts while processing and organizing the things we see. Knowing and understanding these methods is a crucial element to a successful user interface design.

Rules of perception are perceptive mechanisms our mind uses to connect and organize partial information into a coherent whole. They're the top-level design knowledge that we can also use unknowingly. Many of the methods are common sense.

The underlying idea for our perceptive mechanisms is that **the brain subconsciously tries to connect the dots** and see an image as a whole, instead of a subset of smaller elements. We group and connect similar objects and easily differentiate things that don't belong in the group.



While designing the UI, our goal should be to do so consciously. We need to be able to back our decisions with facts. Why did we create this element in a specific way? What was the reasoning behind it?

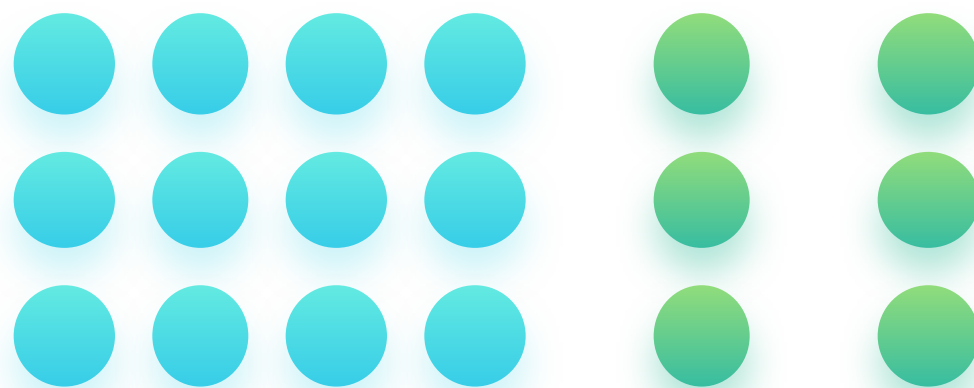
Knowing the rules of perception can make it a lot easier to explain our choices and will help with avoiding many common mistakes. Applying that knowledge will make our designs better and easier to understand.

“

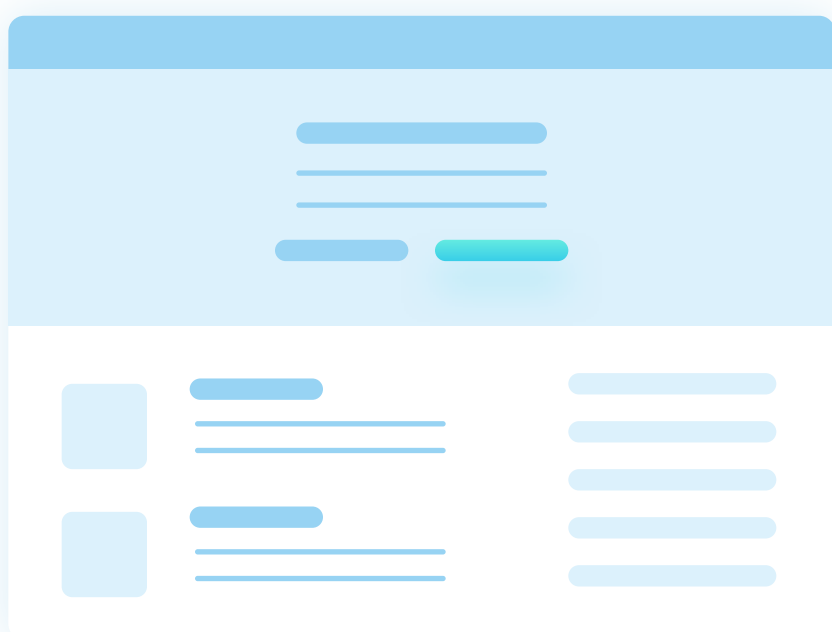
**Our brain
is always trying
to fill the gaps.”**

THE RULE OF PROXIMITY

Objects placed close to each other are automatically understood to be a group. Placing them farther apart will give the impression that they're entirely separate entities.



In UI design, we can use the rule of proximity to group similar components together and organize the general layout. It works well for the main navigation, cards, buttons, content, and icons. You can use whitespace to move different elements further apart, creating a natural hierarchy. Proper hierarchy helps with scanning and understanding the different types of the displayed information.



Logical, grouped layout



Randomly placed objects
make the design look chaotic.

It's one of the essential rules that directly affect the usability of the interface. A product not adhering to this rule (and using random distances instead) will be chaotic and much harder to understand.

THE RULE OF SIMILARITY

All visually similar objects are considered a part of a group. All elements that are different visually, our brain sees as individual objects.



But what do we understand as a similarity? It's all of the defining features like color, shape, size, texture, or position in space. Some of these features we process much faster than others. The strongest similarity is with color, then size and shape.

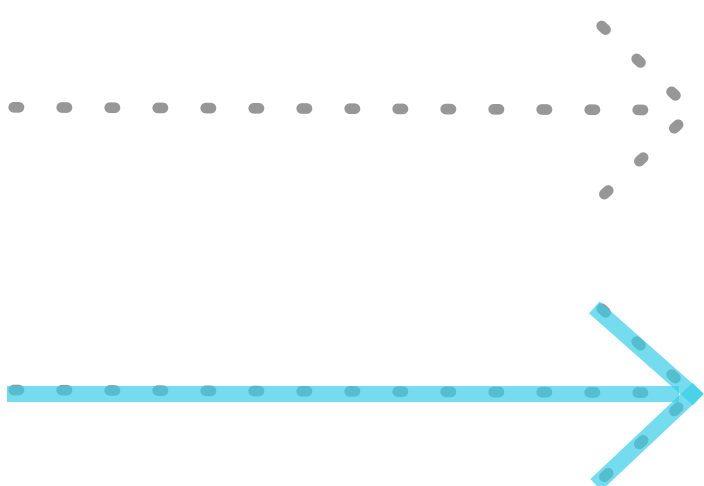
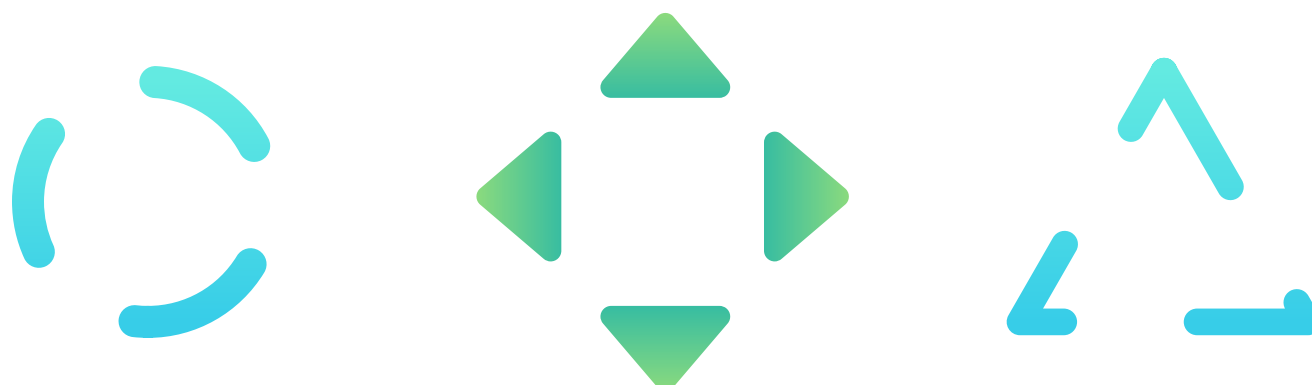


Our brains groups similar objects together automatically.

Keep in mind that an object drastically different than everything else (an anomaly) won't be able to fit into any existing group - making it very visible. You can use this differentiation effect to make your Call to Actions more visible.

THE RULE OF CLOSURE

A set of objects can naturally form a recognizable shape. All the empty spaces between those objects we subconsciously fill. It means our brain finishes incomplete shapes for us by adding the missing parts.



You can use the rule of closure in many different ways. One example is a dotted or dashed arrow. Our brain fills the blank spaces and connects the dots. We perceive the result, not as a set of dots or dashes, but rather a full line.

It also helps to discern more abstract shapes in icons and give them meaning. Most icons are purposefully simple not to overload us with information, and our brain fills in the blanks.



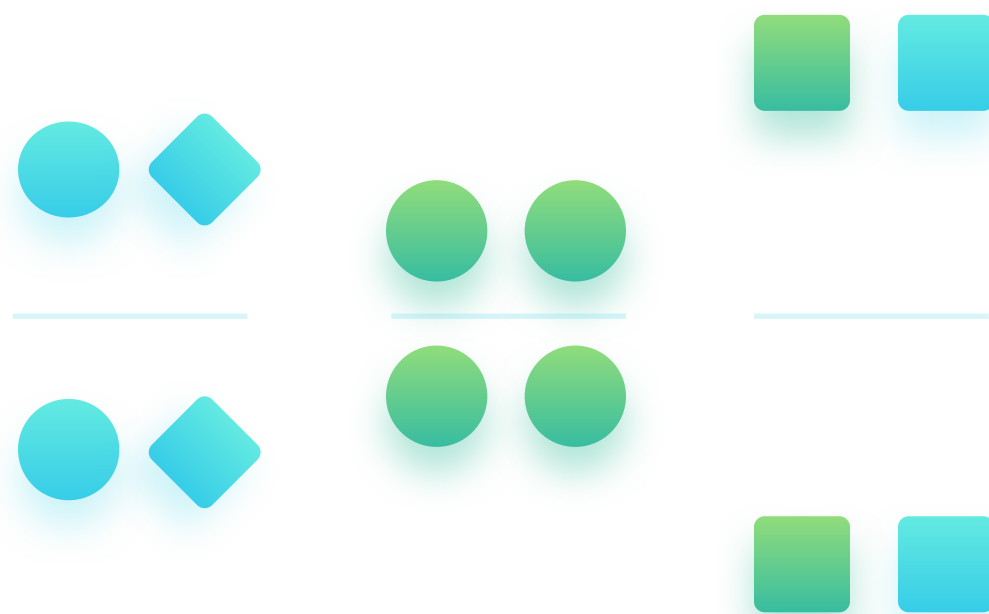
IBM



Another excellent example of the subconscious connecting of empty spaces is the WWF Logo Panda shape or the IBM logo, where dashed lines form the letters.

THE RULE OF SYMMETRY

Objects aligned symmetrically, we understand as connected, no matter the space between them. Once there's symmetry, there's order, and we are very quick to see and understand that.



Our brain loves symmetry! All objects that are symmetrically aligned are more visually satisfying and are considered more beautiful. Some say that symmetry is the aesthetic of fools, and while this may be true in print and graphic design, it is not the case in UI. Symmetry is king.



A symmetrical layout like the above will make our design easier to understand and more friendly.



No symmetry causes chaos - we don't know which object to focus on.

Just as with the rule of proximity, asymmetric placement of an object will make it instantly stand out.

THE RULE OF CONTINUITY

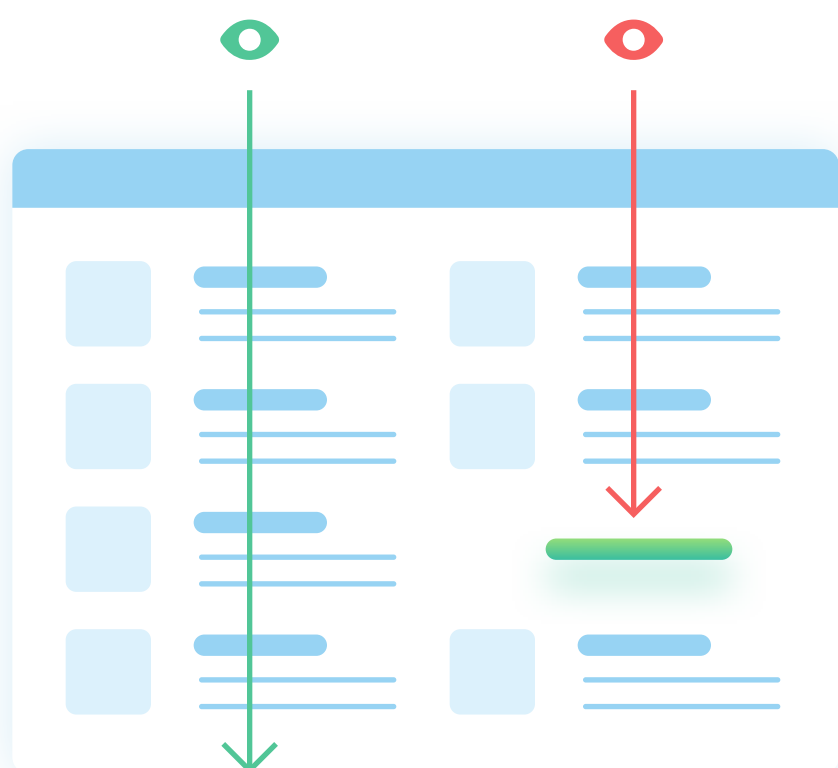
Objects aligned along a continuous line are also naturally grouped.

Our brain is always looking for the smoothest path to follow. That's why when using the rule of continuity, make sure the line (or curve) is uniform and predictable. The more uniform the line, the easier the resulting shape will be to identify correctly.

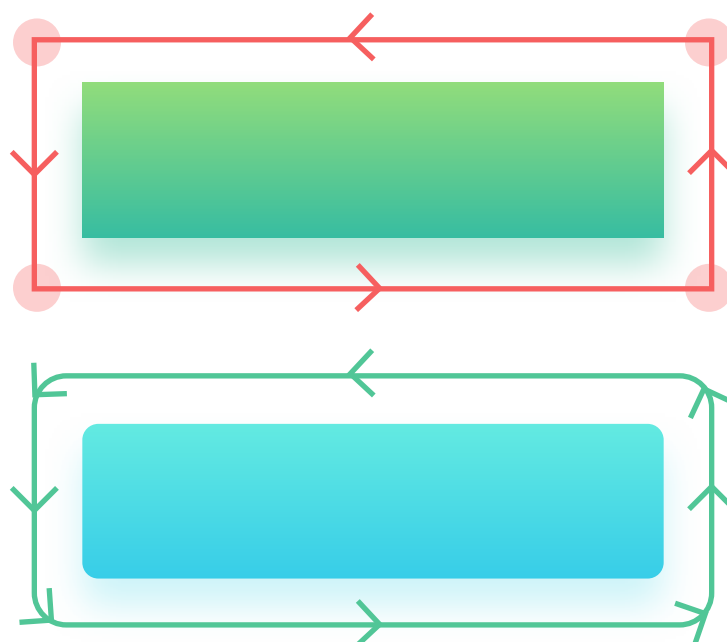


When designing an interface, we use the rule of continuity in a couple of ways. One of them is aligning the content of a side-scrolling carousel. We tend to scan the similar content types by following the text from left to right and jumping between the carousel boxes.

Continuity also helps when designing the vertical rhythm. Similar types of content should always be aligned to the same edge. An out of place element breaking the continuity will naturally slow down our scanning.



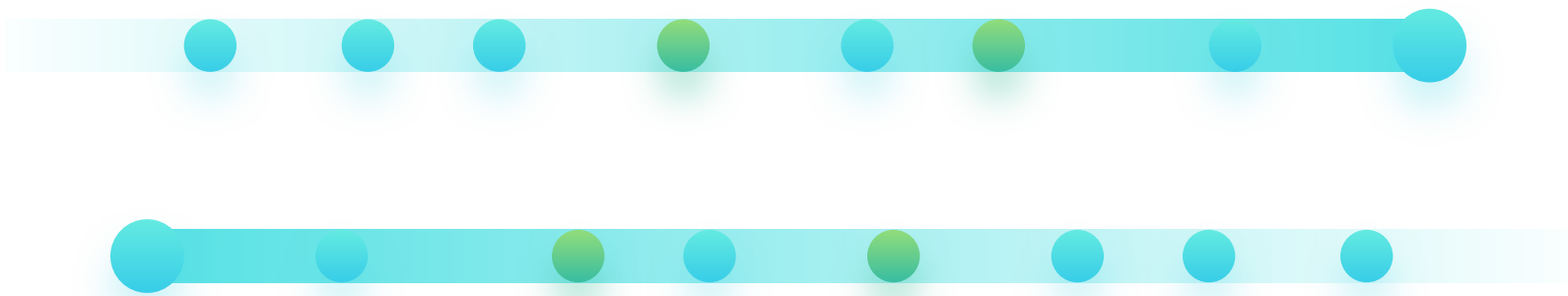
Above, you can see two examples of scanning a page from top to bottom. In the right-side column, a lonely, out of place element breaks the flow.



Sharp corners are slightly slower to process, as our eye has to pause and 90-degree turn on each side. A smoother, rounded edge will help our eye go around much faster.

THE COMMON FATE RULE

Objects moving in the same direction, with the same velocity, are considered to be a group.



One of the use-cases of this rule is the V-formation of birds. Even though there are many birds in the formation, we see the entire V shape as one, more significant, moving object.

This rule is helpful when designing carousels, dropdown lists, or animating objects onto the screen during transitions.

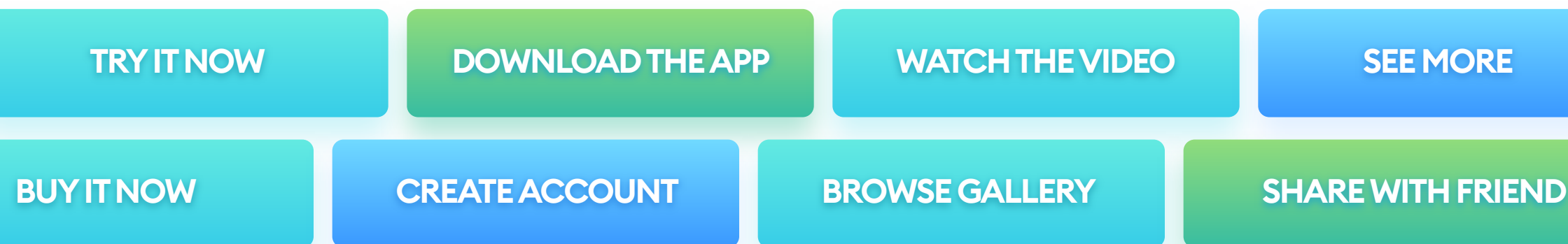


For example, our eye follows the content of a horizontally scrolling carousel.
It also follows a dropdown list as it expands down.

HICK'S LAW, MILLER'S LAW

The more options to choose from (and the more complex they are), the harder it is to make a choice. (Hick's Law)

Our minds can only process around 7 (+/- 2) objects at the same time. (Miller's Law)



According to **Hick's Law**, the number of possible choices should be limited. Having too many things to choose from can cause disorientation and irritation in users. As a result, the user can skip taking action altogether or take a lot longer to do so.

The ability to process information is directly tied to the **cognitive load**. It's the amount of information our brain can process at a time. It varies from person to person, but a popular range is somewhere around seven elements (**Miller's Law**). The higher the number of items, the longer the time needed to process them. The result is being less likely to choose at all.

While designing, try not to exceed seven options, and for optimal results, try keeping it between four and five. That means the number of main navigation items, number of buttons in a component, dropdown choices, or carousel slides. When giving the user choice, it's good to highlight the most popular or recommended option as well. It often leads to much faster decision making.

We instinctively differentiate between an object and a background.



Since most interfaces are using layers, we end up with objects and some form of background (or backgrounds). While looking at our design, the user tries to discern the background from all the essential, actionable elements.

To avoid confusion and breaking hierarchy, we need to define all the elements of our interface clearly. The background shouldn't be too busy, as it can steal attention away from the foreground. Less important objects and functionalities should be less visible and noticeably different from the important ones.



You can use contrast, position, and shadows to help the user differentiate the foreground and the background.

A visually attractive (high aesthetic) product is often seen as more useful.



According to this rule, a modern, symmetrical, and overall beautiful interface will have a more positive effect on the user. The user can outright ignore some usability problems if the visuals give a good enough first impression.

The feeling of using a high quality, polished product stays with the user much longer than the perceived ease of use. Of course, the product itself has to be usable, as the aesthetic usability effect only works if there's already some usability. A beautiful interface won't help if the product doesn't work at all.

But the main reason why we should work on aesthetics is **building trust**. A high quality, polished interface helps to build the trust of our users. A design that looks chaotic and ugly will make the users feel unsafe. Even when they can't name the exact visual problems, they see that something is wrong with the product or that the product is half-baked. The high quality of UI is especially crucial for banking, finance, and medical apps. Everywhere we need to give out personal information or where there is money involved.

THE SERIAL POSITION

We always remember the first and last object in a group best.



The most relevant content or objects should always be placed either first or last in a group. Our brain will remember them much better, and they will also catch a lot more attention than if they were placed randomly between other elements. It's an important rule to consider when building navigation, dropdown lists, or any marketing or sales materials.

VON RESTROFF'S ISOLATION EFFECT

In a group of similar-looking objects, we always remember the one that doesn't match the rest.



You can use this rule in many ways. One of the most popular isolation effect is with CTA (Call to Action) buttons. By making them a different color or size (or both) than all the other buttons on the screen, we make them naturally stand out and catch attention.

You can read more about CTA's in the buttons chapter.



7.

Screens

Screens

THE HISTORY

The UI design is all about displays. Based on the fact that the UI is everything we see, it's essential to understand what do we see it on.

In the early days of the web, it was quite simple - you designed somewhere between 640 x 480 and 800 x 600 pixels. We can, of course, go back even farther - to the first Macintosh or Xerox's very first Desktop machines, but let's assume old school color CRT's as a base here.

But that was the late 90's, and things started rapidly changing after that. Most of the displays of that era had similar resolution ranges, were super heavy, and couldn't display too many colors.

They also had low refresh rates that resulted in subpar scrolling experience. The displays of that era often limited what websites could achieve visually.



Before today's flat panels, most of the UI's were displayed on old-school CRT monitors. They often had a bit of a blur to the image (even at higher resolutions), but in most cases, you could easily distinguish individual pixels.

“

**Always design for
the base screen
resolution or @1x.”**

THE BASICS

The technology leaped forward since the early internet days. Nowadays, the phone you are holding in your hand has a resolution that didn't exist twenty years ago. They also display millions of colors and have super high refresh rates.



A quick comparison. A typical late 90's CRT display had about 1/6 the resolution of the current iPhone that's much smaller in size.

PPI - PIXELS PER INCH

At some point, 72 pixels per inch became the display standard. That all changed with the introduction of the iPhone 4 with its retina display. The premise was the pixels were so dense you couldn't easily see the individual ones anymore. A base (or 1x) resolution was the same as the previous iPhone, but the pixel density was a multiplier of that resolution resulting in sharper text and images.

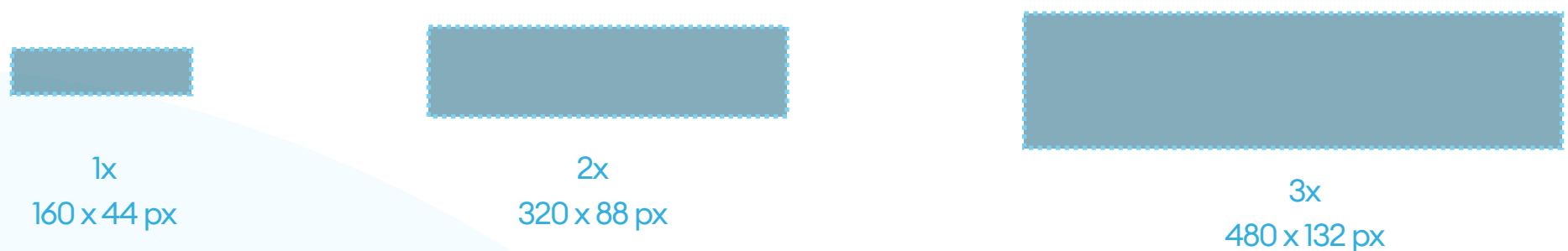
The iPhone 3GS had a 320 x 480 base resolution, which became the 1x size. The iPhone 4 doubled the number of pixels to 640 x 960, but the actual design of elements was kept at 320 x 480 and upscaled on the device.

PPI

After the iPhone 4's 2x pixel count, we started seeing 3x, 4x, and larger pixel densities in both phones, tablets, and laptops. Since the UI's we design are all inside vector-based tools, we don't need to worry about actual resolutions anymore. The base resolution of 1x serves as a template that works on low-density screens and is sharper and more precise in high-density ones.

For example, the iPhone X used a 375 by 812 base resolution, but its actual pixel count is three times larger at 1125 by 2436.

To give an example - if you're designing a button, you need to make it at least 44p high. That means that designing at 1x it will be 44 pixels (at 1x a pixel is the same as a point) while at 2x it will be 88 pixels, but still 44 points in your design.



Most popular 1x screen sizes in 2021 are:

1440 x 900
1280 x 800

Typical laptops

390 x 844

Most popular
iPhone

360 x 640

Most popular
Android Phone

1024 x 768

Most popular
Tablet

BUT YOU DON'T HAVE TO REMEMBER THIS.

Most design tools have already created presets while starting a new design that caters to all the latest devices you might need. Those resolutions are then set automatically at 1x in every case.

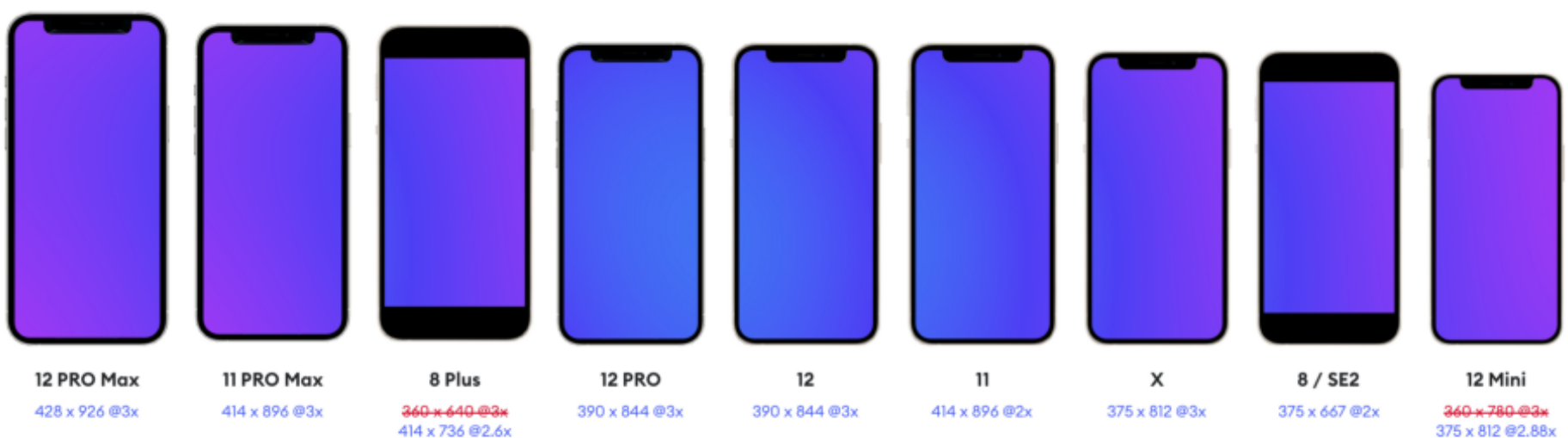
FRAGMENTATION

Sadly a growing number of screen resolutions end up with a very fragmented display landscape. We design for TVs, laptops, tablets, phones, watches, and IoT devices, which requires a lot of planning and device-specific modifications for the design to work.

The first questions to ask when starting a design is what kind of screen it is going to work on and what's the typical viewing distance. A TV app should have higher contrast and more significant UI elements than a mobile phone app, mostly because it's often used from across the room, while a phone app is just inches away from the face.

DOZENS OF PHONE SCREENS

But the place where fragmentation hurts the most is from within one device category. The most ubiquitous one - the mobile phone - has so many potential resolutions and aspect ratios that there is no way to make one design fit all. There are dozens of screen resolutions outlined in the Google Play store for Android-based mobile devices. Apple used to have a more consistent set a few years ago, but has since then abandoned the path and went towards individual resolutions for nearly all the devices.

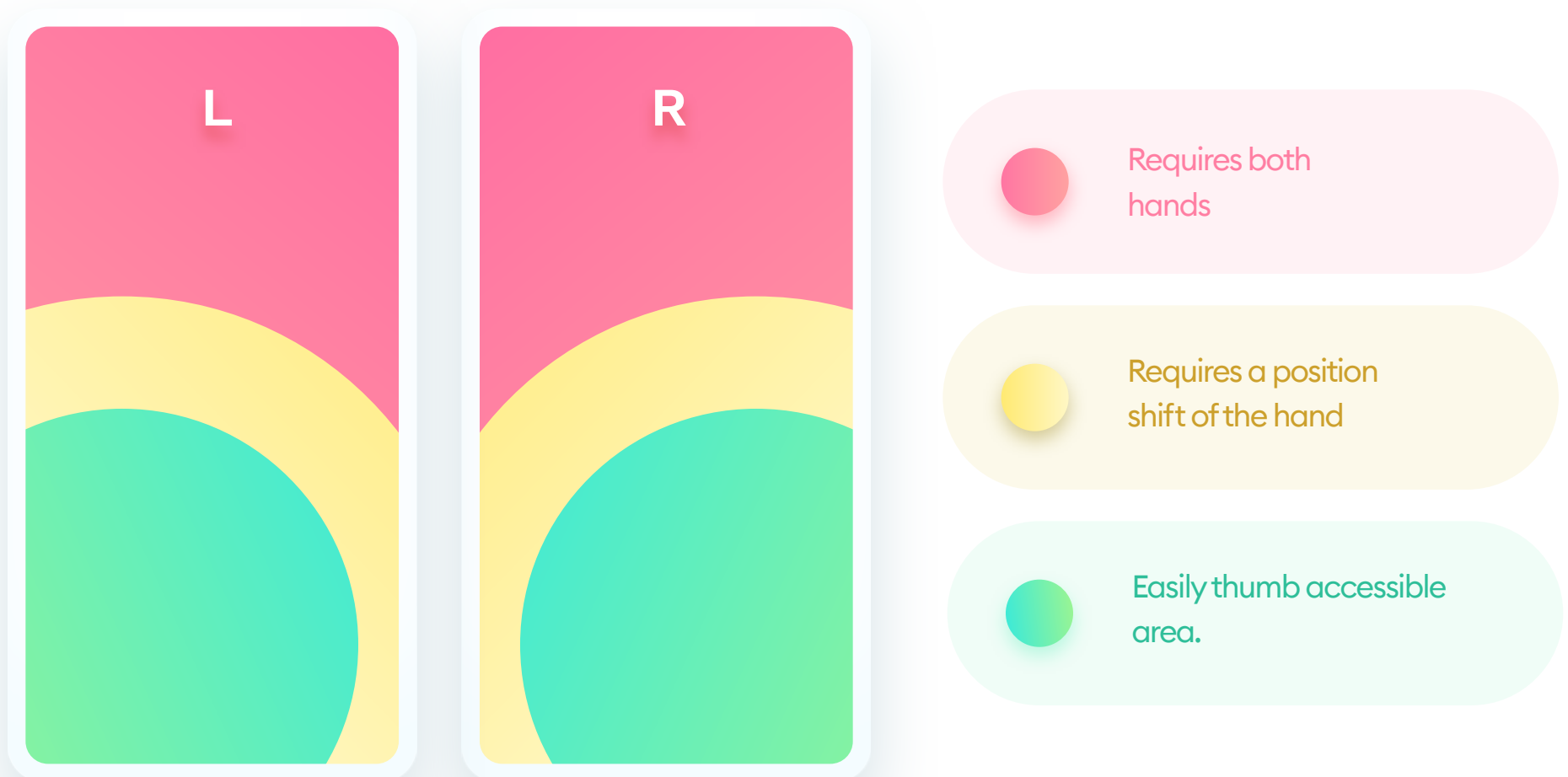


There are nine current generation sizes with six different screen resolutions for just the iPhone and countless more for Android phones.

RANGE AND REACH

In mobile devices, it is also essential to think about the average reach. A randomly pieced together interface can be complicated to use with one hand and lead to frustration.

We assume that the typical phone usage pattern is a single hand holding the phone with the thumb of the same hand doing most of the on-screen work.



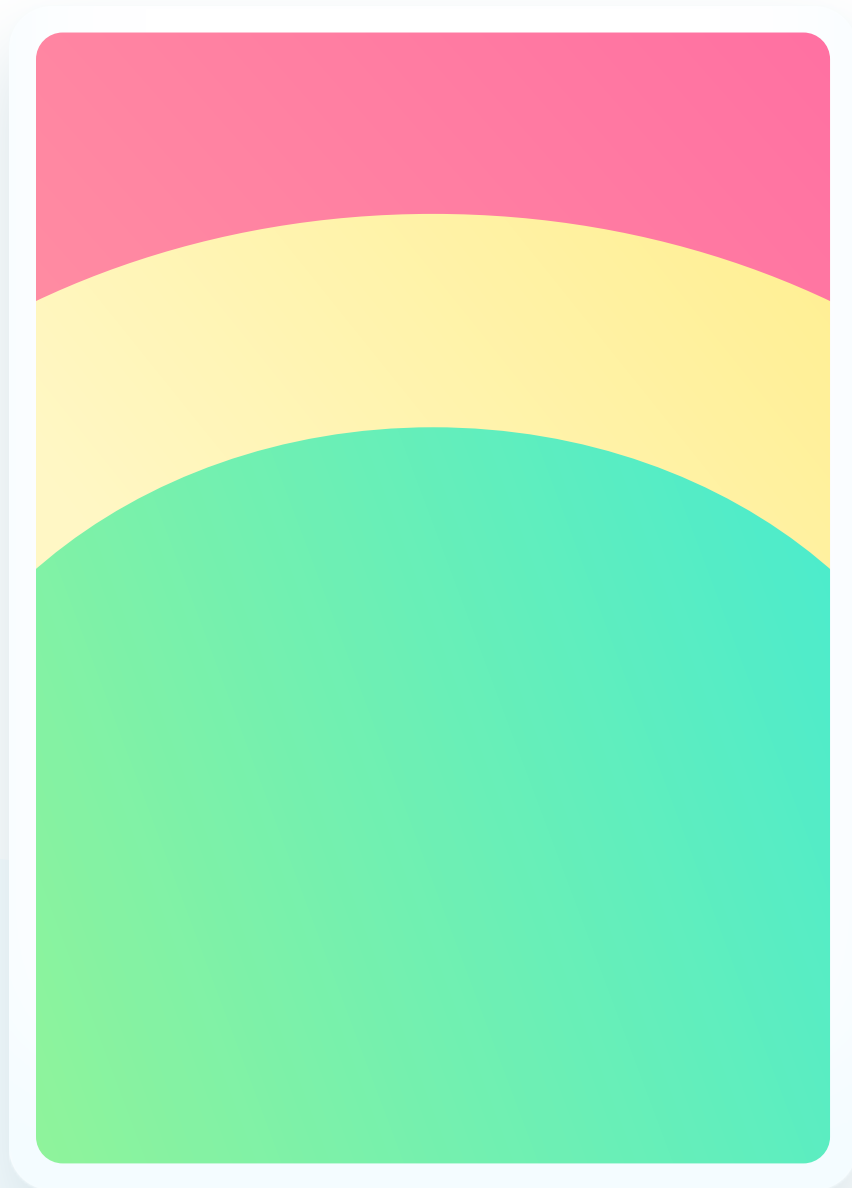
Reach can also help determine how easy it will be to navigate the product.

The popular hamburger menu design pattern is in the least favorable spot imaginable for right-handed users.

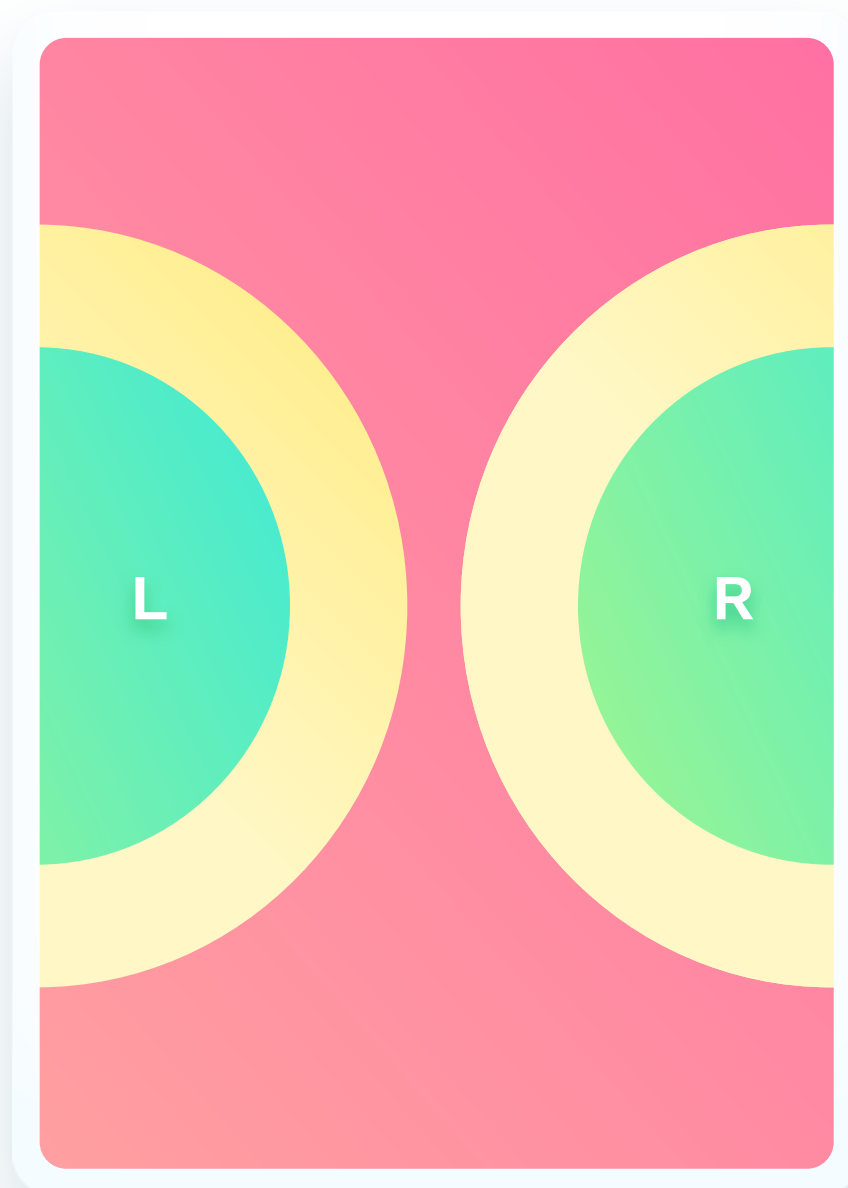
Bottom aligned tabs are, in most cases, the easiest way to design menus and should be considered for nearly every product as the first choice.

TABLET REACH

We usually use our tablets differently. The most common way is one hand holding the device, while the other one is actively interacting with the UI. It dramatically increases the range both in portrait and landscape orientations.



The most common usage pattern for tablets is using both hands. It allows for a different approach to the interface planning that merges the larger reach area with the fact of having a much larger canvas.



In the rare cases of holding a tablet with both hands, the average thumb reach extends on both sides of the device.

This pattern works great for split keyboards and gaming controls.



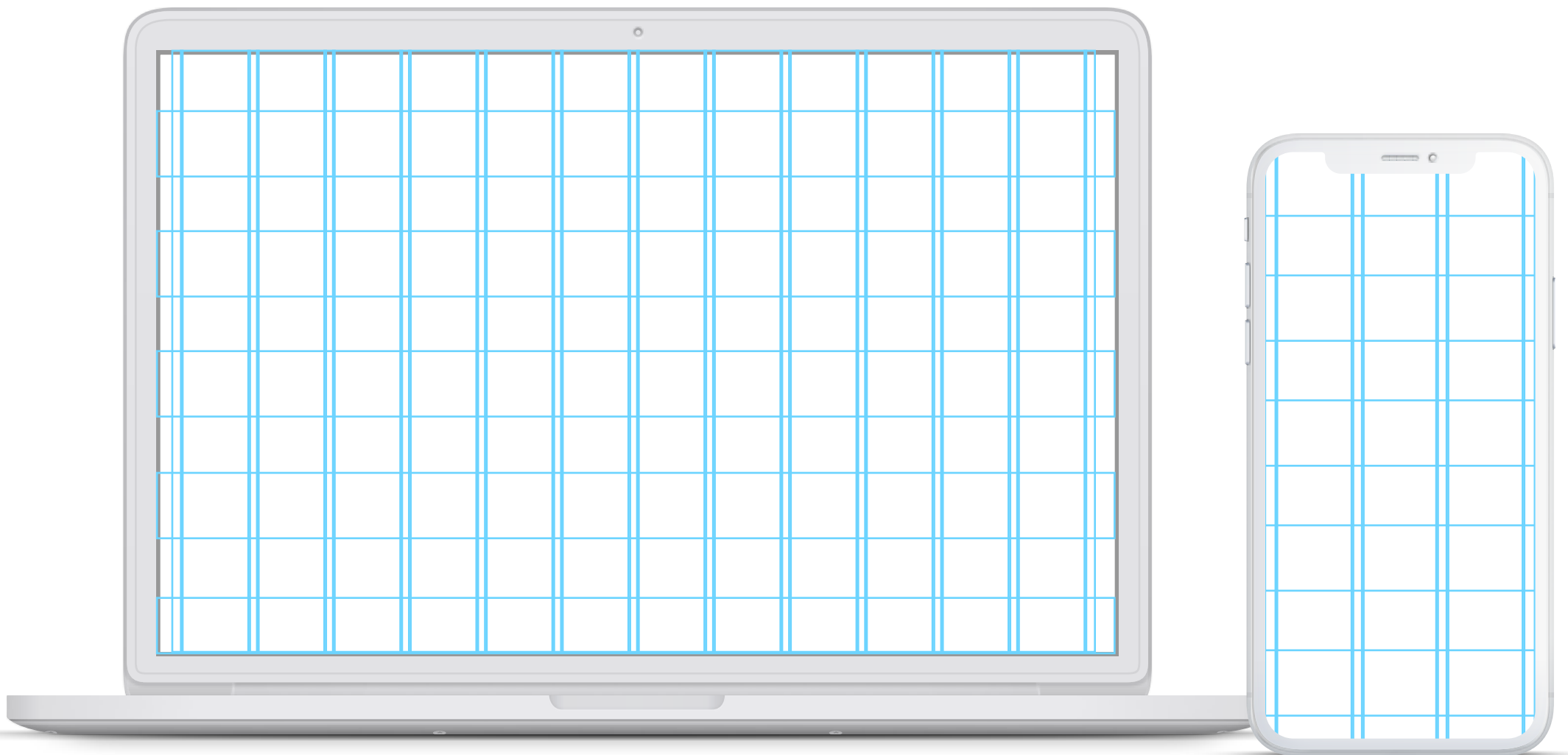
8.

Layout & Grid

Layout & Grid

THE BASICS

Grid is a structure of lines that helps keep your layout together. It's the structural blueprint of any good project as it helps to create the hierarchy between elements and allows for better understanding and processing.



Even though we don't see the gridlines in the final project (usually), even a subtle change that goes outside of the pre-set rules will have a significant impact on how we perceive the final result. It's the subconscious difference between a good project and a great one.

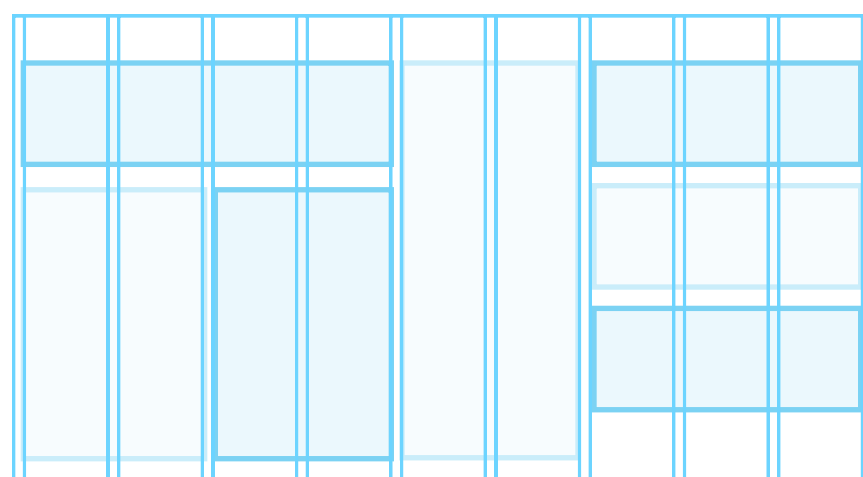
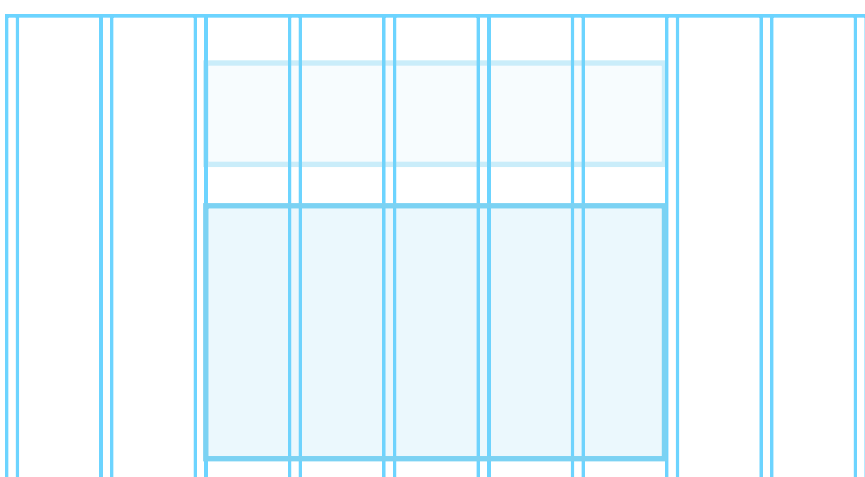
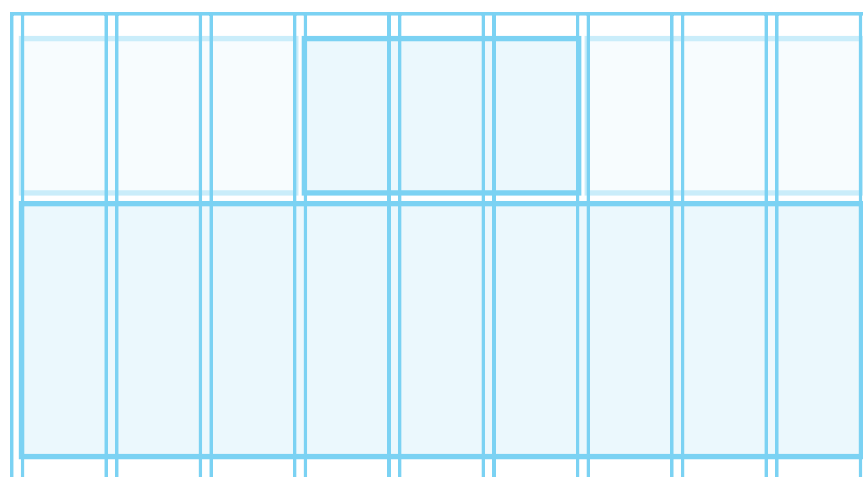
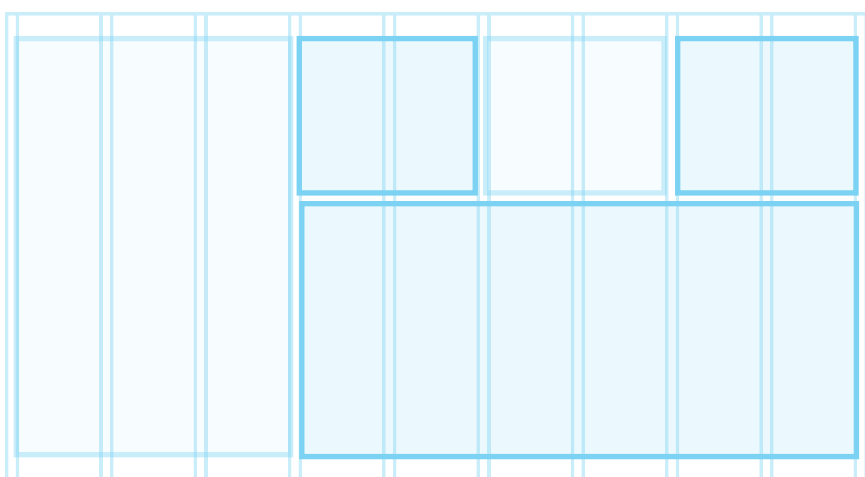
That's why it's critical to set the grid early in the project. Creating rules and guidelines near the beginning will allow for greater consistency and faster deployment.

“

**Grid is the blueprint
and backbone
of your design.”**

HOW TO CREATE A GRID?

On your initial designs, try to build out all the necessary screens and content types before attempting to pick the right grid for the project. Be sure it will cover all possible scenarios. You can also use wireframes to see the kinds of layouts you will eventually build and pick an appropriate grid.

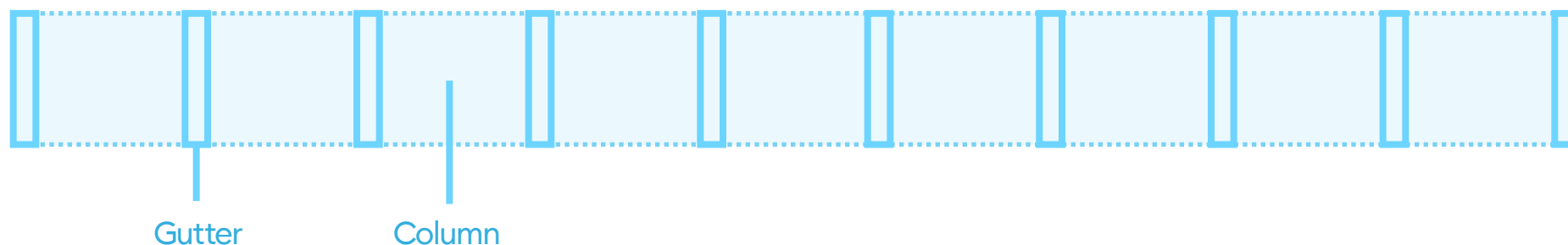


Once you start testing the grid, do it against a couple of screen types - with photos, data tables, forms, or large text blocks. That will help you be ready for every possible scenario from chart and graph heavy dashboards to simple, one-column blog posts.

If your project is going to be complicated, you can try using a more significant number of columns (up to 16) to accommodate a larger number of possibilities.

TYPES OF GRIDS

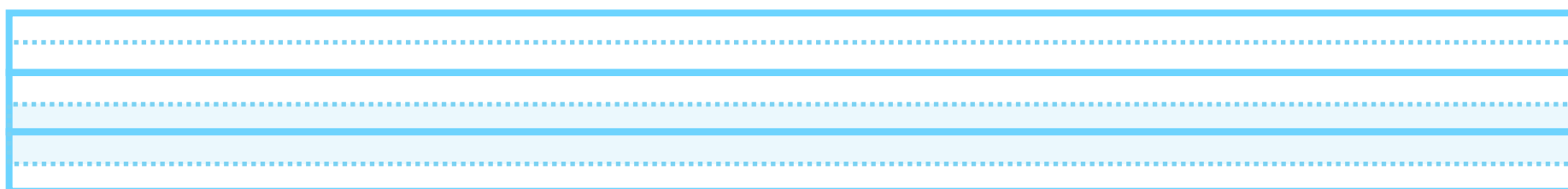
HORIZONTAL (DEFINING HORIZONTAL LAYOUT)



A horizontal grid is composed of vertical columns and margins between them, called gutters. Both of them can have either pre-set or flexible widths. They allow for a better horizontal alignment of the interface and are the base of a layout. Above, you can see an example grid with nine columns.

The name "horizontal" can be misleading when we see vertical lines, but it's derived from how we use it to align design elements horizontally.

VERTICAL (DEFINING VERTICAL LAYOUT)



The vertical grid is less prevalent. It can help set the heights of elements, sections, and vertical whitespace. It can work best with a grid-content presentation on blogs or news portals. It's composed of horizontal lines, aligning the content vertically.

The main goal for this type of a grid is to make the sections easy to scan and discern quickly, making it faster and more comfortable for the user to browse a large amount of content.

FLUID VS FIXED GRID

FLUID GRID

The fluid grid assumes an outer margin and gutter widths and adjusts the column widths to fit the screen. That makes the columns have varied width, and the gutters are what's keeping the alignment together. It's one of the most common grid types, as it allows for easier adjustments for lots of devices and screen sizes.



Here are two examples of a 4 column grid with a 16 point margin and an 8 point gutter. We only adjust the column widths to fill the entire screen size.

FIXED GRID

The fixed grid works with a set value for both the columns and gutter widths. If our screen is wider than the grid, we end up with blank space on the sides. This type of grid works quite well for websites and news portals, as stretching the content over 1400px will decrease readability anyway. In that case, it's better to leave the sides blank.



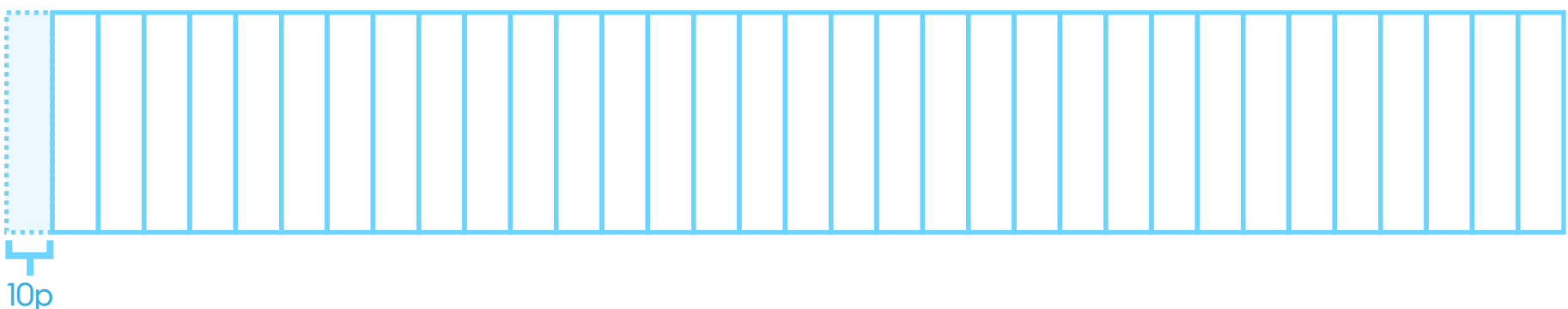
Here are two examples of a 2 column fixed grid with a margin of 16 and a gutter of 8. The column width is set at 36. As you can see on a narrow screen, it can fill the entire width, while on a wider one, it will leave space on the sides.

BASE VALUE

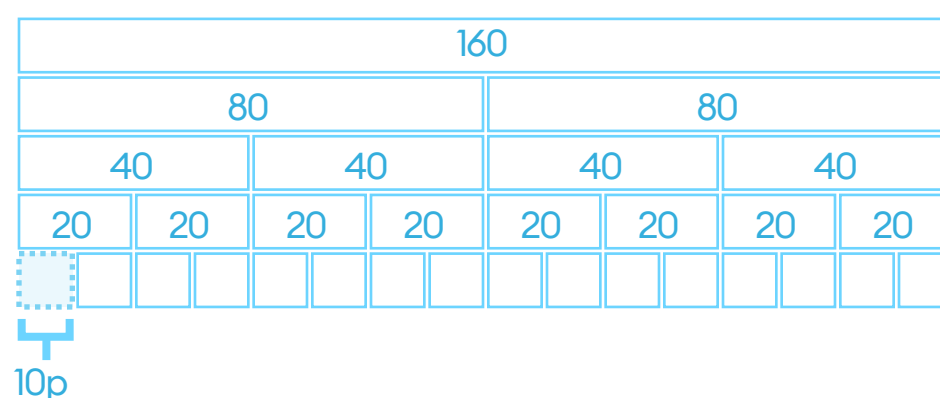
Building a grid should always start with choosing its base value. It's the smallest number we use to set all the other values. In general, all our grid values should be divisible by that base number.

10 POINT GRID

Number ten is one of the most popular base numbers. The main reason for this is the fact that it's quite easy to divide other, larger numbers by 10. So the reason for its popularity is mostly practicality and ease - you don't have to do the math. Ten is also the "nudge value" in most design apps - it's the value by which you move objects while holding the SHIFT key. In most cases, changing that default value requires an additional plugin.

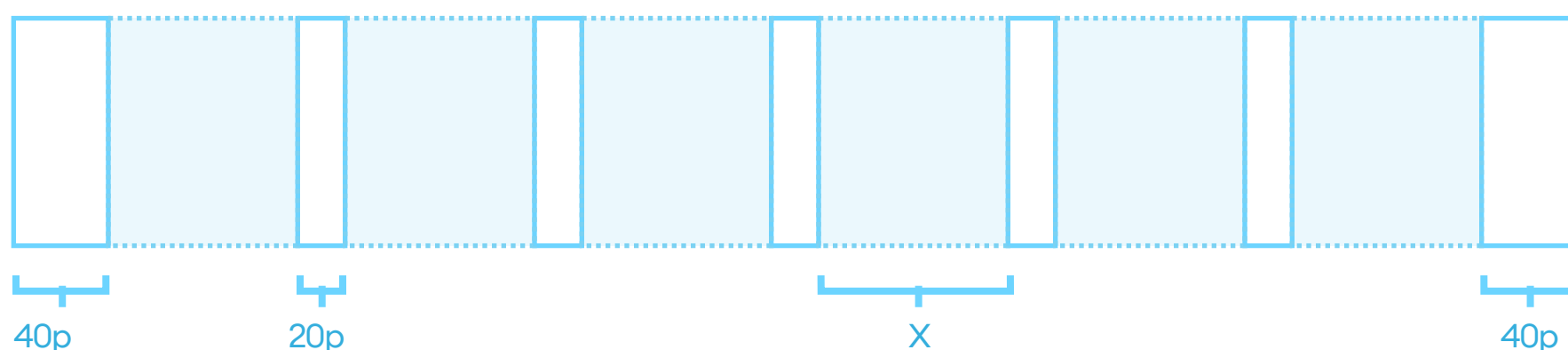


Because of that ease of use and 10 being the default nudge, it's good to start your first designs using the 10 point grid. Multiply it by two to create the base of a very simple, easy to implement grid system that you can later tweak.



10 POINT FLUID GRID

Creating a fluid grid using a 10 point baseline is one of the best starting points for any project. As ten itself is probably too small of a number to work well as our gutter (for readability), we go with double that for gutter and quadruple for outer margins. Assuming six columns, we end up with the example below.

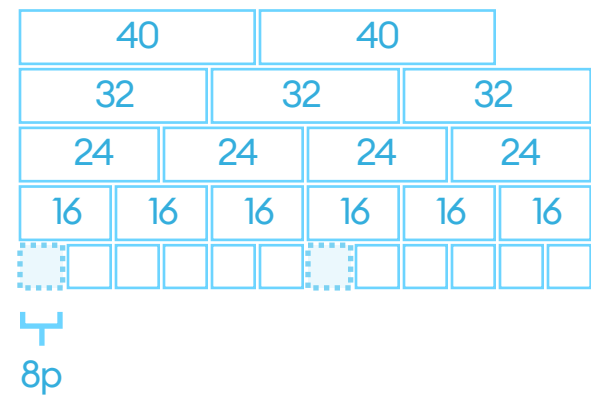
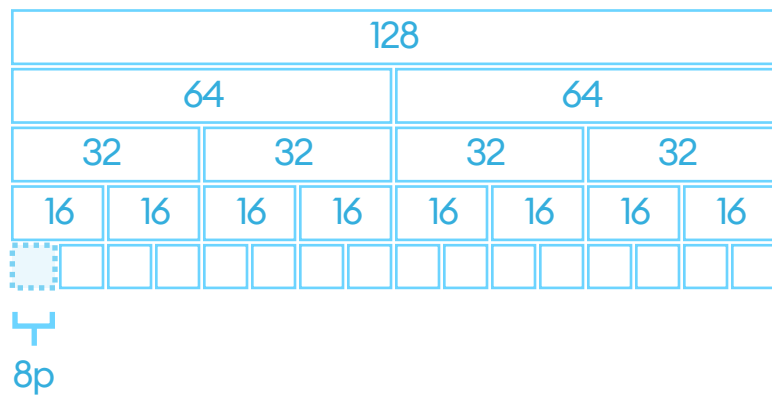


The X value comes from the equal division of our screen by the numbers of margins and gutters. With six columns and a 390p width, it's 35 pixels and will grow the wider our screen gets.

$$390 = 2 \times 40 + 5 \times 20 + 6 \times 35$$

8 POINTS

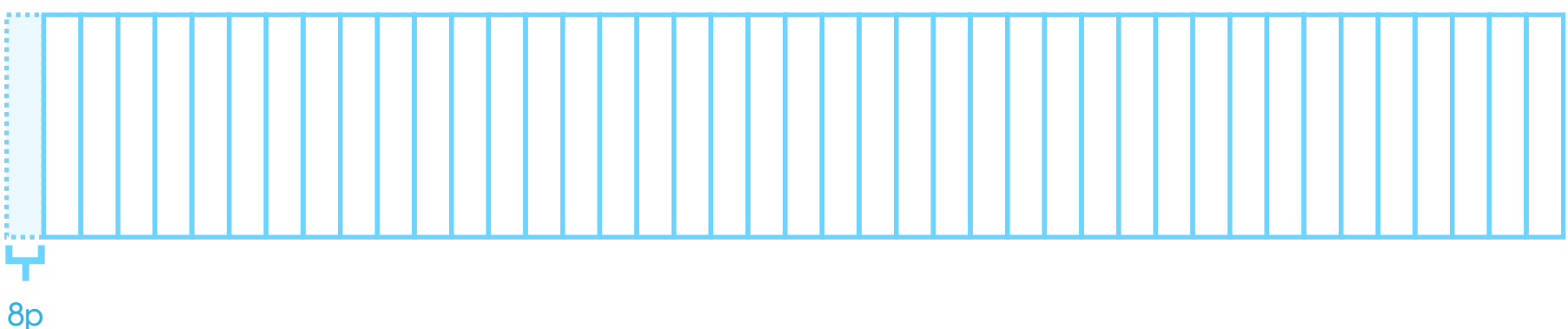
With 8 point grids, the general rule is the same as with the ten-point one. You choose eight as your base value and then either multiply it by two or - for more flexibility - add 8 to your numbers for additional sizes.



The 8-point grid is currently the most popular grid-type in modern UI design. It requires a bit of extra effort, but it's well worth trying it.

The main benefit is the fact that it gives you more flexibility with spacing - you can fit more complex (and smaller) elements within the same space and have them all still fit the grid.

Many popular screen resolutions are also divisible by 8, which can be helpful when designing for mobile devices.

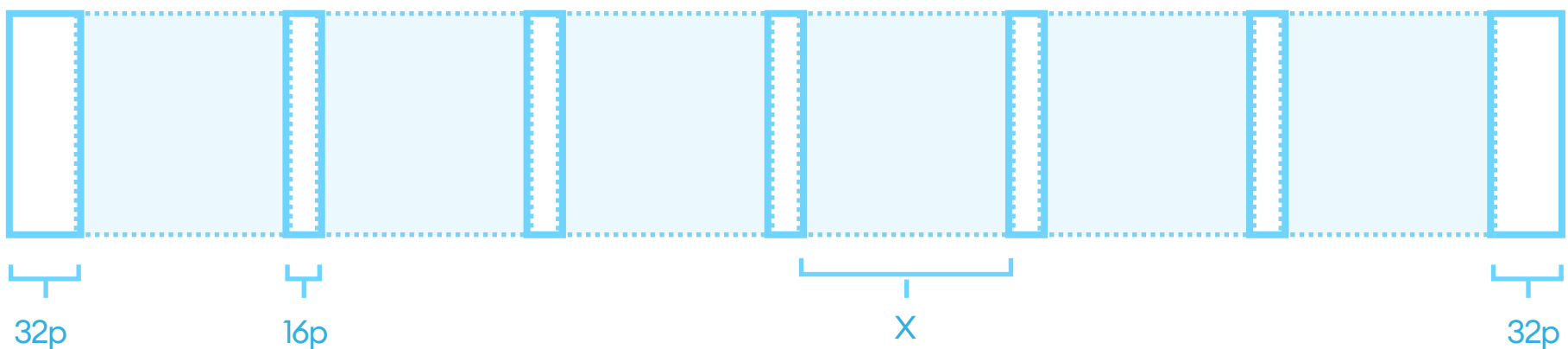


As you can see in the example above, we can place more elements within the same width.

8 POINT FLUID GRID

Creating an 8-point fluid grid should start with multiplying the base number by two. Eight in itself is too small to be the right choice for separating objects. Having them so close to each other negatively impacts readability, as you can't easily discern where one block ends, and another begins—knowing that you can start by setting the gutter at 16 and outer margins at 32 (or 24) with fluid column width.

If we're always multiplying the base number by two, why not set it at 16 in the first place? While the double-base-number works well for the overall layout, the smaller value of eight can come in handy when positioning elements within individual components (like the text on a button).

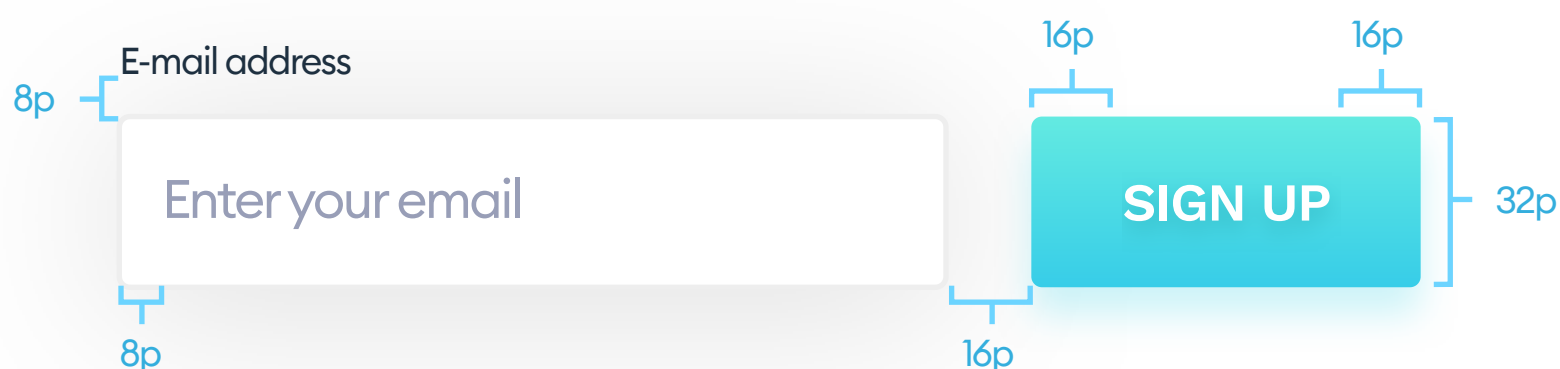


The X value comes from the screen width left when we subtract our margins and gutters from it. In the case of 6 columns and a 390p screen width, we end up with a 41 point column.

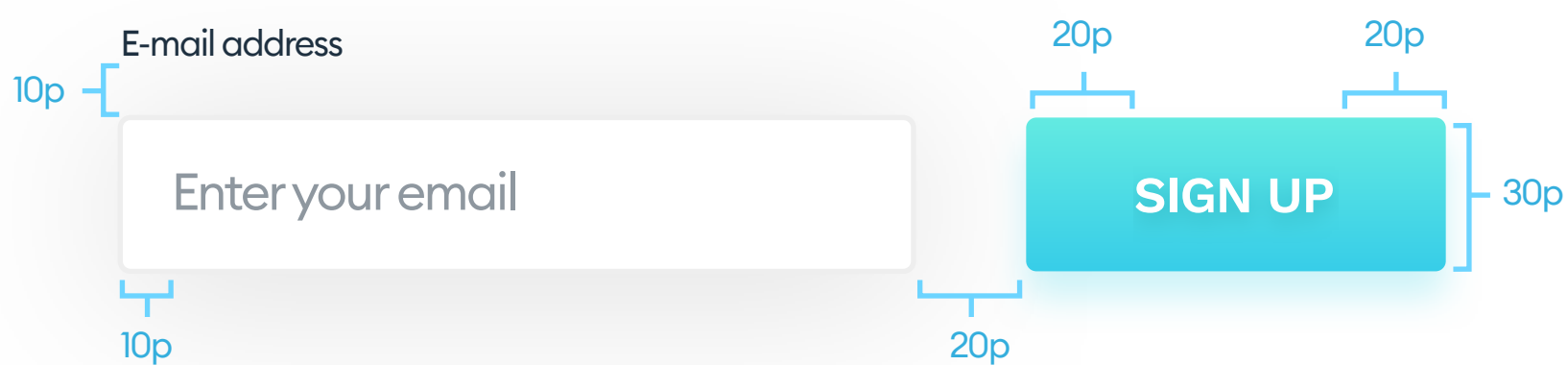
$$390 = 2 \times 32 + 5 \times 16 + 6 \times 41$$

WHY 8 IS BETTER THAN 10?

Why are so many skilled designers choosing the 8-point grid instead of the 10-point? One of the primary reasons is the fact that if we use the multiples of 8, it allows us to have more freedom with setting both external and internal margins in an object. The result is much nicer looking components.



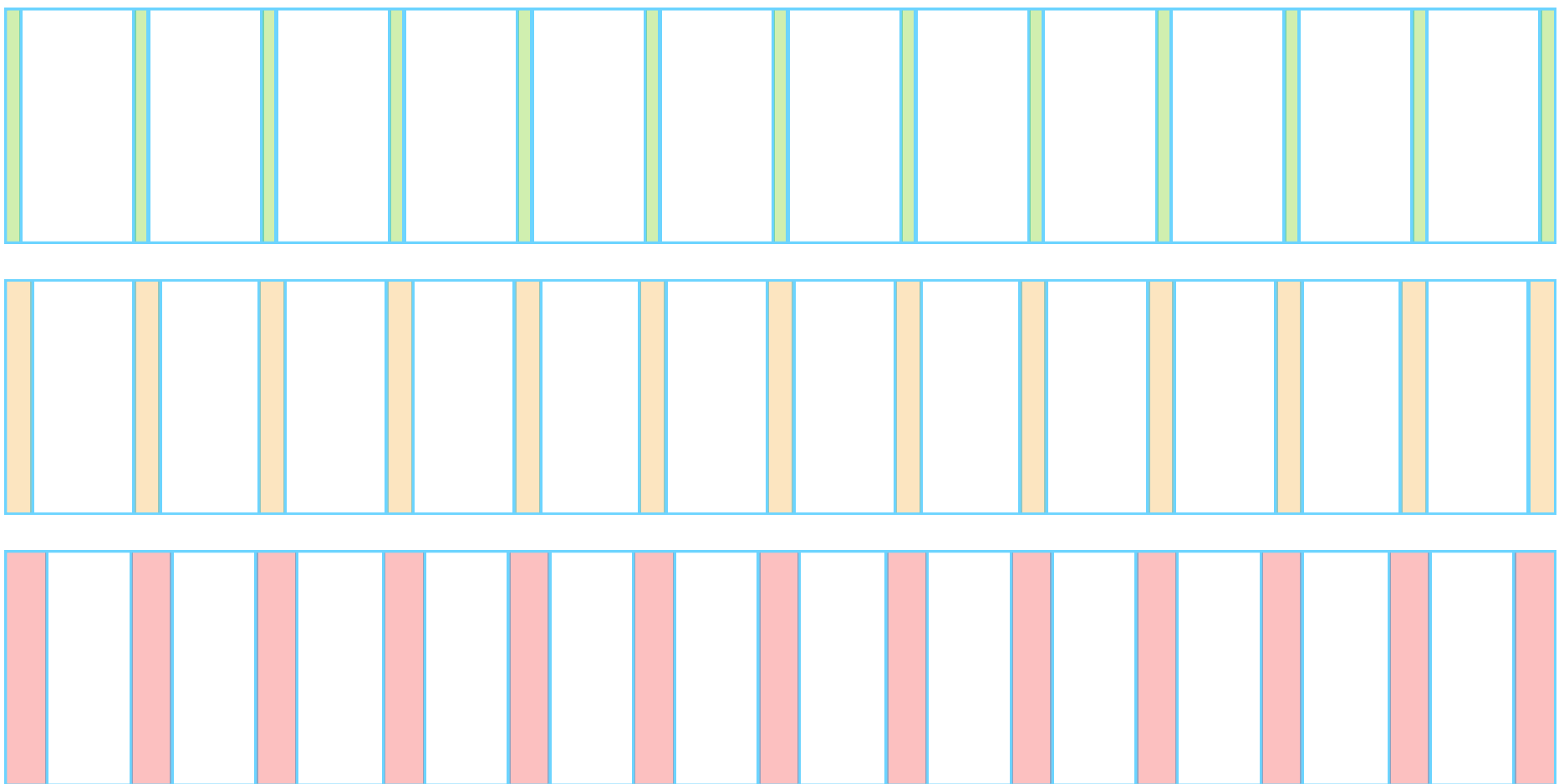
The same component built using the 10-point grid would look like this:



When our minimum nudge value equals ten, it forces us to have much larger margins (of 20). In some edge-cases, it can even reduce readability, or require using additional numbers (like 15) to avoid objects being too far apart or too wide.

MULTIPLE GRIDS

Using a single, well-planned grid is more than enough for most projects. For complex portals, you can consider using multiple grid types. It can make sense when you have both content-heavy screens (like blog posts) and very dense dashboards with various complex components spanning both the X and Y-axis.



When building multiple grids give them easy to understand names, so the developers will have it easier to define them in code.

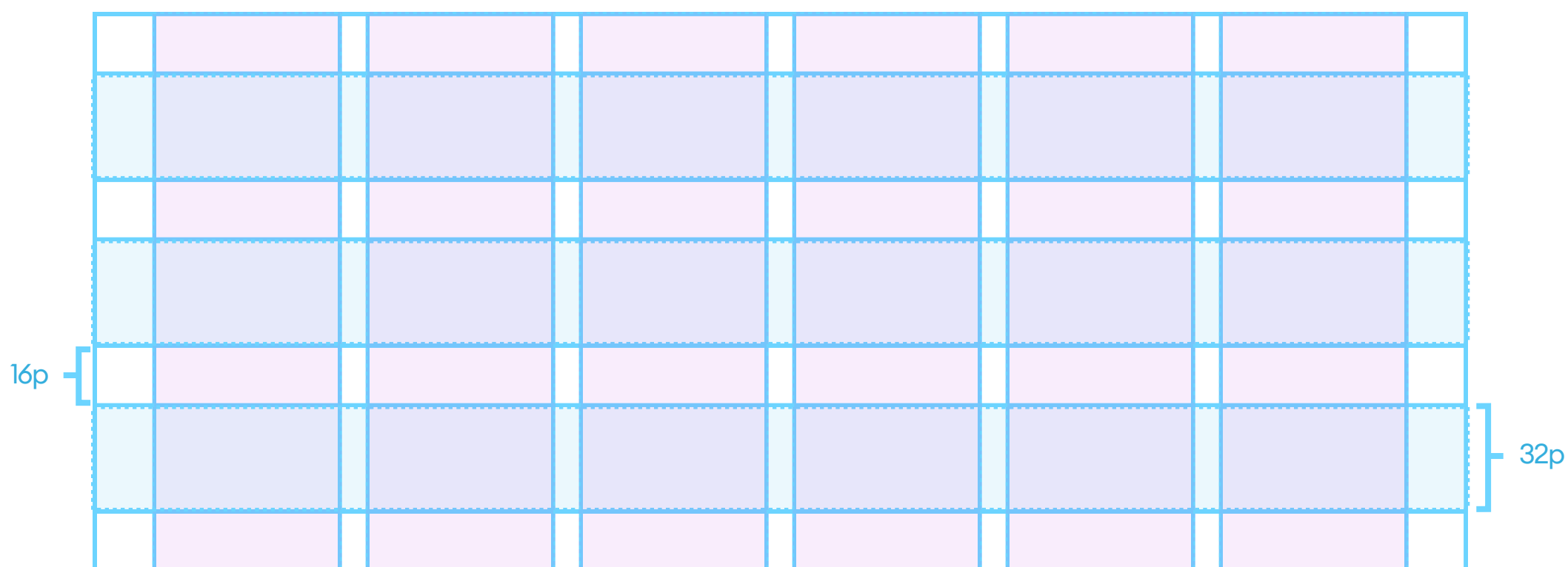
FLUID GRID OPTIONS

To create a set of fluid grids, you define three (or more) gutter values, while still keeping the columns proportional in each one. Keep in mind to have all of them divisible by your top-level base-value (8).

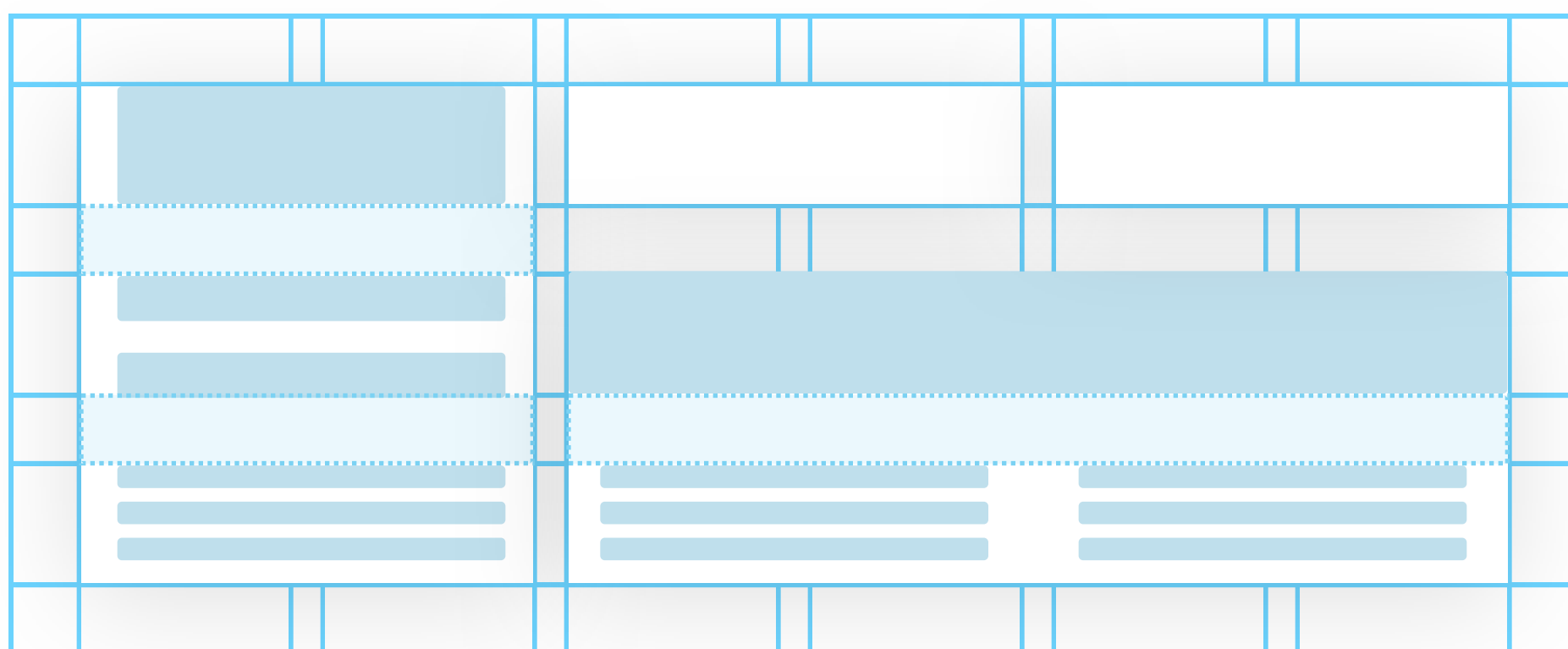
An example gutter scale can equal 8, 16 and 32. Doing this can help with more complex dashboard screens while keeping the content pages more readable. It is, however, a more sophisticated way of handling the grid, and you should avoid it unless it's necessary.

VERTICAL RHYTHM

The vertical grid usually comes after the horizontal one is already thoughtfully tested. You can use the same base number to set it. The main difference here is that it's best to pick the column height as well as gutter height. One example is to have a 16/32 gutter column/combination or 16/16, which can help with more text-heavy designs.



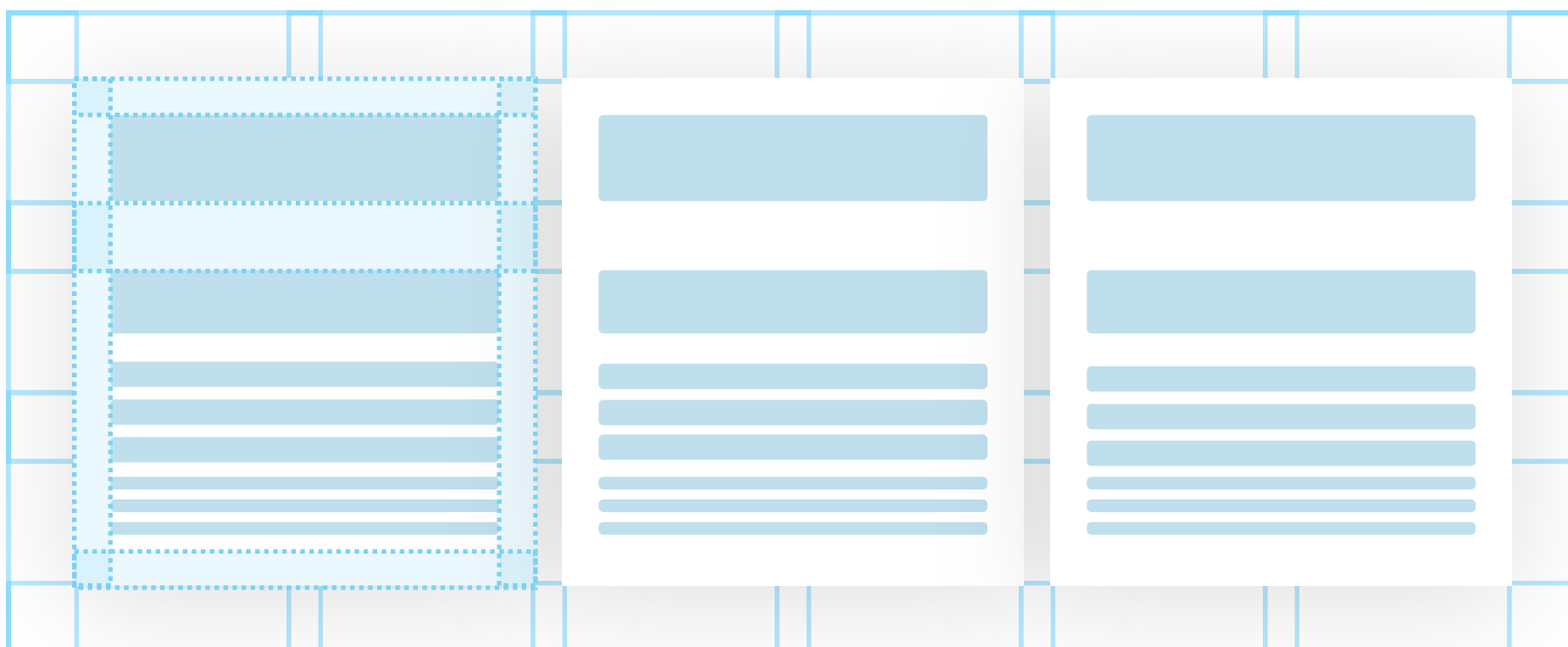
Example of a vertical grid using a 16p gutter and a 32p column.



If you align (most) of your objects to both the horizontal and vertical grid, they instantly become more readable and easier to process.

COMPONENT

A component is a group of objects that belong to the same category. An excellent example of a component is a menu-bar, a product card, or even an individual button. Components should fit our overall layout grid, but they should also follow the same rules inside the actual object.



The main difference between the general layout and an individual component is that you can use smaller grid values inside a component. If you're using 16 and 32 as your main grid values, you can easily use 8 and 16 (or even 4) inside a component.

This margin gradation will help with setting the hierarchy - as we automatically assume objects close to each other are a group.

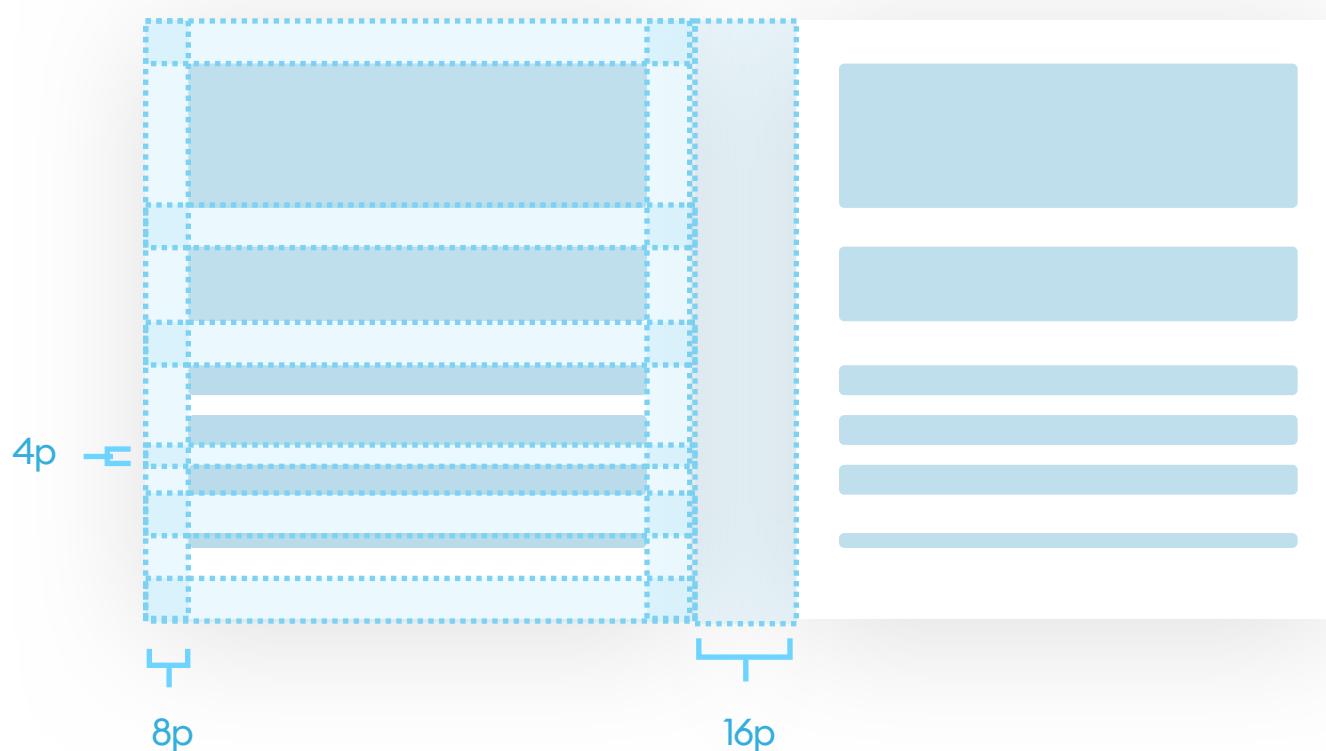
“

Soft grid helps with setting a hierarchy between the overall layout and individual components.”

SOFT GRID

A soft grid is a rule of aligning objects using the base number and its multiplies. It means that both outer margins and all the paddings (in layout and components) are divisible by the base number. It's a more natural way to both design and code digital products, as implementing a hard, fixed grid is pretty complicated.

While it's great to know and understand fixed and fluid grids, in many projects (especially mobile apps), you end up using the soft-grid more often.



Elements that belong inside one group (component) will be closer to each other, while top-level objects (components set in a layout) will be farther apart.

To simplify the soft-grid terminology, we came up with the Red Square Method outlined in the next pages.

The Red Square Method

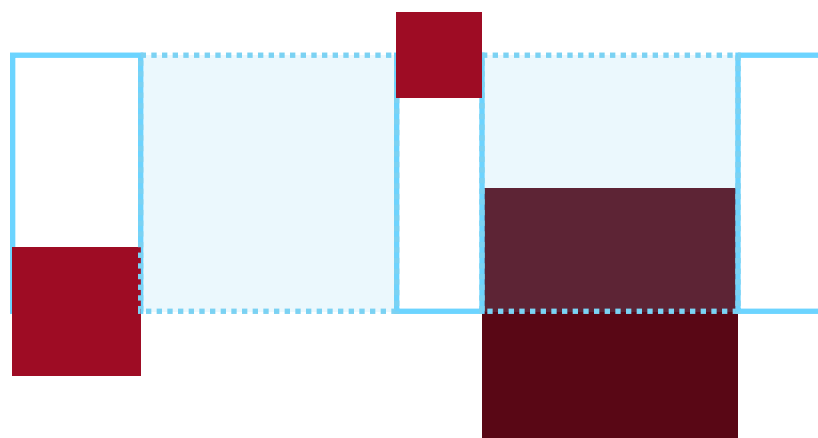
THE BASICS

During our years of both designing and auditing digital products, we outlined a couple of most common grid-related mistakes. The necessity to communicate these problems to stakeholders (usually non-designers) led to us creating a simple alignment testing method. It's a slightly modified soft-grid approach that yields excellent results - especially in audits.



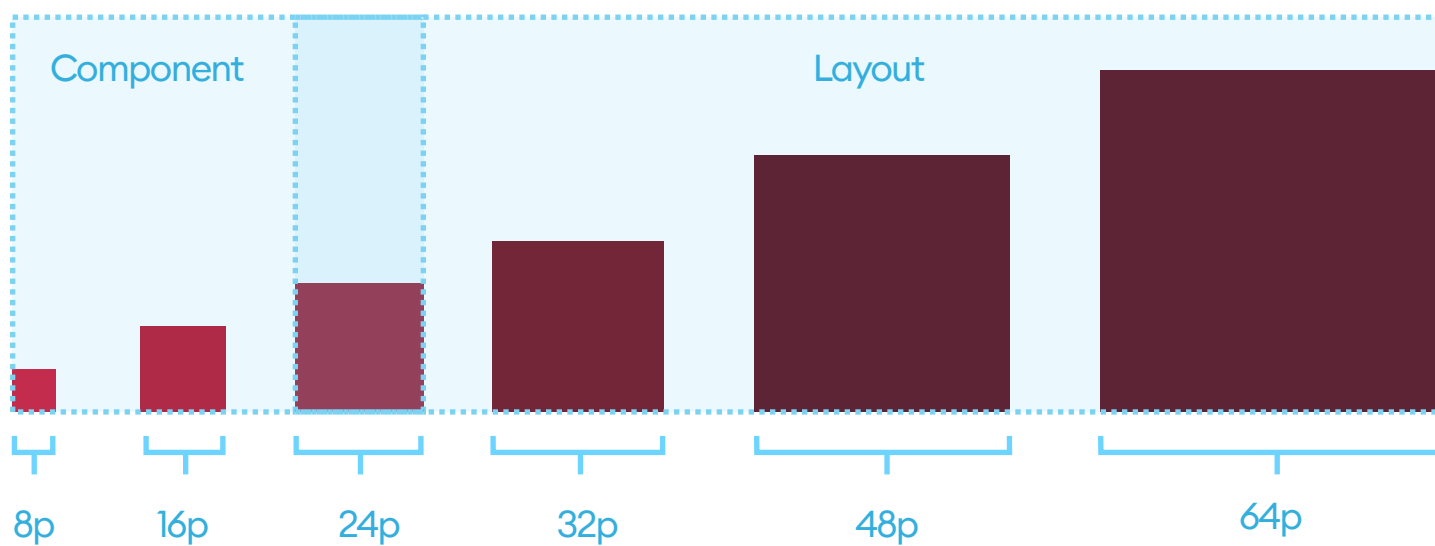
We called it the Red Square Method. Why? A Square is a full shape that's easy to process by anyone, while red makes it look important.

Gridlines can be a massive help while designing, but the main problem with them comes from the assumption that they exist between the actual pixels of our grid. While in fact, they take up at least one pixel to show the line itself. A Red Square is a full shape that you can place between the lines of any grid to check alignment.

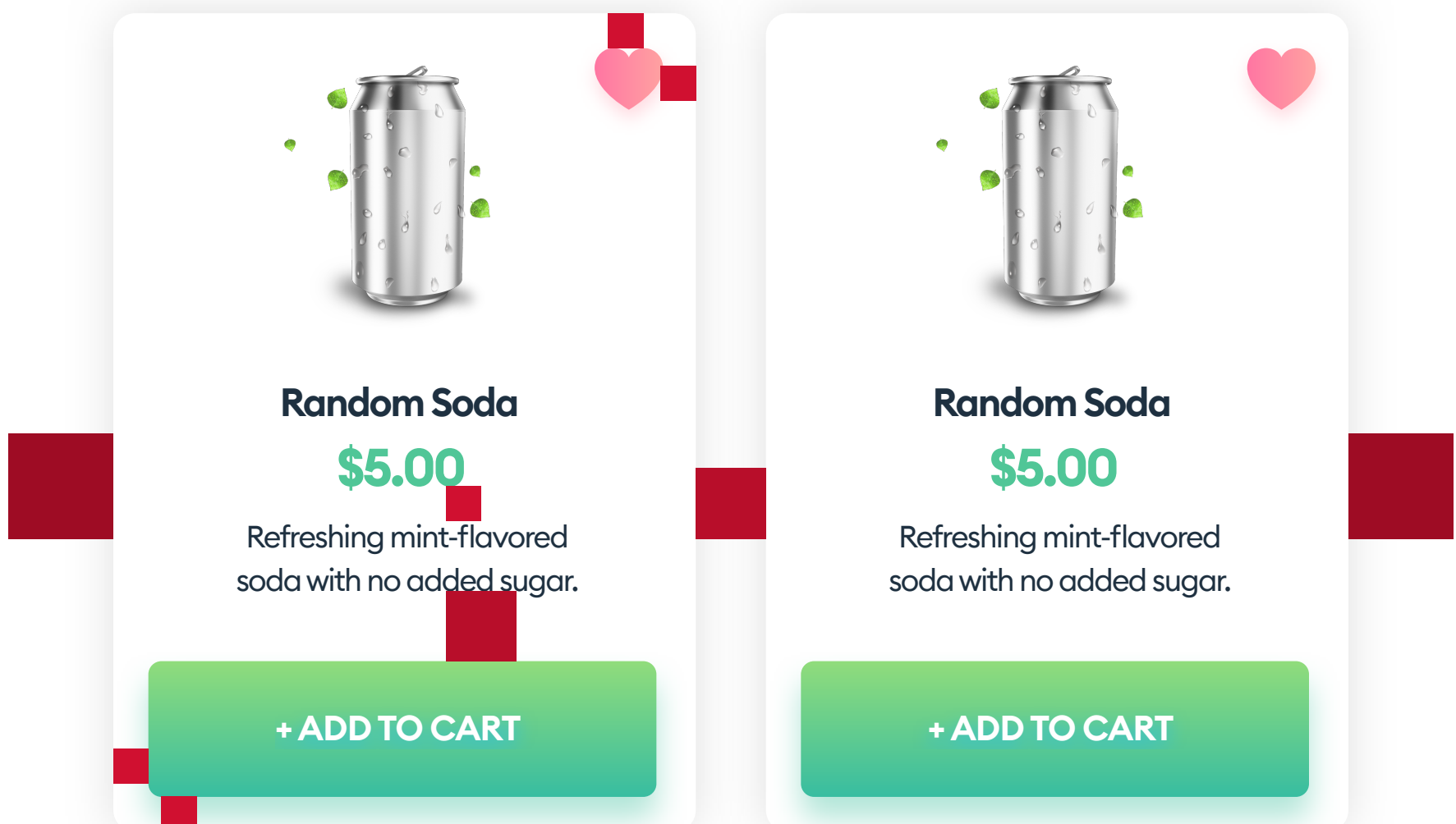


THE RED SQUARE METHOD

Just like with a standard grid, the first step is choosing the right base number. Then we create a red square with the side equal to that number. Then we create more squares by either multiplying (by two) or adding the base number to itself. Starting with an eight-base, we end up with the setup below:



The first two (smallest) values should be for components only. The third one will work well for both layout and components, while everything larger than that should be used for layout only. In general, the third and fourth values are great for setting the space between actual components.

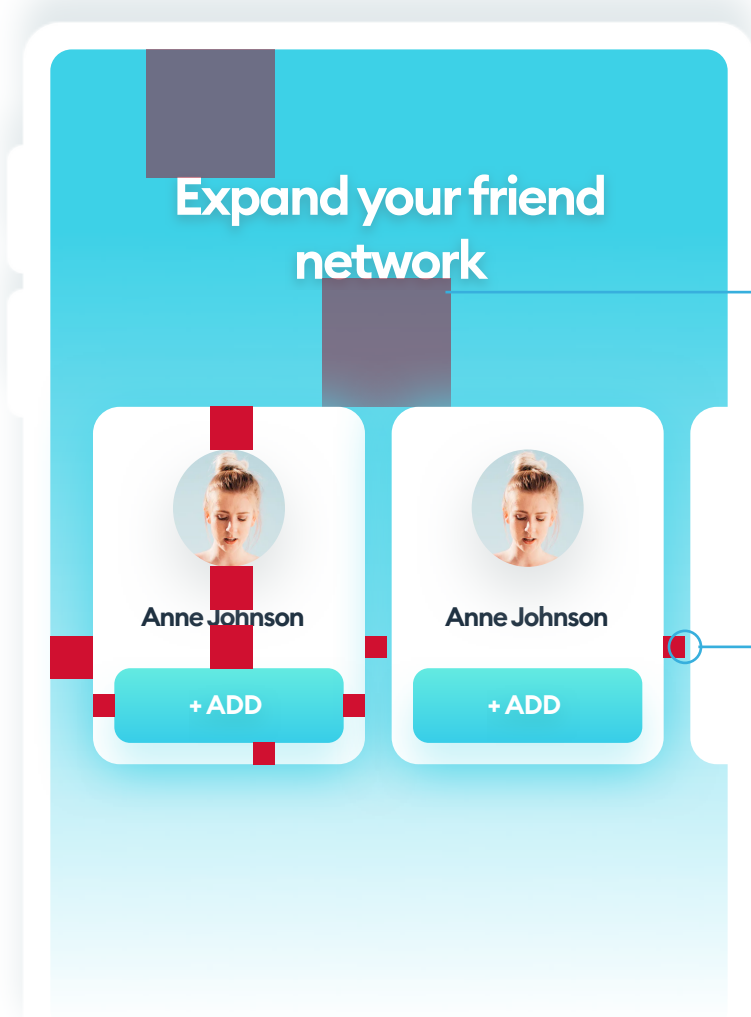


THE RED SQUARE METHOD

Every multiple of our base-number should use a different shade of red (you can use either brightness or hue to differentiate). You can also use different colors for separate values, only starting with red for the base value. Whatever method you choose, the sizes must be visible right away.



If we're auditing an existing design, we can use our squares to determine the distances between objects. Find a square size that fits in most spaces and use it as your base number. What you should be looking for is deviations from the rules - small differences in set spacings and things positioned out of place.

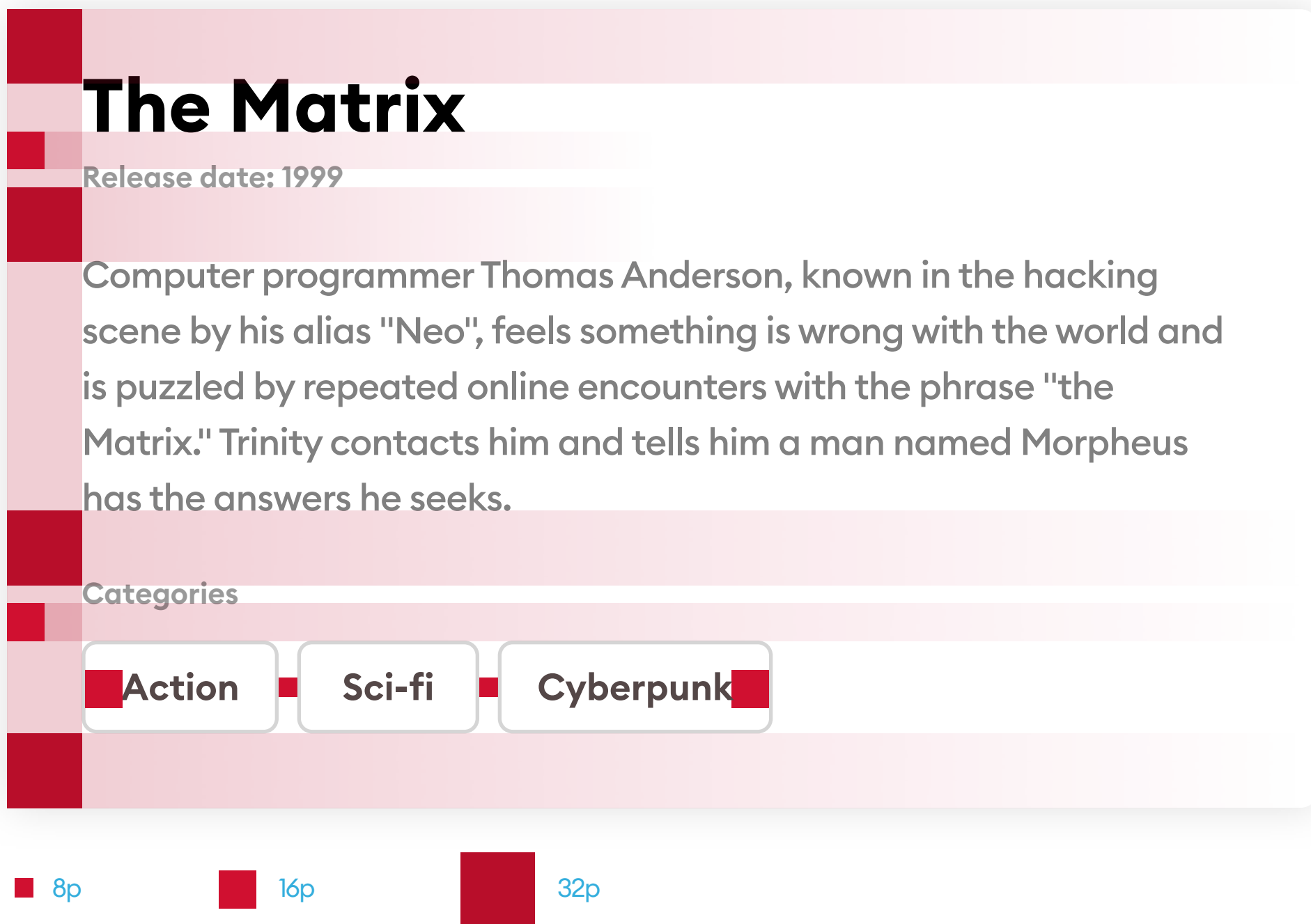


Text should be tested against its baseline.

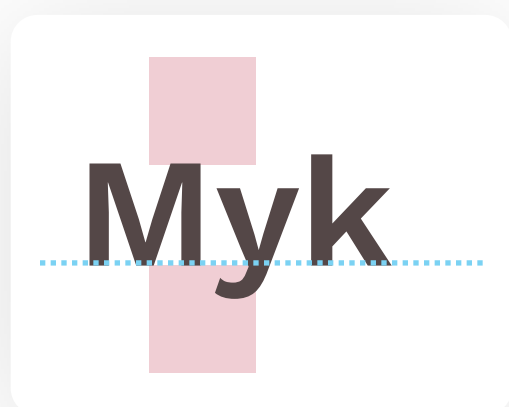
As you can see the distance here is too big (by 1 point) - aligning our smallest square clearly shows the problem in this example.

TEXT ALIGNMENT

The Red Square Method is one of the easiest ways to align blocks of text for proper hierarchy and higher readability. There are more methods for good alignment (including purely optical ones), but this one works great, especially when you're just starting your design career.



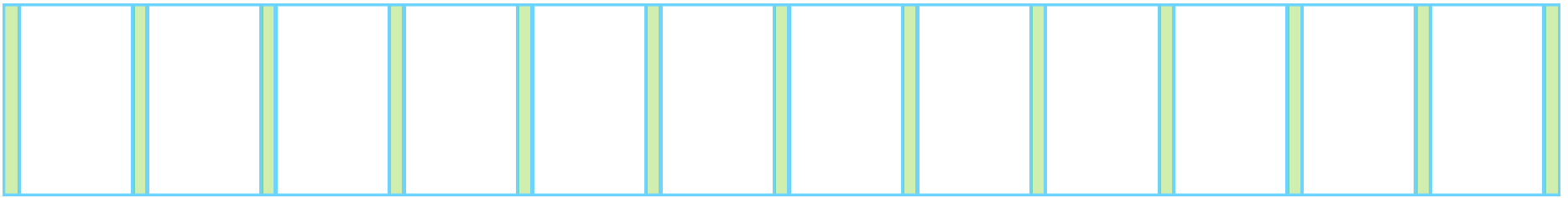
As you can see in the example above, we're using only three sizes (8,16, and 32) to align all of our content (including category buttons) and give it a clear hierarchy.



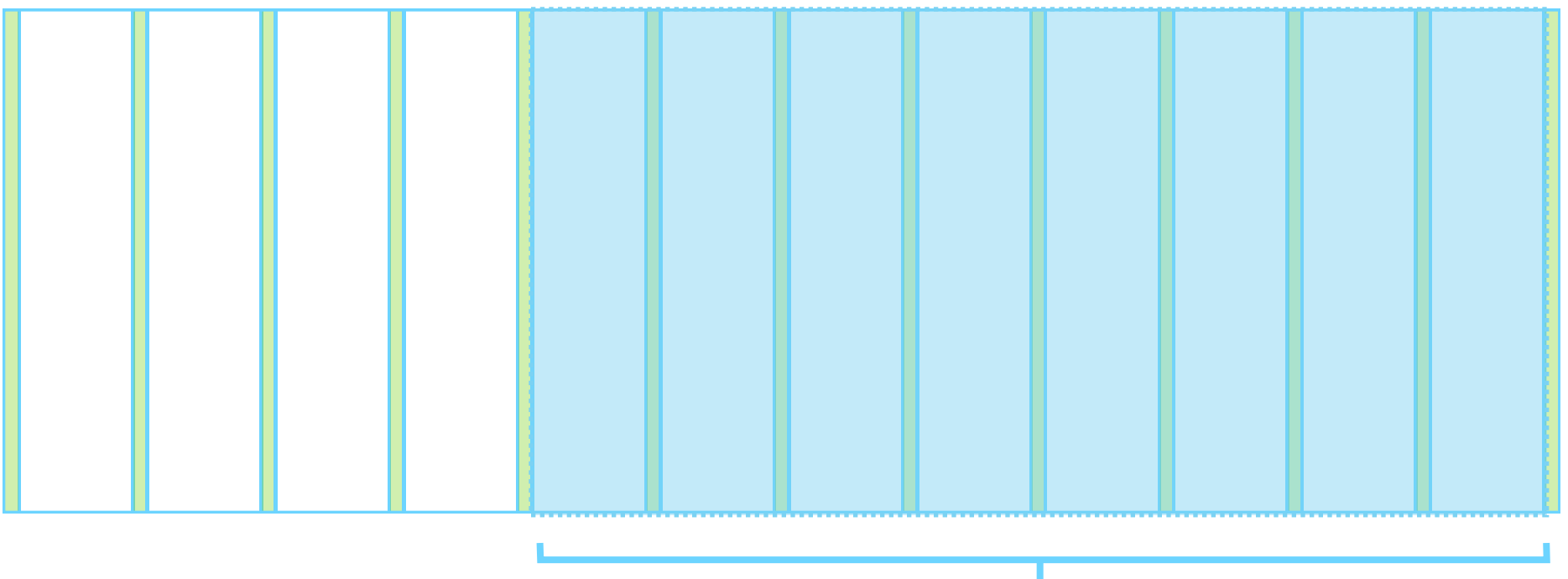
We are aligning our squares to the CAP-height (see Typography), which also means they align to the baselines of our font. If you use this rule consistently, you'll end up with clear, readable content. For some fonts it's also ok to align to x-height (lowercase x).

GOOD PRACTICES

Layouts should be simple. In web design projects start with either 12 or 16 columns for greater flexibility.



Longer text (body copy) on a 12-column grid should not exceed eight columns. That roughly translates to about ten words per line and makes it more readable and more comfortable to scan.



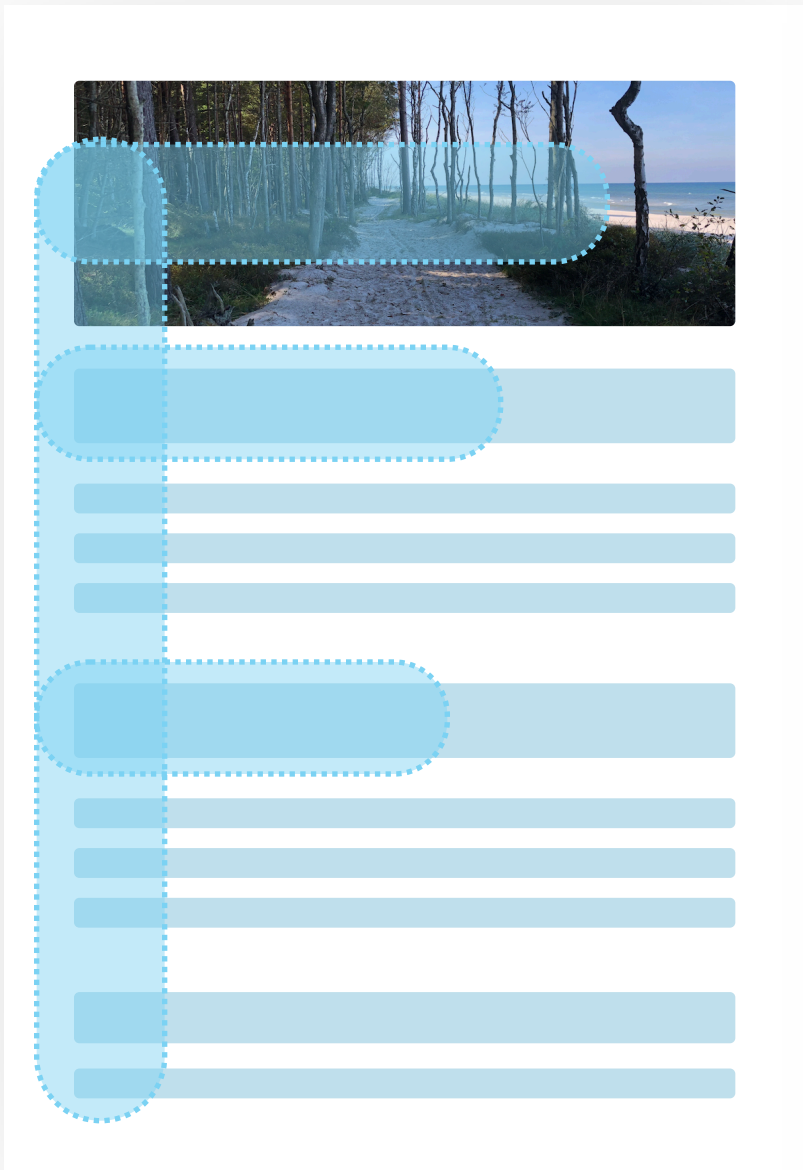
Try not to stretch longer text to the full width of the page. You can use the four extra columns here for navigation or suggested content.

DEFINE THE GRID ASAP!

Once you get some initial component styles nailed down, define the grid, and adjust them to fit. While it slows down the pace in the initial design stages, it helps with greater consistency, fewer bugs, and frictionless development in the long run.

HOW WE SCAN CONTENT

We don't read anymore. We scan. There are two main patterns that our eyes follow while looking at content. Knowing them will help you plan your layout for maximum effect.



F-PATTERN

The most popular across Europe and the US is the F-pattern. It shows that our eye follows the left edge of the content, and jumps to the right through photos and headings only.

Whenever possible, when designing for the left-to-right readers assume they'll be using an F-pattern.



Z-PATTERN

Z-pattern happens when a large photo or video breaks an F-pattern. We jump back to the left diagonally, right through that element.



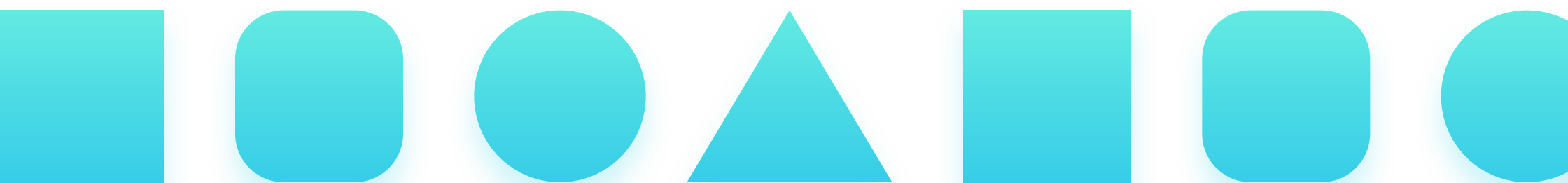
9.

Objects

Objects

THE BASICS

All currently available UI design tools work by manipulating vector shapes. That means the forms themselves are represented by numbers that define them and can be stretched and enlarged without losing quality. Their parameters are just numbers that generate a new shape according to the criteria with each change.



If you're planning to design interfaces, be prepared that most of the time, it will be about moving various rectangular objects around. Rectangles are the most popular shape (right before ellipses) in all the interfaces.

The general idea of UI design is about moving rectangles around in just the right way.



This chapter covers the very basics of all the UI design ingredients.

“

**UI Design
is moving rectangles
around.”**

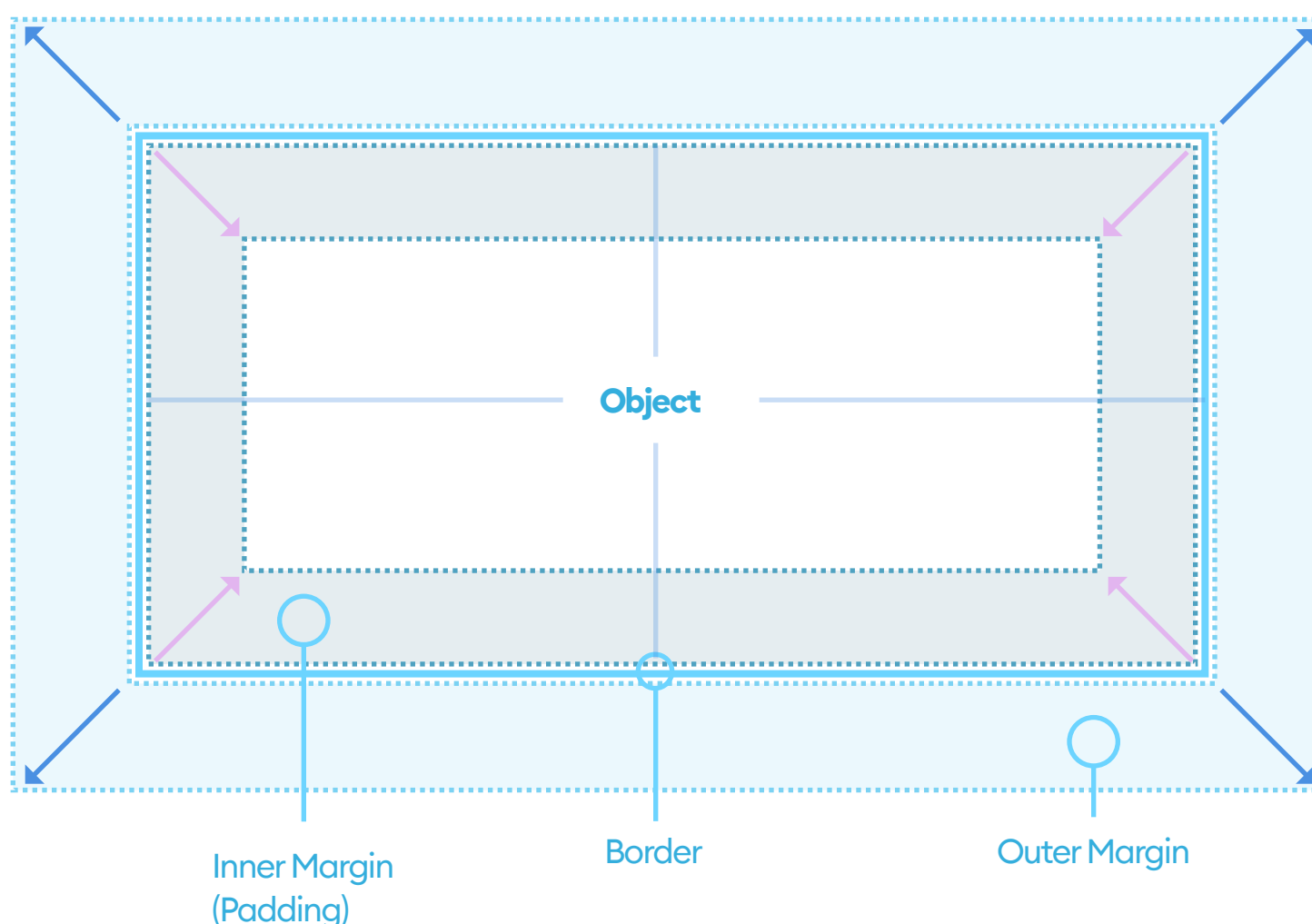
THE BOX MODEL

The Box Model is the most basic way of defining a digital interface object both in design and in code. Most of the design we do works according to the box model, so it's essential to get to know it well. It consists of four main elements:

- ✓ Fill
- ✓ Border
- ✓ Outer margin
- ✓ Inner margin

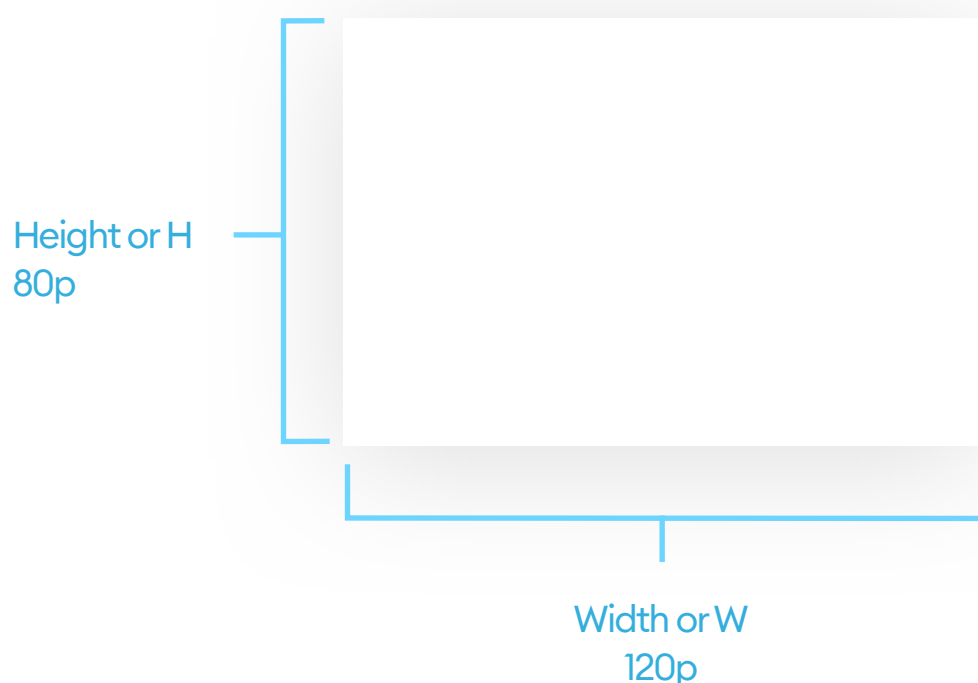
The inner margin is also called padding. It means the more extensive it is, the bigger the safe area inside the object becomes. In the example below, it's the grey area inside the rectangle.

Outer margin (often just called "margin") is the area outside our object. It allows us to position it with enough safe space concerning everything around it. It's the blue area outside the rectangle below.

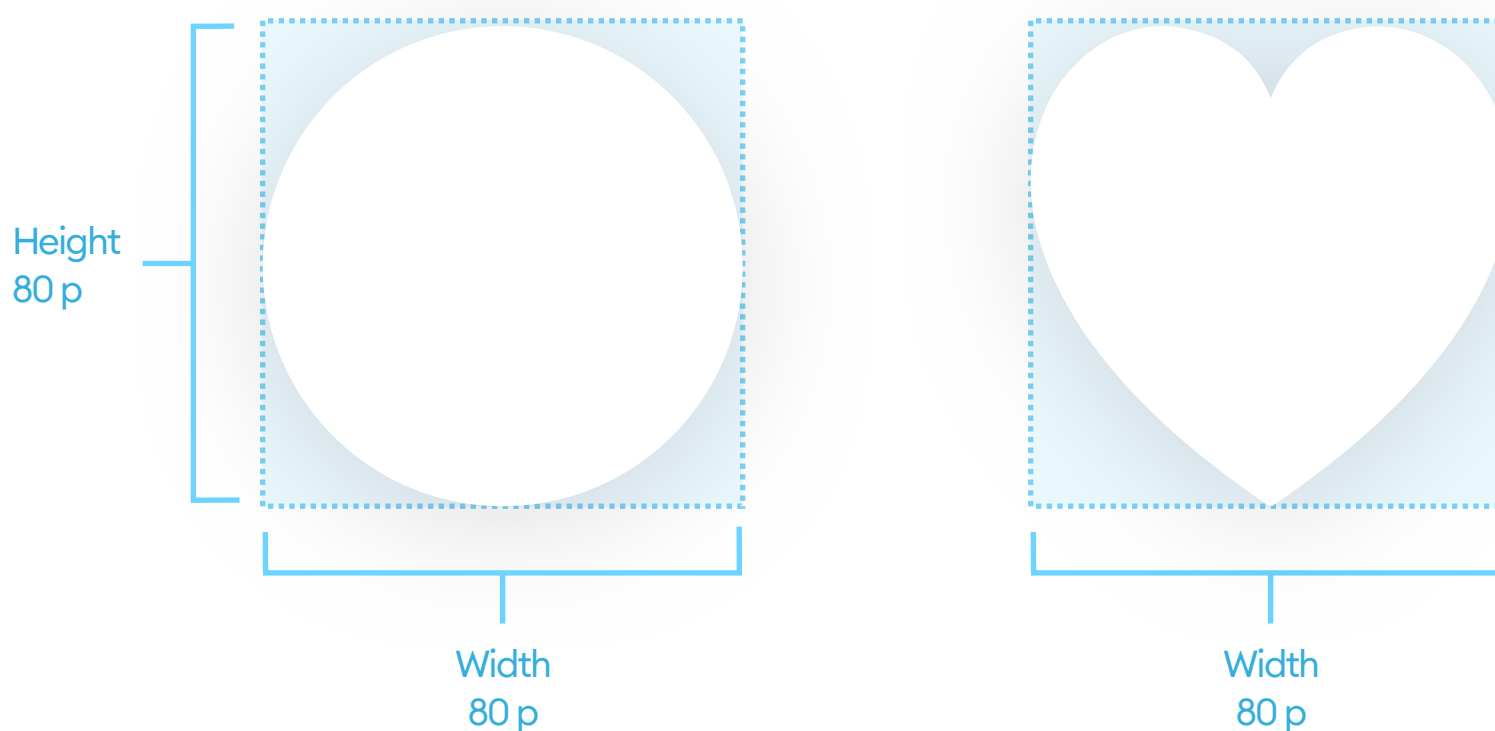


SIZE

Width and height (in points) define the size of an object. In most cases, we use W and H for short. As it's a two-dimensional space, the width is a scale on the horizontal axis, while the height is a scale on a vertical one.

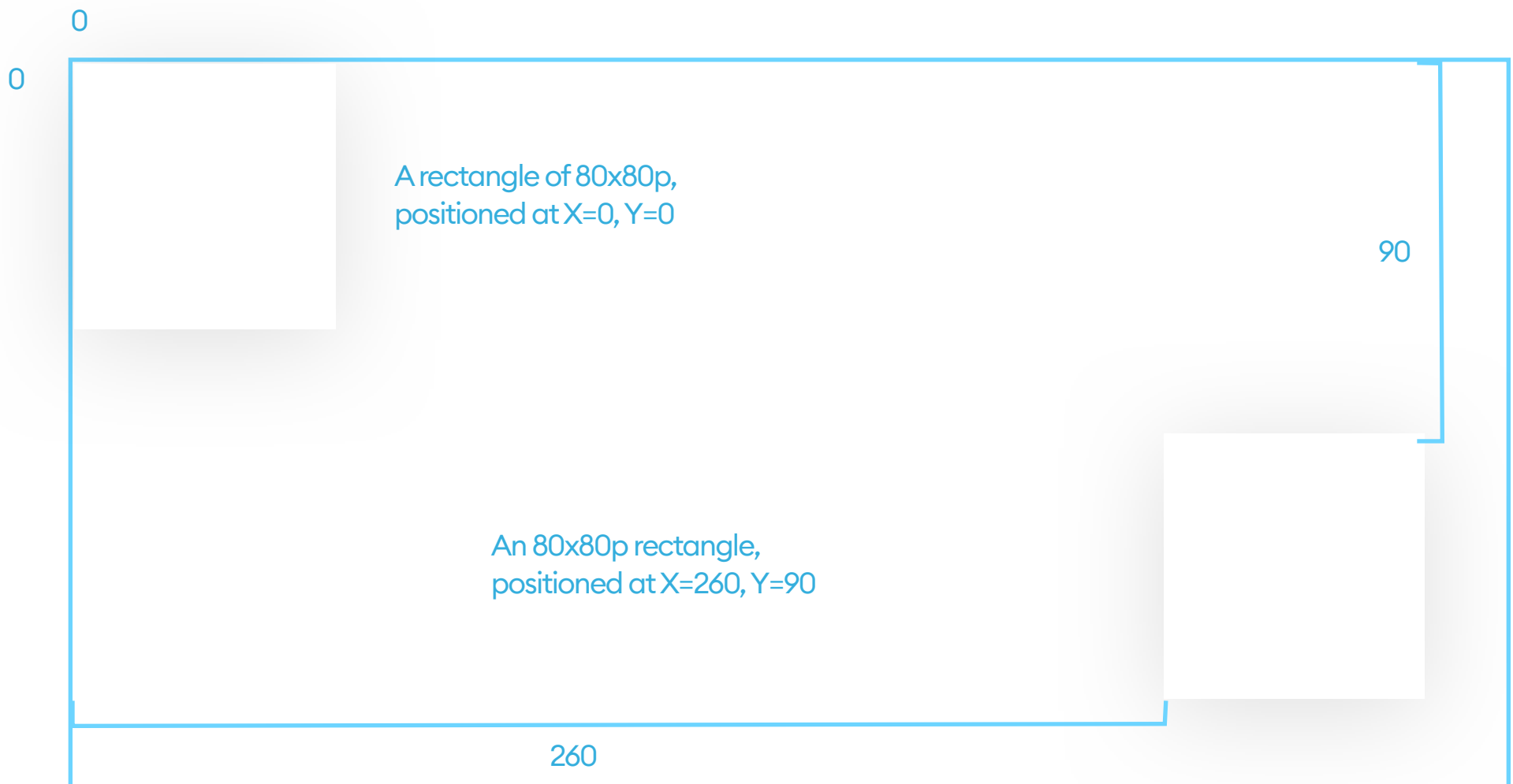


The width and height are always of the smallest possible rectangular box that can contain our object - even if the object itself has an irregular shape.

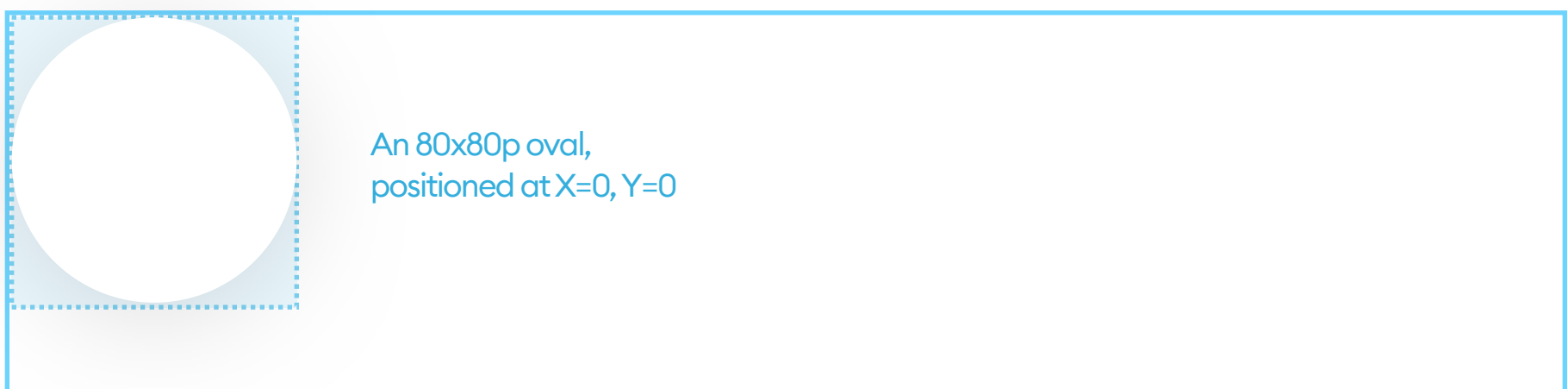


POSITION

The position of the object is a set of numbers on the X and Y axis, defined by the artboard that contains it. The X is the position on the horizontal axis, while the Y is the position on the vertical one.

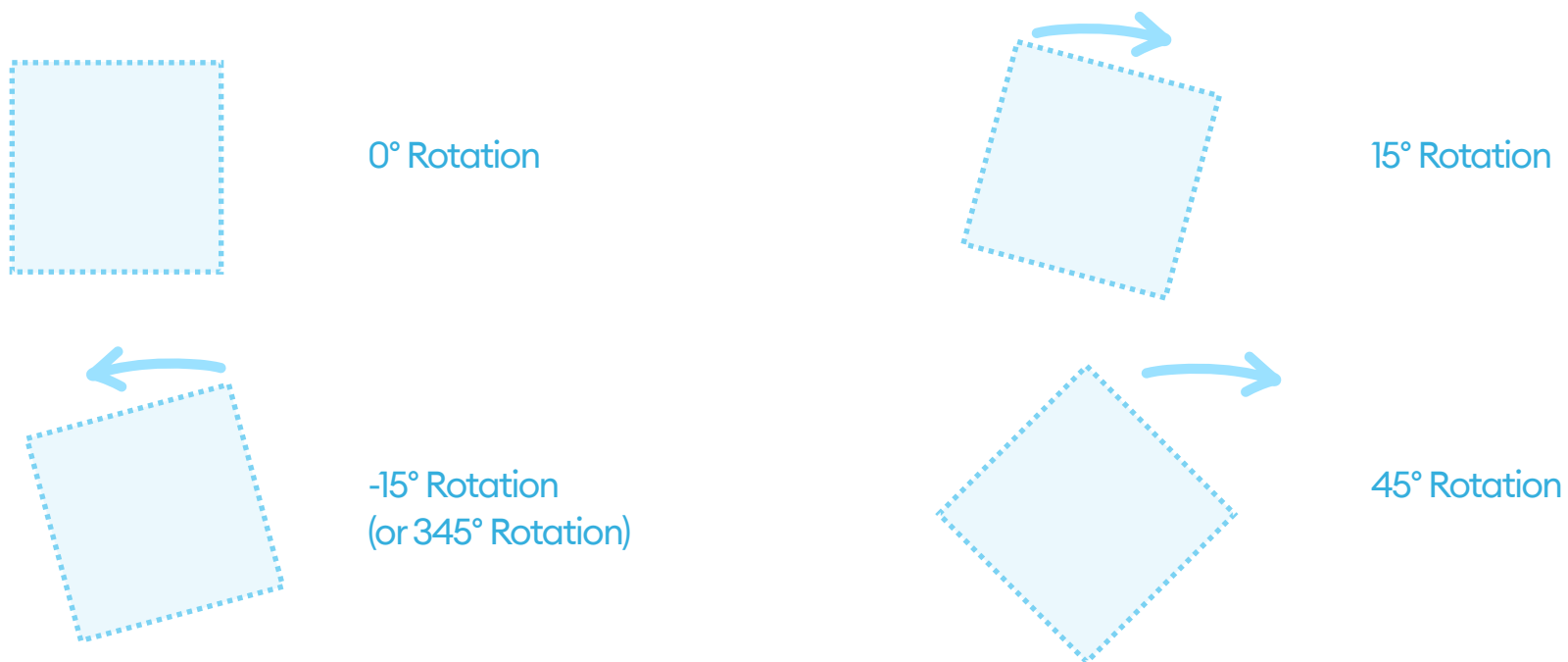


Irregular objects positions are the position of their bounding box that contains the actual shape within.



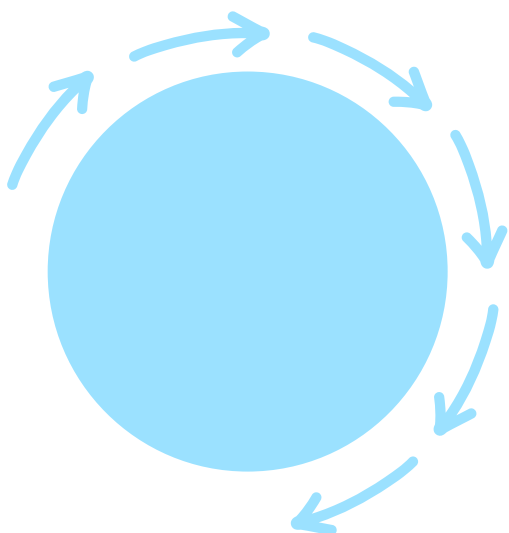
ANGLE

Angle defines the rotation of our object going clockwise - the default being 0° . Rotation can also be a negative number. It's worth remembering that a -15° rotation is the same as $360^\circ - 15^\circ$, which equals 345° .



If we want to have a consistent interface, it's best not to rotate objects by hand, but instead enter the right angle from the keyboard as it is way more precise than manually turning the object.

If we flatten the object (it's an option you have in most design tools), it resets the angle to 0, but it is still rotated - only the new perspective becomes the base for further rotation. Flattening also converts rounded corners into a rounded path so only do it if you really have to!

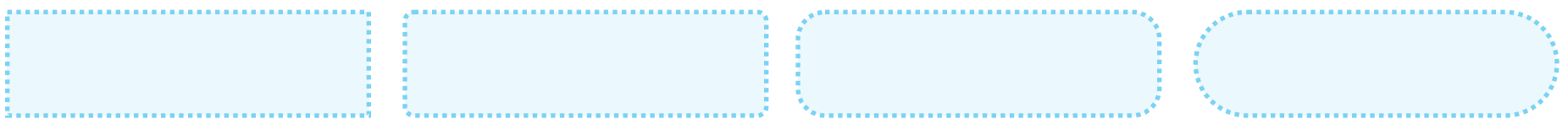


A perfect circle will look the same no matter of its angle.

BORDER-RADIUS

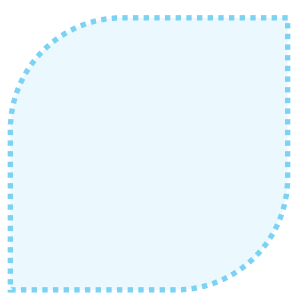
A border-radius is a number. Just like the width and height, it's also presented in points. The larger that number, the larger the roundness of the corners of our shape. You can use it for all of the edges at once or set them individually.

In general, even slightly rounded corners (between 2p and 6p) are considered more friendly than sharp ones (of 0p). However, it's good to have consistency in the roundness. The grid base unit that we select should define the border-radius. It should also match its surroundings.

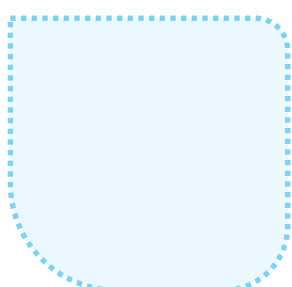


Starting from the left a border-radius of: 0p , 2p , 6p , 100p

LEARN MORE



A square with only two corners rounded.



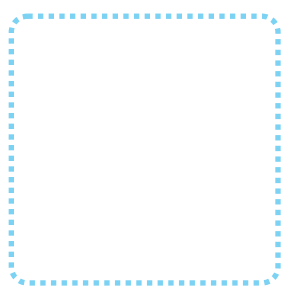
A square with each corner having a different border-radius.

Different border radiuses in the same object can make a great button if it also matches the brand guidelines (i.e., the logo also has a similar pattern).

If you decide to make your buttons that way, remember to be consistent with them across all the screens.

FILL

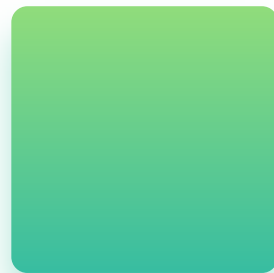
Fill is another name for the object's background. It can be a color, a gradient, or a photo. Each of these can also have a different level of opacity. If an object doesn't have a fill, is also lacking a border or an effect, it won't be visible in the interface as it requires some defining characteristics to be visible. You'll still be able to select it, however, move it around, and modify it. It will also still exist in the layers list.



None



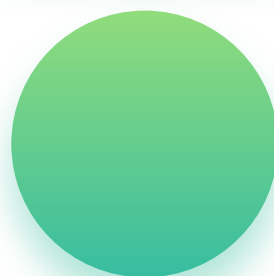
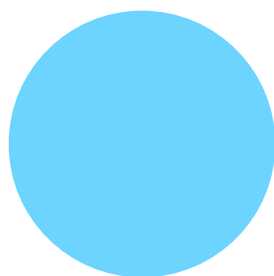
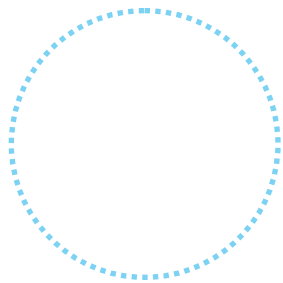
Color



Gradient



Photo



You can combine more than one fill on the same object - for example, adding a transparent color to a photo will colorize the result so it can match the rest of your UI.



None



Color



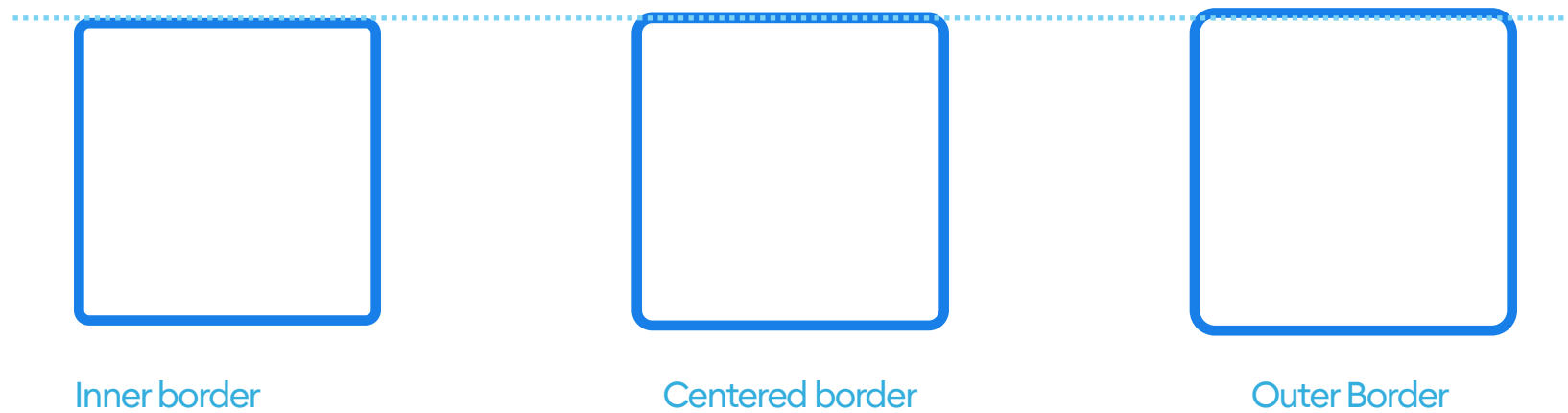
Gradient



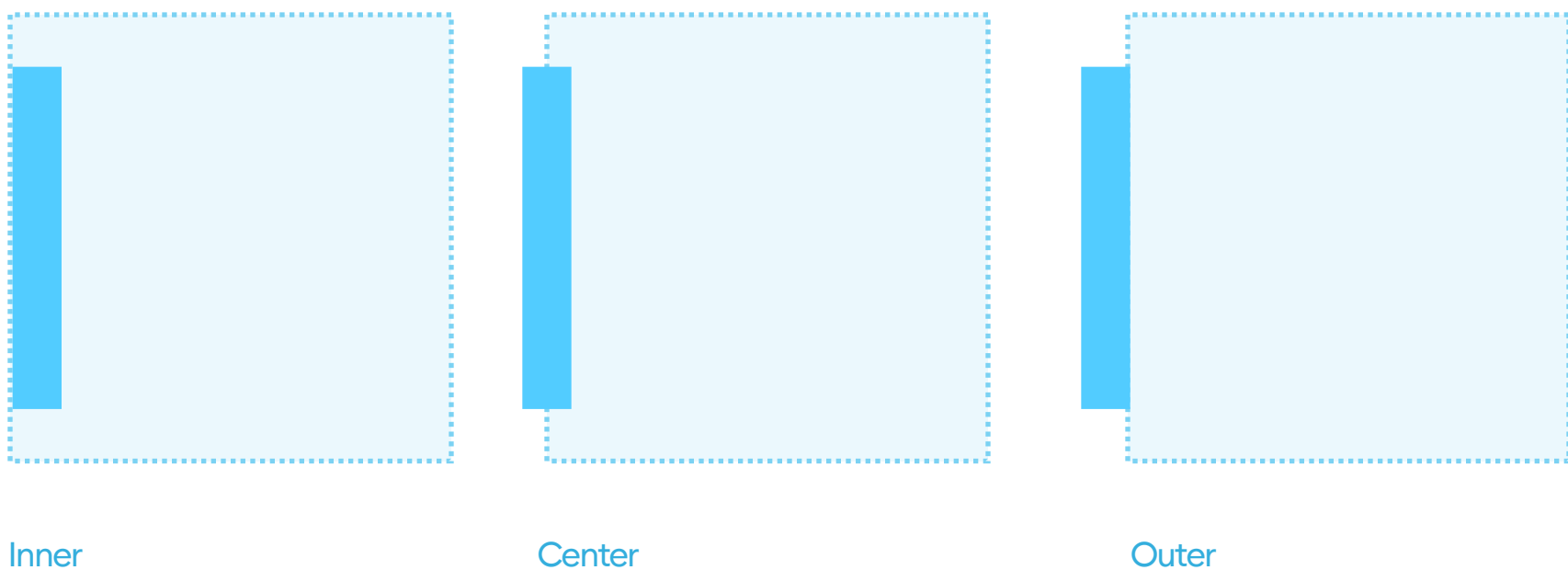
Fade-out
Gradient

BORDER

The border is a line that goes around our object. It can go inside (inner border), outside (outer border), or between them (center). Remember that only the inner border doesn't visually make the object larger.



Above, you can see a 60p sized square with a 2p border around it. From the left: Inner-border, centered, and outer. As you can see, only the first one is still that 60p size visually.



A 2p border width that's centered adds 1 point inside and one outside of our element. That makes our square 62x62. The outer border adds 2 points to each side, which makes the square 64x64.

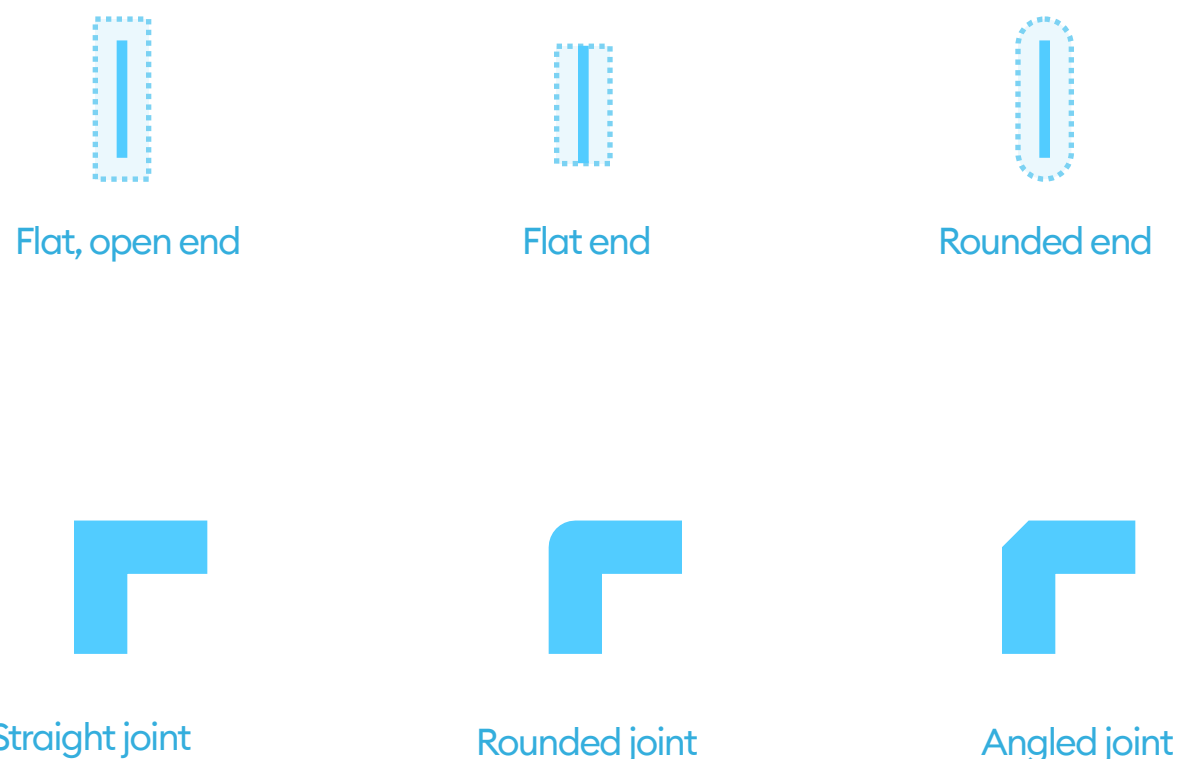
BORDER

The border can have different weights (width) in points and can be a dashed or dotted line. We can also fill it with both a color and a gradient.



Most of the design tools allow you also to modify the endings and joints of your lines.

The ends can be open, flat, or rounded while the joints can be sharp, rounded, or angled. The joints and the endings are not connected in any way, so you can have sharp joints and rounded endings in your shapes. For consistency, however it's best to have them match.



DROP SHADOW

Outer shadows (or drop shadows) are easily the most common effect used in UI. A typical shadow relies on an offset from center (x, y, or both) a blur and an opacity value.

Some tools also allow for a "spread" value, which makes the shadow look like a smaller element is casting it. It makes the shadow look more natural as well, as it accounts for the perceived distance of the object from the background.



X=0, Y=2,
Blur= 0, Spread= 0

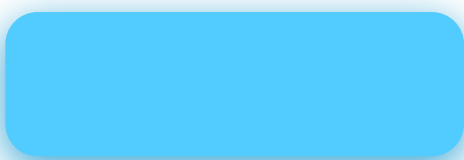


X=2, Y=0,
Blur= 0, Spread= 0

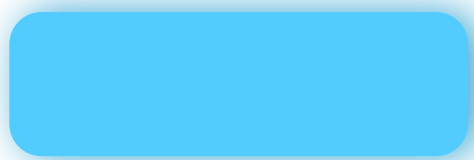


X=2, Y=2,
Blur= 0, Spread= 0

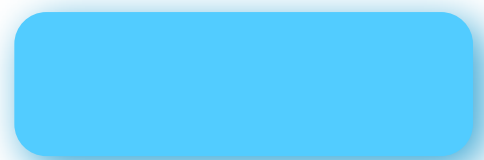
Blurring our shadow makes it more natural. It's essential, however, that the opacity value is never higher than 40%.



X=0, Y=2,
Blur= 8, Spread= 0

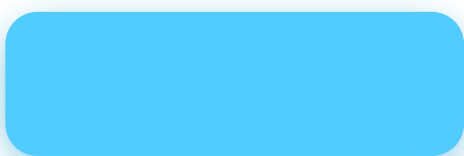


X=2, Y=0,
Blur= 8, Spread= 0

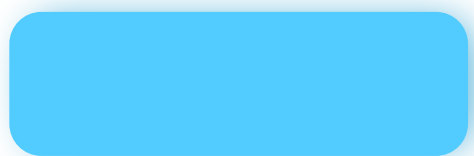


X=2, Y=2,
Blur= 8, Spread= 0

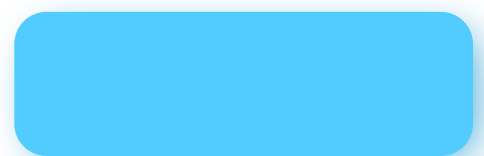
A negative spread value makes our shadow look more natural.



X=0, Y=2,
Blur= 8, Spread= -2



X=2, Y=0,
Blur= 8, Spread= -2



X=2, Y=2,
Blur= 8, Spread= -2

“

Using a drop shadow helps with establishing a hierarchy between our interface layers.”

DROP SHADOW

A natural-looking shadow is one of those elements that make the most impact in design. The most important part of looking natural is avoiding pure black shadows and using a shadow derived from our primary color instead. Pure black makes the contrast ratio too big to look natural. If you study real-life shadows, you'll notice they often vary in shade and tone.



If our primary color is green, then a green button can cast a blurred dark-green shadow. With a big enough blur and a negative spread value, we end up with a good looking result. You can also use a more subtle, grey+base color mix that is not as colorful.



Too sharp, dark shadow creates too much contrast. That high contrast black is often a default value for shadows that we should always modify.



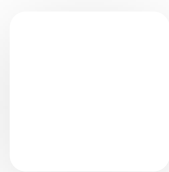
If we change the black color to a darker shade of our primary color, it starts to look more natural and softer already.



With a negative spread value, we eliminate the shadow spillover, making it look more realistic.

INNER SHADOW

Inner shadows are relatively rare in UI. It has the same parameters as a drop shadow, but it appears inside of the object.



Outer shadow



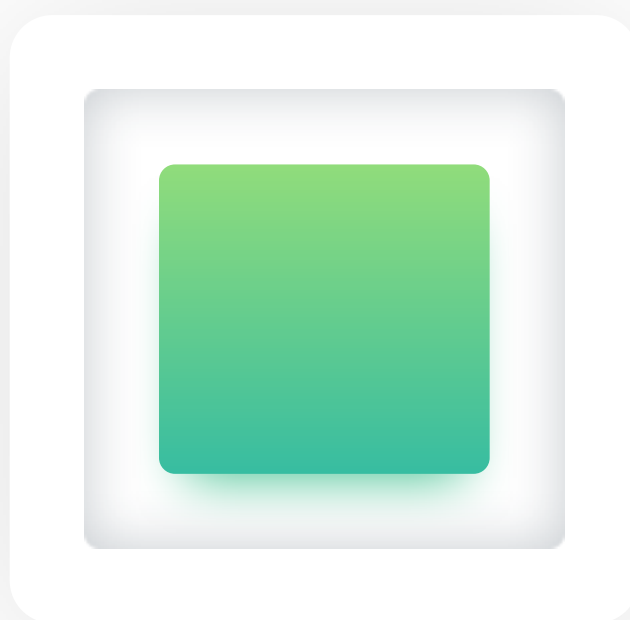
Inner shadow

They are not as popular because most interfaces are a series of layers stacked on top of one another. In that case, an outer shadow makes sense as it provides the depth. An inner shadow would suggest the object has a hole in it.

The only use-cases of this style are form inputs (both form fields and checkboxes or radio-buttons) and extruded shapes in the Neumorphism method. They can be used to make the objects appear more life-like in some instances but should only be used moderately.



Modern UI's are using the shadows to build a visual hierarchy and layered structure of elements.



An inner shadow makes us think the object has a hole in it, which breaks the expected layering structure and can be confusing.

GAUSSIAN BLUR

Most design tools nowadays have a Gaussian type blur that extends the effect in every direction evenly. Its primary value is the radius. The larger it gets, the more prominent the blur effect.

Gaussian blur is the most often used UI effect. You can employ it into transitions between screens, or show a bit of realistic depth of field by selectively blurring the background.



0



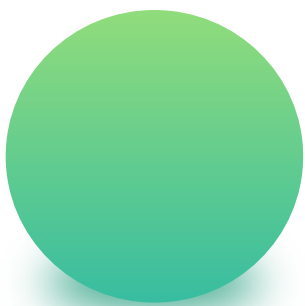
2p



8p



16p



This type of blur can also help you generate non-standard, point-shadows under objects. The easiest way to achieve it is by blurring an ellipse and placing it under the object casting the shadow.

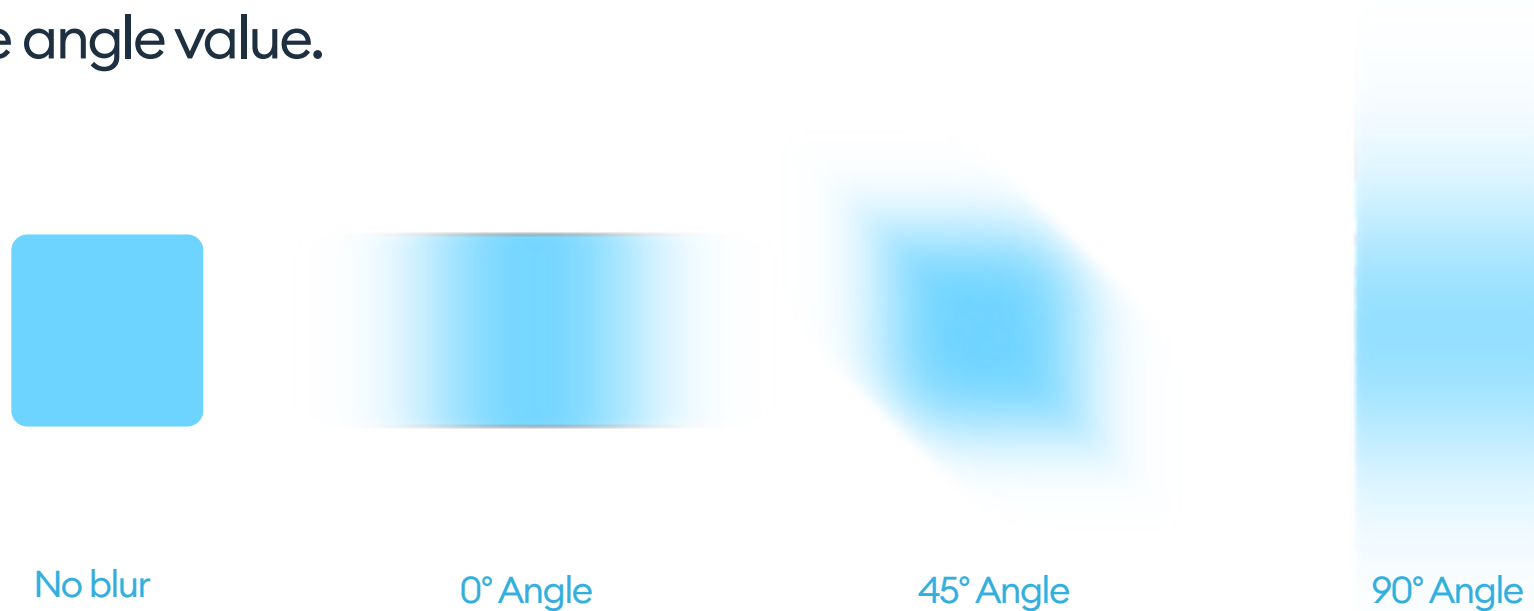
BACKGROUND BLUR

Background blur became popular when Apple started using it in their OS to achieve that behind smoked-glass effect in some screens. An object with this effect applied blurs everything under it. If you want to find out more, check out the Glassmorphism chapter in UI Design Styles.



MOTION BLUR

A motion blur simulates the movement of an object in a direction defined by the angle value.

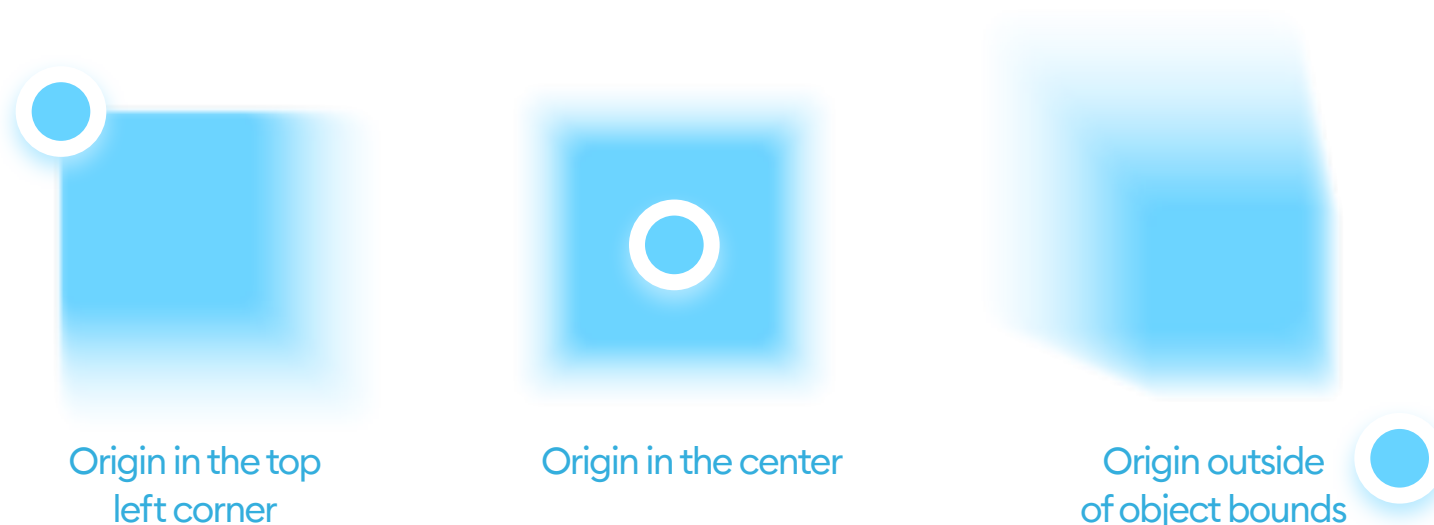


ZOOM BLUR

A zoom blur happens when the object becomes blurry from the inside out. It's often used in photography, but not a great pick for interface design.



In this particular blur type, you can also set the origin of the blur. By moving that point around, you can achieve some interesting effects.



↵
Colors

Colors

10.

Colors

Colors

Colors

Colors

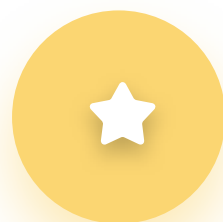
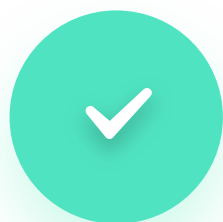
THE BASICS

Choosing the right color palette is one of the most crucial, essential steps in the design process. The hues we use affect the style and mood of the design, create the "look & feel" and are highly memorable.

It's also the reason why people who don't remember the product name refer to it as "that blue app" or "that black and white website."

Colors also have strong ties with our emotions, so we shouldn't choose them based on aesthetics alone.

Some colors, like red or green, have powerful emotional associations in most cultures.



According to many studies, over 90% of whether we like a product comes from the color palette that it uses. When choosing colors, try to establish the target market, message, and style of your product down to the emotions you want your future users to have when looking at your design.

“

**Colors can have
strong emotional
associations
in people. ”**

Color, also known as the hue, belongs in one of two groups: the colorful and non-colorful variants (white, greys and black). Most people can recognize up to 10 million different hues.

The human eye is an organ with light-catching receptors, made of a series of rods and cones. Rods and cones allow us to see the light and color differences using a protein called Opsin. Various types of that protein react to different wavelengths of the light. Opsin S handles short waves (450nm), which translates to seeing blue and purple hues. Opsin M perceives the greens (around 540nm), and opsin L takes care of long waves around the red spectrum (closer to 590nm).

These three parts can are enough to describe nearly every color we can see. Those rules are also the context of the RGB Color system (Red, Green, Blue) that is common in most displays. It's not precisely matching how humans see colors, but we can approximate some hues, like yellow (570-580nm wavelength), by merging red and green light sources.

White reflects a wide range of visible wave spectrum, while black does the opposite - it absorbs everything. That's why some people say that black is the absence of color, while white is all the colors combined.

40% of women can see colors in four instead of the standard three hue channels. Because of that, they can discern a more significant number of colors.

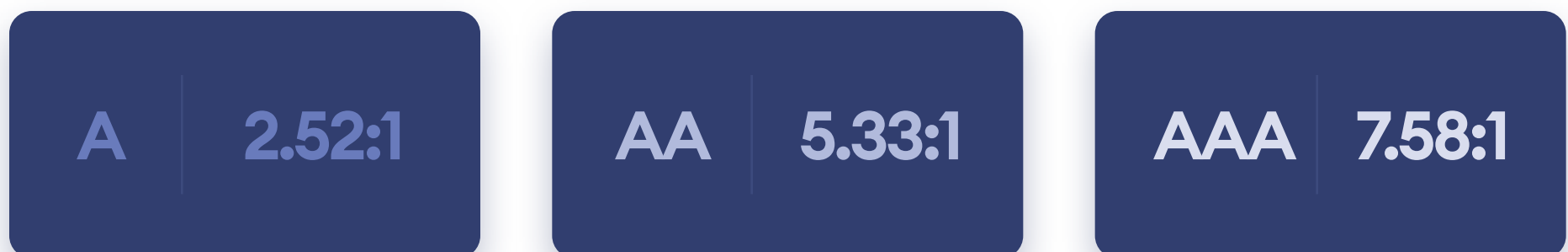
8% of men and 0.5% of women suffer from some form of color blindness.

ACCESSIBILITY

Accessibility of a digital product is a science of making sure the product is suitable for older users, visually impaired, color-blind, and others (including people on low contrast, low-quality displays). The general idea is that a digital product should be easy to "see" for everyone.

Color components like brightness and saturation influence the contrast, which directly affects the readability. The higher the readability, the easier the product is to use for a wider group of people.

Sufficient contrast levels help users with vision problems to be able to use the product as intended. We must remember to check the **WCAG 2.0** contrast for all essential UI elements. We can ignore the contrast on decorations and non-important parts. WCAG defines three contrast levels that depend on the size of the element.



A - low contrast

AA - average contrast

AAA - high contrast

When designing your interface, remember to have at least a 4:5:1 contrast ratio (or AA) for all of your most crucial UI elements (buttons, forms, essential text).

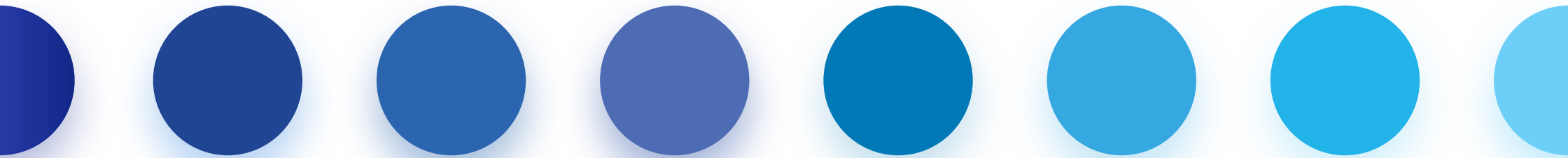
You can test your contrast levels by using a range of dedicated, often free plugins for the design tool of your choice. They work by entering two colors and calculating the contrast between them on the WCAG scale. There are also online tools that you can use to check and calculate contrast.

“

**Contrast is
essential for
accessibility.”**

BLUE

Blue is the most popular color when it comes to digital products. Many popular apps and websites are utilizing shades of blue for their interfaces or their branding.



Can you recognize the brand by its color?

From the left: Dropbox, PlayStation, Tumblr, Facebook, LinkedIn, Skype, Vimeo, Twitter

Statistically speaking, blue is also the most liked color - both by men and women. Because of that, using blue hues is a safe way to design a product - they rarely ever create negative emotions in people.

COLOR PSYCHOLOGY

Blue is the color of calm and relaxation. It's often associated with trust, professionalism, experience, or wisdom.

WHEN TO USE BLUE?

Blue is often the best choice for IT, finance, banking, health, and social media. It works well on large canvases, as backgrounds or background parts. Most shades of blue may not be the best choice for Call to Action and accents.

Because of the ubiquity of blue designs, a blue app icon can disappear between other similar looking products. It may be a good idea to choose an unpopular shade of blue, to better stand out from the already big, blue crowd.

GREEN

Green is the color to which the human eye is most sensitive. We can also discern the most shades of green, from all the other colors. That is because the cones that see green are sitting in the most sensitive part of the eye.



Can you recognize the brand by its color?

From the left: Carlsberg, Starbucks, XBOX, Greenpeace, Evernote, Spotify, Android

COLOR PSYCHOLOGY

Green stands for health, nature, calm, and relaxation. Depending on the shade, it can communicate emotions like harmony, stability, growth, security, or energy.

WHEN TO USE GREEN?

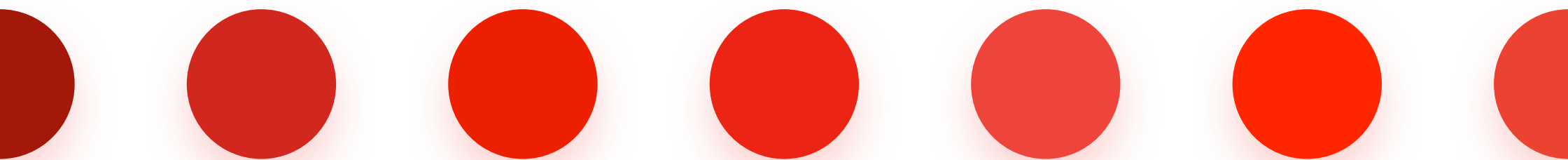
Green is widespread in IT, finance and banking, health, fitness, ecology, and food.

Green is also growing in popularity - as it's closely connected to ecology, it's often the primary choice for marketing materials for eco-conscious companies.

It catches attention and stands out from all the other colors. Green also evokes positive emotions and taking action, so it's an excellent choice for CTA buttons. In UIs, it's often associated with a successful action or process.

RED

Seeing red makes our heart beat faster and produces more adrenaline than other hues. It's a warning color, which helps it catch attention or cause a reaction. It's one of the most popular colors associated with sales.



Can you recognize the brand by its color?
From the left: KFC, RedBull, Toyota, Netflix, Rayban, CocaCola, Youtube

COLOR PSYCHOLOGY

Red is associated with both positive (energy, passion, strength, love) and negative emotions (danger, adrenaline, warning, aggression).

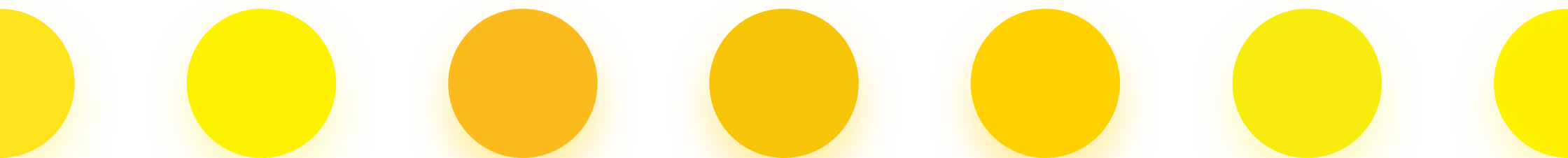
WHEN TO USE RED?

Red is most common in products for sports, food, services, cars and telecommunications.

Try to use this color responsibly - it can attract attention, but it's often a negative one. In interfaces, red is usually the color of an error or a warning. Using red as a CTA can be risky - in some industries, like finance, it's also associated with lower scores or value decrease.

YELLOW

Yellow evokes mainly positive emotions, as it's associated with the sun, warmth, and gold. Because of its positive connections, it's popular in advertising. But let's not forget that yellow is also a warning color.



Can you recognize the brand by its color?
From the left: MailChimp, Ferrari, Kodak, McDonalds, Ikea, Snapchat, PostIt

COLOR PSYCHOLOGY

Yellow is enthusiastic, self-confident, happy, optimistic, and fun. But to counterbalance all that, it's also a warning color, that we react to subconsciously.

WHEN TO USE YELLOW?

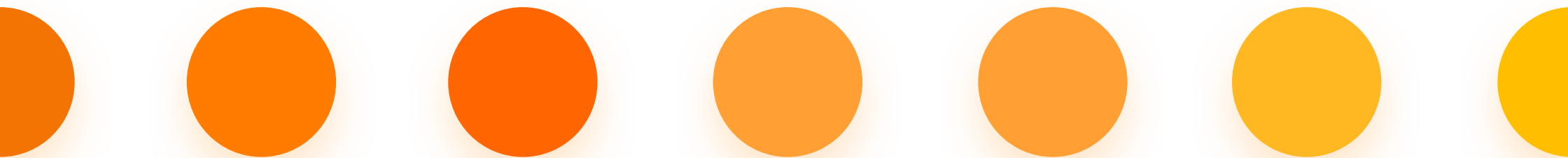
Yellow is popular in industries like food, sales, creative services, and art.

Add too much yellow, however, and you can cause discomfort in your users. Just like with red, it strains the eye faster and causes some discomfort in large doses.

In digital products, yellow works best with its lighter, pastel hues - mostly used as background. Darker, more aggressive yellows can be work as accents, and in places where we want the user's attention to go.

ORANGE

Orange is an energetic and optimistic hue - just like yellow. It calls to action and symbolizes activity, so it's a popular choice in sport suits. Just like red and yellow, it's also a warning color.



Can you recognize the brand by its color?

From the left: Harley-Davidson, Orange, Nike, Nickelodeon, Amazon, Lufthansa, McDonalds

COLOR PSYCHOLOGY

Orange is associated with youth, energy, creativity, fascination and activity. It can also mean bad taste or being cheap.

WHEN TO USE ORANGE?

Orange is popular with the food industry, sales, telecommunications, and products for children.

Just like yellow, orange can make a physical product (and its packaging) seem affordable and approachable.

In digital products, orange works best for accents and CTA's, because it's not as aggressive as red or yellow. It's also not as strongly associated with risk as with the other two.

PINK

Pink is the stereotypical color associated with the feminine side. Products and services targeting women, mothers are often using shades of this hue in their branding.



Can you recognize the brand by its color?

From the left: LG, Barbie, TMobile, Cosmopolitan, Johnson&Johnson, AirBnb, Victoria's Secret

COLOR PSYCHOLOGY

Pink is associated with women, motherhood, innocence, youth, romance, and gentleness. It can also mean naivete and childishness.

WHEN TO USE PINK?

Pink is popular with the cosmetics industry, fashion, pregnancy, female health, foundations, and non-profits.

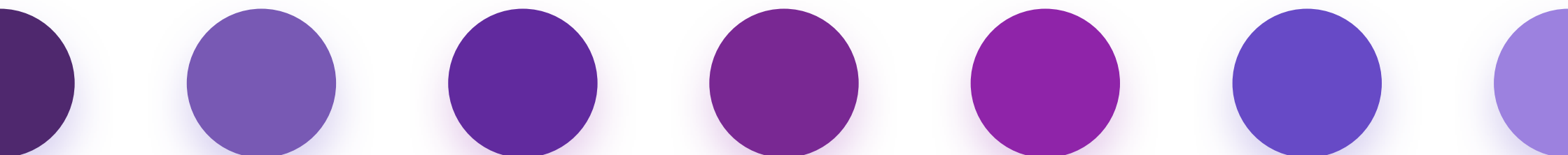
Because of its strong ties to a specific group, it works best in products for women and children.

It's darker, more subtle hues can be suitable for telecoms or advisory services.

Pastel, soft hues of pink work well as backgrounds, with the more saturated ones as accents.

PURPLE

Purple is the rarest color, both in digital products and in nature. It used to be a universally bad choice, but that is starting to change now. You can often see purple in telecoms and technology.



Can you recognize the brand by its color?
From the left: Premier League, Twitch, FedEx, Yamaha, Yahoo!, Cadbury, Milka

COLOR PSYCHOLOGY

Purple means luxury, power, wealth, and secrecy. These associations mostly come from European culture and history. Other meanings include professionalism, wisdom, trust, high quality, and modernity.

WHEN TO USE PURPLE?

Purple is most popular with the IT industry, luxury goods, finance, banking, and food.

It's also a very designer-friendly hue - it works well with many other colors like blue, green, orange, and yellow.

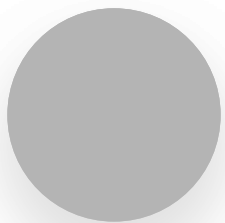
It can be a bit overwhelming when used extensively, so monochromatic purple designs may not be the best idea.

BLACK AND GREY

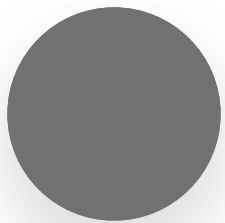
Black and **shades of grey** are serious, formal, and emotionally neutral colors. They're associated with elegance, minimalism, professionalism, and luxury. They can, however, feel a bit exclusive and, when used in excess, can be quite depressing.



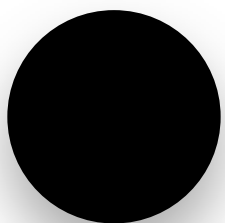
In the design world, grey is most common in wireframes and specific UI elements (like text fields or inactive icons).



Light-grey items suggest being inactive or disabled, so try not to use it for essential objects or text. Light-grey text is usually associated with low-importance labels, explanations, and suggestions.

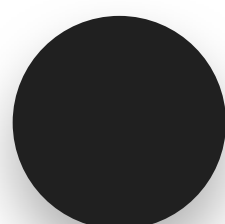


Shades of grey are also the base of setting hierarchy through contrast in dark mode interfaces.



Pure black
(#000000)

When designing for computer displays, we should use pure black (#000000) scarcely, if at all. That particular color is not present in nature, and our eyes are not used to seeing it. In designs, it can create unnatural, repulsive effects due to its super high contrast against other colors and ghosting effects on OLED mobile screens.



Dark grey (#202020)

Try to use very dark shades of grey instead of pure black whenever possible.

WHITE

White is the color of minimalism, often suitable for creating clean, "airy" designs. Due to being the brightest of all colors, it's often associated with clarity and sterility. It's popular in both architecture, art, and fashion.

White is the color we work with the most while designing user interfaces. In most projects, we are using white backgrounds, margins, cards, text-boxes, and drop-down lists.

White is also the base for the name "white space," which is the negative space that sets hierarchy and order in design.

WHITE COLOR VS WHITE SPACE

White is often confused with white-space, with many junior designers assuming whitespace has to be white. The term comes from the print industry, where most of the paper types are white, but we are not limited by background colors in digital products.

You can achieve white-space on any color, and you should as it's an essential part of building hierarchy and structure in a layout.

COLOR PALETTES

A color palette is a set of colors that work well together, combined to form a brand or a concept. It helps to set a mood for the final design. A palette can use colors close to each other on the color wheel or include accents from the opposite sides.



Defining a color palette is essential for the design process. It allows for color consistency, setting hierarchy, and understanding action types.

A typical color palette should contain both neutral and accent colors. It's also good to include notification colors (green for success and red for failure) - especially if forms are an essential part of the functionality.

You can find many color palette generators online that can help you choose the right hues and shades that match your initial color idea.

Before you start using auto-generated sets, however, it's worth knowing how it all works.

“

**Defining a color
palette helps with
keeping visual
consistency. ”**

MONOCHROMATIC

A monochromatic palette uses the shades of just one color as the entire set.

It allows for using almost the entire spectrum of that shade, including black and white. Monochromatic is a safe choice, as it's challenging to come up with clashing colors in a set like this.

The downside is the contrast can be a bit lower than with other palette types, and our designs can end up looking quite dull.



Title or heading

Subheading

Longer paragraph text.

MAIN ACTION

SECONDARY ACTION

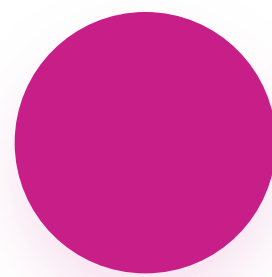
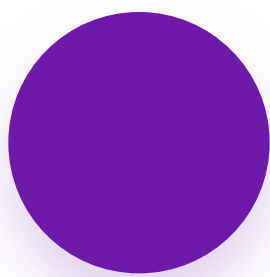
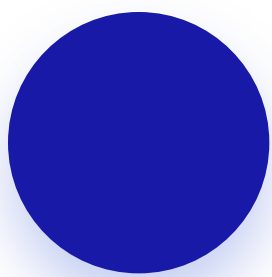
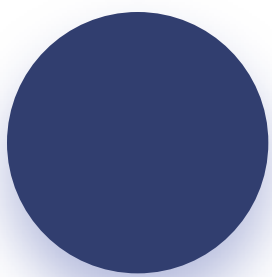
TERTIARY ACTION

ANALOGOUS



The analogous palette is a collection of colors that sit closely together on the color wheel. The downside of this choice is a slightly lower contrast between the colors, than in other palettes. To make it work, try experimenting with the shades.

This palette is a good, safe starting point allowing for many sets of fitting hues.



Title or heading

Subheading

Longer paragraph text.

MAIN ACTION

SECONDARY ACTION

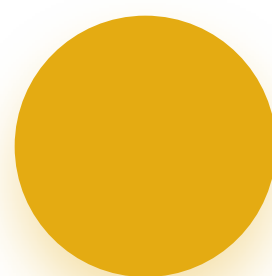
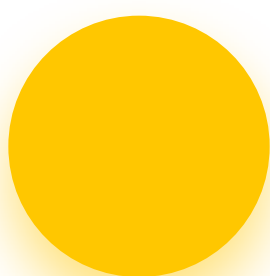
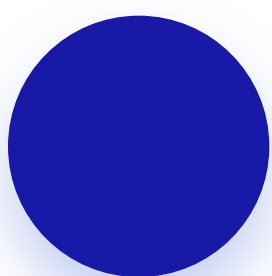
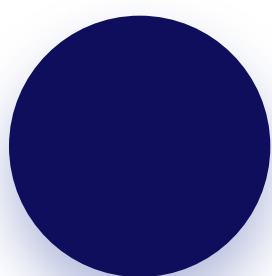
TERTIARY ACTION

COMPLEMENTARY



The complementary palette uses colors that sit on opposite sides of the color wheel. This way helps with achieving a high contrast between the hues.

Using the complementary palette, keep in mind to avoid clashing colors by testing their overlapping contrast. The best way to find the matching set is by choosing colors with less than 85% saturation.



Title or heading

Subheading

Longer paragraph text.

MAIN ACTION

SECONDARY ACTION

TERTIARY ACTION

TRIADIC



The triadic palette is an equilateral triangle on the color wheel. You can start by choosing its tip, and then pick the resulting two bottom corners. Triadic color sets are often more colorful and playful.

When using this palette, try to set your primary color as the tip of the triangle, with the other two being complementary. You can use the golden ratio here as well, with 60% for the primary, 30% for secondary, and 10% for tertiary.



Title or heading

Subheading

Longer paragraph text.

MAIN ACTION

SECONDARY ACTION

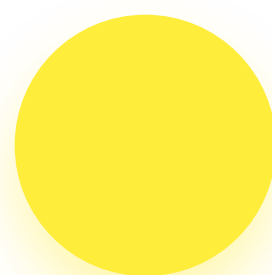
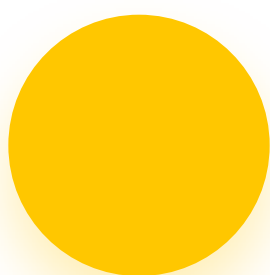
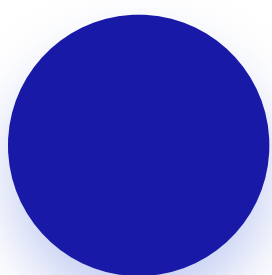
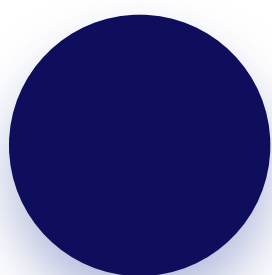
TERTIARY ACTION

SPLIT-COMPLEMENTARY



The split-complementary palette starts with choosing one color and adding two colors on both sides of its complementary hue. It forms a narrow triangle on the color wheel. This type of palette is a safe way of achieving excellent results that work well together.

You can test the colors to choose the right hues and shades. The goal is to have the right amount of contrast, neither too high nor too low.



Title or heading

Subheading

Longer paragraph text.

MAIN ACTION

SECONDARY ACTION

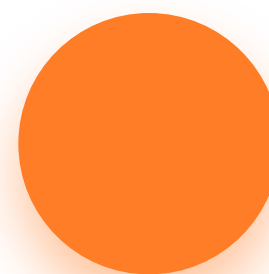
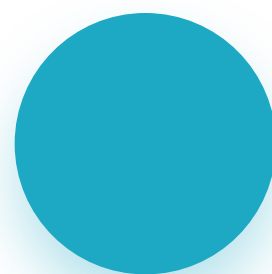
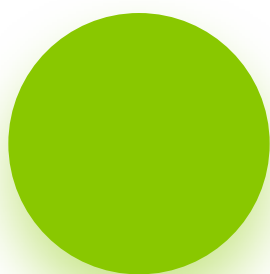
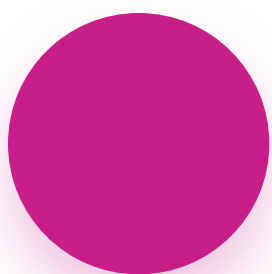
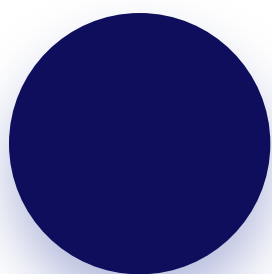
TERTIARY ACTION

RECTANGULAR



A rectangular color scheme uses four colors positioned around the color wheel in the shape of a rectangle. The best way is to choose one complementary set first, and then choose another one that forms a rectangle.

With that many different hues, it's good to choose one primary color, and use others as accents. It's good to balance warm and cold shades for the best result.



Title or heading

Subheading

Longer paragraph text.

MAIN ACTION

SECONDARY ACTION

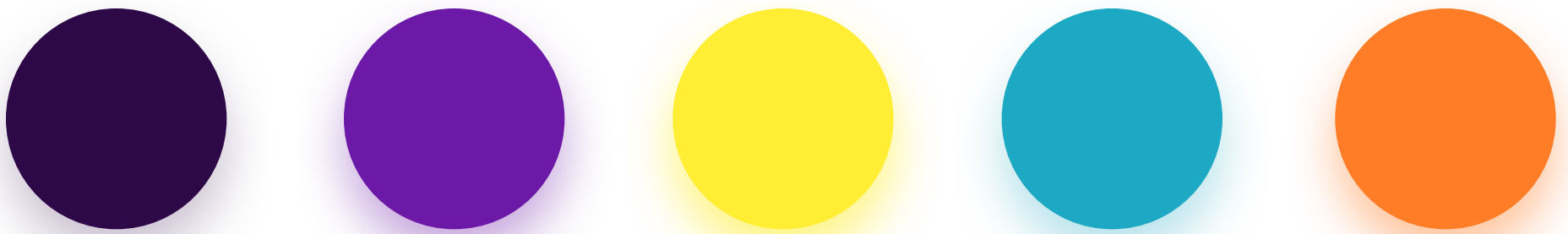
TERTIARY ACTION

SQUARE



The square palette is a twist on the rectangular one, with the main difference being the two color spaces between our corners. The resulting shape is an even square.

Just as with rectangles, try to use one color as primary, with other more scarcely represented.



Title or heading

Subheading

Longer paragraph text.

MAIN ACTION

SECONDARY ACTION

TERTIARY ACTION

GREYS

When defining your color palette, it's a good idea to create a mix of primary and grey hues.

If your greys have a hint of the primary in them, they're much easier to fit well into the design and make it feel more polished.



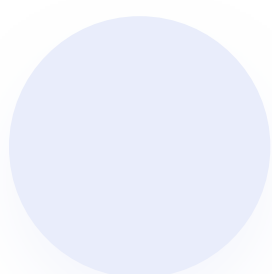
Primary color (#2F51D2)



Pick your primary color and decrease its Lightness and Saturation to about 30. You'll end up with a dark shade of grey mixed well with the primary hue.



To create the mid-range grey try setting the Brightness to 90 and decrease Saturation by ten from the initial value.



The lightest primary-grey shade uses even higher Brightness, at around 95. This shade works well as a background color.

COLORS THAT DON'T GO WELL TOGETHER

There are some color combinations that nearly never work well together. They're either an eye-sore, low legibility, or both, thus decreasing the overall quality of the product.

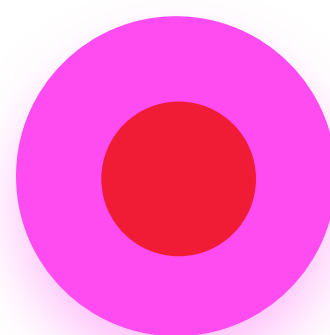
Because our screens emit light right into our eyes, the line between those colors can sometimes seem to "shake."



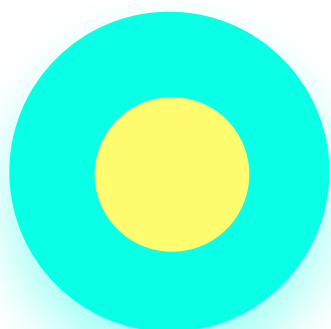
Blue and red



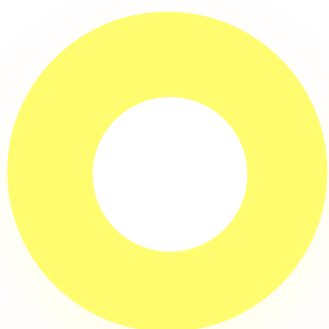
Green and red



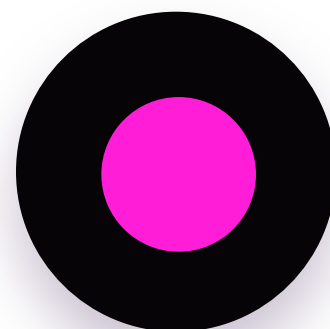
Pink and red



Two very bright colors



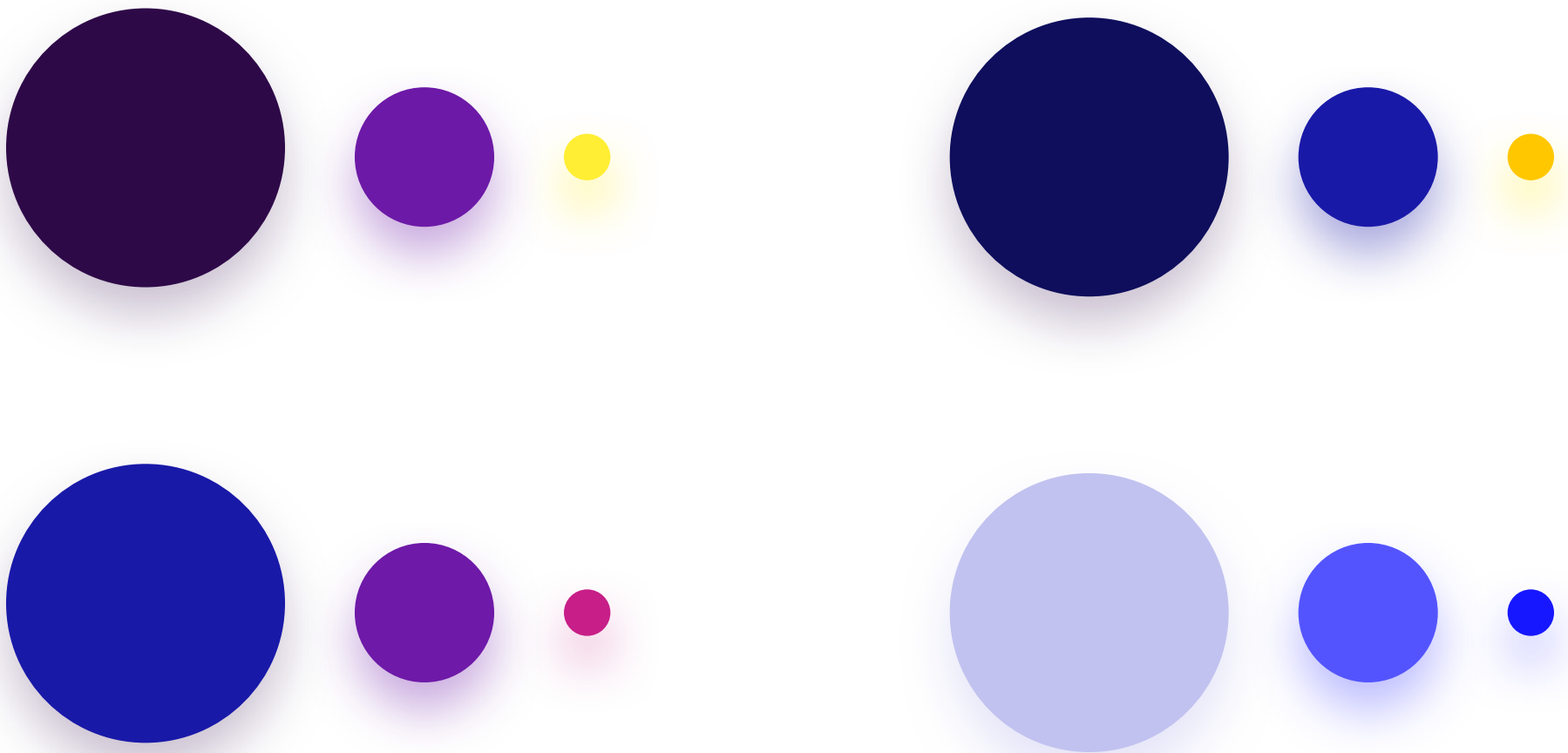
Yellow and white



Black and pink

BALANCING A PALETTE

When choosing a color palette, it's also essential to balance the colors in it. If you plan to use all your colors equally, the result may end up being random. The best approach is to use the Golden Ratio.



GOLDEN RATIO 60/30/10

For example:

60% Dark Blue, 30% Purple, and 10% Pink.

Dark blue is the primary color - all of its shades (darker and lighter hues) are best suited for backgrounds and low-priority elements. It takes about 60% of the space in the design.

Purple is the supporting hue, great for higher hierarchy objects. For it to be sufficiently visible, its use is limited to about 30%.

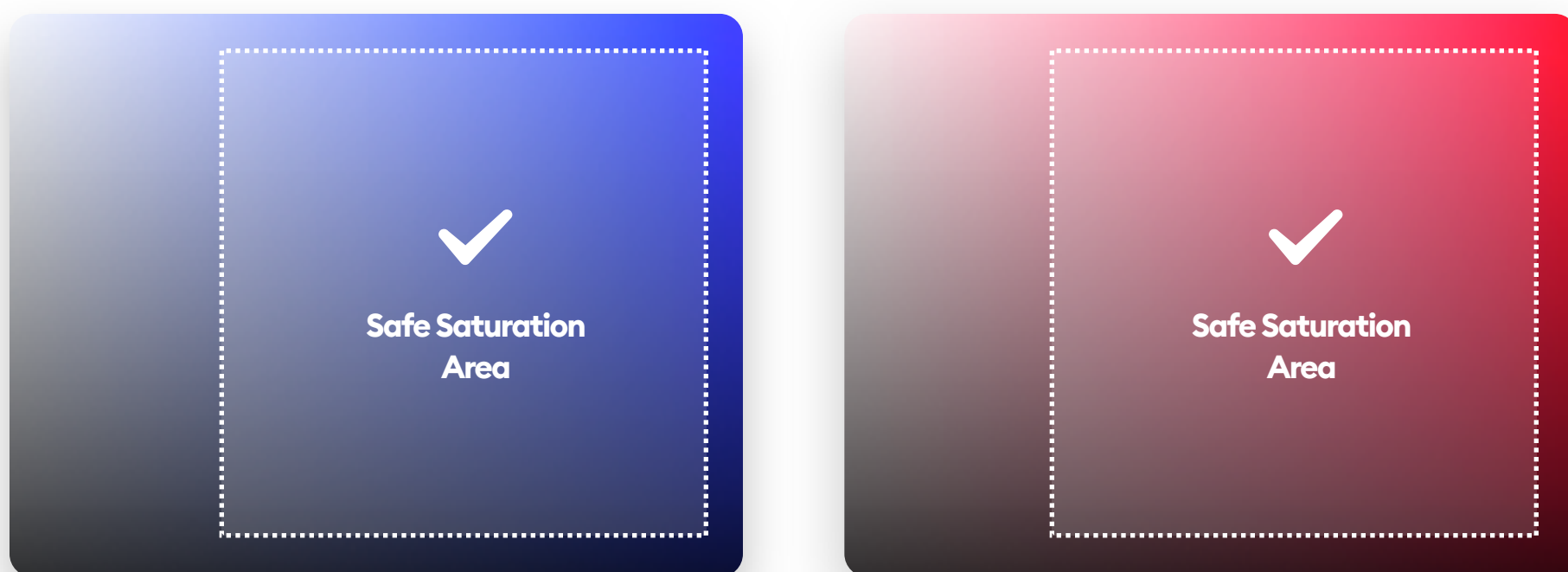
Pink is the color for accents and CTA's, so 10% coverage works best, as it highlights the most important elements.

SATURATION

Contrast is one of the most basic readability metrics.

However, in cases when contrast is too high on an already bright display, it can cause a strain on our eyes. It's good to find common ground between accessibility and aesthetics.

If you decide to use highly saturated, very contrasting colors, try to do that purposefully. It can work on event or product pages, but may not be the best choice for a news portal, e-store, or a banking app.



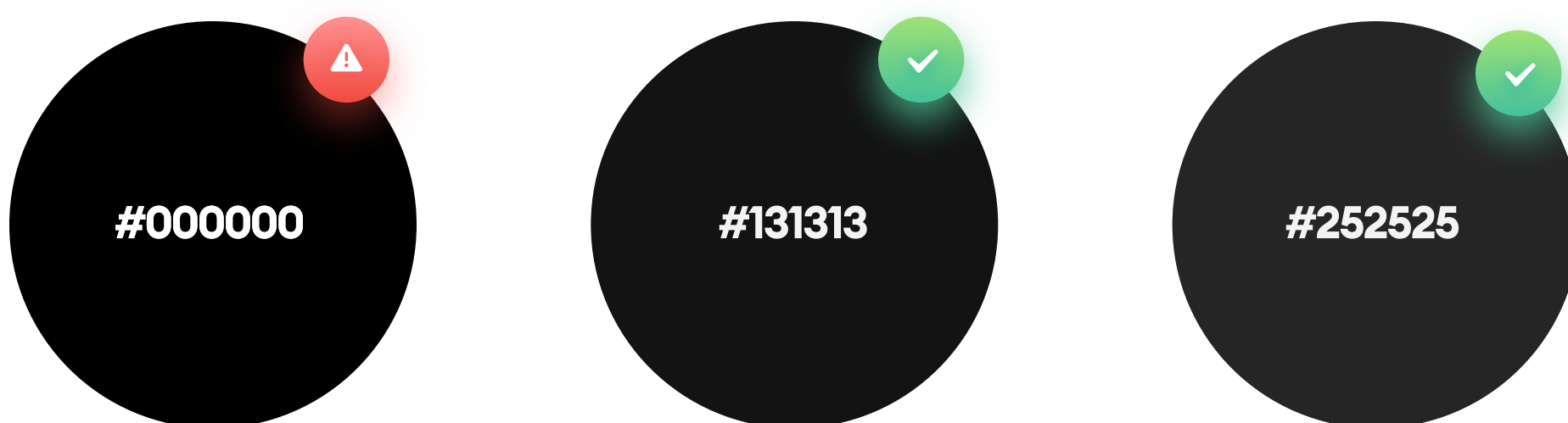
A good practice would be to avoid exceeding 90% color saturation until you get comfortable enough to choose the right color combinations. The image above shows a typical safe area to pick safe-looking hues.

PURE BLACK COLOR DOESN'T EXIST

Using pure black (#000000) in a digital product has been causing controversy for a few years now. Should you use it?

While pure black gives the best possible contrast (especially on a white background), it's not the most optimal choice for digital design. A contrast ratio that high can negatively affect the readability of the interface.

Ink on a sheet of paper is, of course, as black as it gets. But every time we look at it, the light bounces off of it differently. The result is often some shade of dark grey, as pure black doesn't exist in nature, which makes our eyes less adjusted to it. We perceive pure black as something unnatural.



Because the screen is glowing, it creates an effect of a "no brightness" color that glows (except for OLED screens), which can be very unpleasant to look at. When we keep the contrast in the safe ranges, it's best to use a dark grey instead of pure black for the web. You can use pure black on mobile, but it can lead to an additional problem of "ghosting" animation on OLED displays. You can read more about pure black in the chapter on Dark Mode in UI Design Styles companion book.

“

**Try to avoid pure black
(#000000) in your
designs. ”**

SYSTEM STATES

When building a color palette, it may be useful to define our system state colors next to it. The most common states are:

Positive (success / confirmation) - green or blue

Payment successful.

Your subscription is confirmed.

Negative (error / failure) - red

Payment was declined by your bank.

Neutral - grey, blue

Settings saved.

Settings saved.

Warning - orange or yellow

Your subscription is expiring soon.

Terms and Conditions have changed.



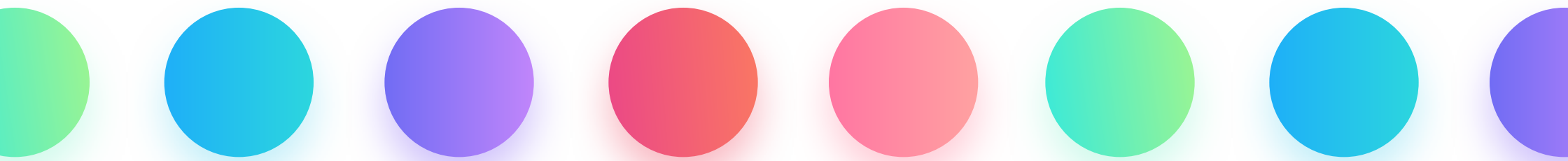
11.

Gradients

Gradients

THE BASICS

Gradients have been a ubiquitous trend during the Web 2.0 time (early 2000's). Partially because they helped achieve the Skeuomorphism look of 3d, realistic buttons. After a while, however, many designers got bored with them and declared gradients unfashionable and a bad-taste design. It all came back to prominence after iOS7, and later Modern Design brought gradients back into the spotlight.



Because of their increasing popularity, many designers started calling gradients "the new colors."



DOWNLOAD

A skeuomorphic button from the first iPhone era had a strong gradient, a thick border, hard shadow, and typical to that style highlight at the top.

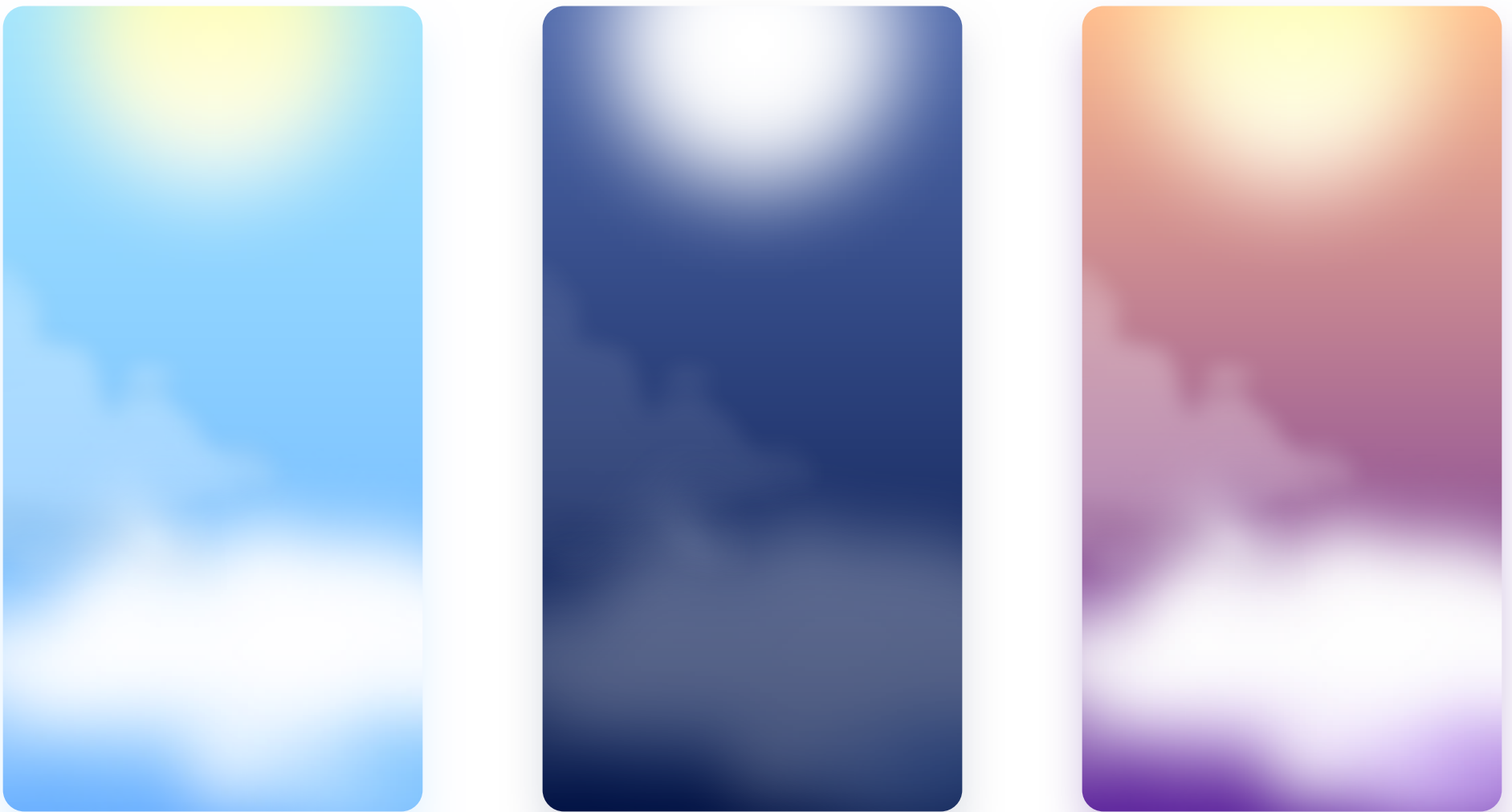


DOWNLOAD

Modern gradient, without any extra decoration, can work with a soft shadow made from the same color, creating a beautiful, semi-realistic look.

WHY DO WE LOVE GRADIENTS?

We like familiar things, especially if they exist in nature. Need proof? Look at the sky at any time of day. The sky is not just one solid color, but a gradient, that depends on the weather and lighting.



Most objects we see are three-dimensional, reflecting light and casting shadows. Because the light is rarely uniform, the resulting colors and shadows are never fully flat or filled with just one color. They are gradient in nature, which makes this particular tonal mix more natural than solid colors.

Look at the everyday objects that surround you. All of them are gradient in nature, and they also change with the change of external light sources.

“

Look around. Nearly everything you see is actually a gradient.”

THE POWER OF GRADIENTS

When we start choosing our gradients, it's crucial to remember about their two main qualities.

One is the ability to create depth and a more defined shape.

The other is about catching the eye and guiding it to certain parts of the shape. Our eyes prefer bright, warm hues, with pretty high saturation.



FLAT BUTTON

Even a subtle gradient will make your buttons more friendly and "clickable."



GRADIENT

Depth allows gradient + shadow combinations to appear a bit closer to the eye - which helps in understanding the interactions behind them.



GRADIENT + SHADOW

That doesn't mean we should completely abandon flat design and come back to skeuomorphic buttons of the previous decade. Both approaches have their benefits and downsides, so it may be useful to create a healthy mix of the two.

The flat design style allows for a less cluttered, more minimal design that focuses on functionality over form.

Skeuomorphic, or simply a more natural design, reminds us that we're designing for real people, who prefer things they already know and understand. Realistic gradients can add that "human factor" to an otherwise minimal project and make it more user-friendly.

GRADIENTS IN BRANDING

In this day and age, it's challenging to find a brand color that's not already "taken" by someone else. Blue belongs to Facebook, Twitter, or LinkedIn. Orange is for Amazon, while Green is for Starbucks, Spotify, and many others.



Colors have, of course, different shades, and we can recall them pretty well. This color memory is the reason we recognize Twitter light blue instantly and won't mistake it for Facebook's "office-blue."

But there are not many significantly different, yet still, good looking hues left for a new brand to use. All the best ones seem to be reserved, and we don't want our app to be mistaken with Facebook (well unless we do ;)

We can, however, merge two colors into a beautiful looking gradient and make it our defining characteristic. You can then mix popular and unpopular hues at a specific angle and have something that is both unique and good-looking.



Tickety



datefinder

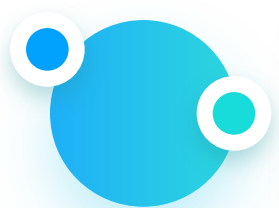
GRADIENTTYPES

A gradient is a transition between two or more colors. The colors can also have a transparency value. You can use that to achieve the fade out effect of an object relative to its background. All colors can even start at a different spot relative to the object, with the default being from its edges. You can drag the gradient handles well outside of your object to achieve an even subtler result.

There are three main types of gradients:

LINEAR

Linear is the most common gradient type. As its name states, it's a linear transition of two or more colors. We can modify the colors themselves, their transparency, and the gradient angle.



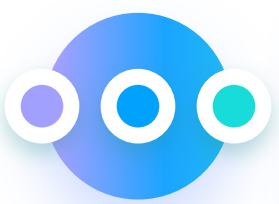
Horizontal, Two Colors.
Left to right.



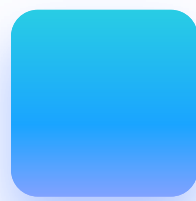
Vertical, Two colors.
Top to bottom.



Diagonal, Two colors.
45 degree angle.



Horizontal, Three Colors.
Left to right.



Vertical, Three Colors.
Top to Bottom.

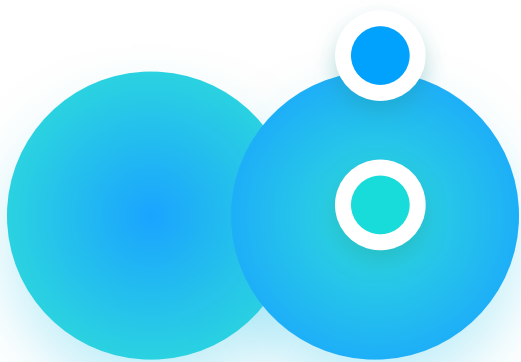


Diagonal, Three Colors
45 degree angle.

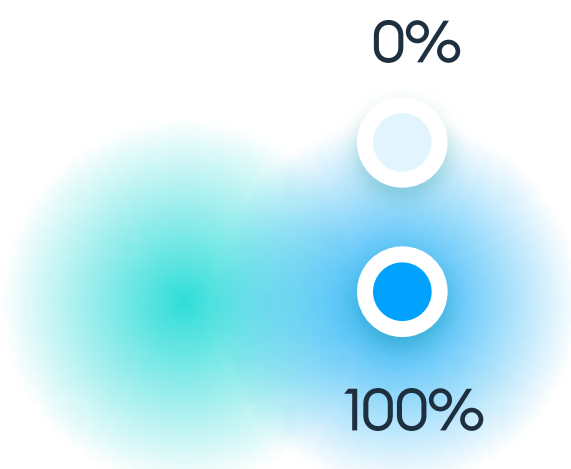
GRADIENTTYPES

RADIAL

A radial gradient starts with one color in the middle of the gradient ellipse, while the other ends up being on its edge. The most common type of this gradient is the circular one, where the transition is distributed evenly along the circle.



Two colors with no transparency can create a 3d look in circles by adding a highlight and shading.



If both ends of the gradient are the same color, with one at full transparency, we end up with a blur or fade-out effect.



A radial gradient works well on non-circular shapes by adding a bit of an organic, realistic style to them.

GRADIENT TYPES

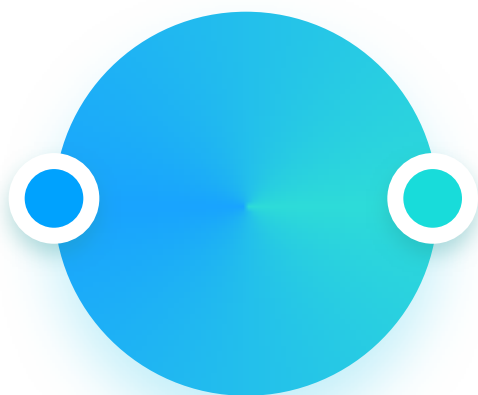
ANGULAR

An angular gradient goes around in a counter-clockwise circle. The angle between the colors defines whether the transition goes both ways (180-degree angle) or has one sharp dividing line in the middle (at 0-degrees).

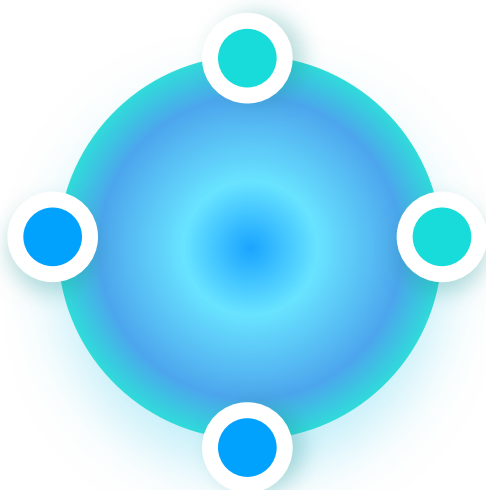
This type of gradient is rarely a good choice for interfaces.



Two colors set in the same spot will have a sharp edge between them, while the gradient goes around the other way.



Two colors at 180-degrees will have a fluid transition between the colors going both ways.



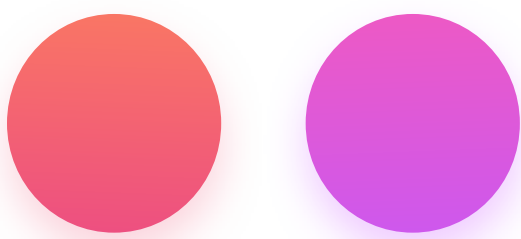
You can add more colors and play around with the angle to achieve some interesting results.

How to create a gradient that both looks good, and makes sense for great interface design? There are a few basic rules to follow to achieve that.



TWO COLORS

Two-color gradients are the best choice for most designs. With more colors, the gradient can become visually busy.



CHOOSING THE RIGHT COLORS

Try to match similar colors, warm hues with warm ones, and cold with cold.



That will help you avoid bad mixes (with the most extreme one being red and green)



SUBTLE GRADIENTS

The best gradients are often very subtle, with not a lot of difference between the colors. It makes them look more natural and easier on the eyes.

BEST PRACTICES

Gradients are a popular choice for buttons and other interactive elements. Here are some tips on how to make them shine.



VERTICAL

Vertical is the "classic" button gradient. If the lighter color is at the top, it looks more 3-dimensional and almost asking the user to activate it. If you reverse the colors, it can look like a pressed state.

HORIZONTAL

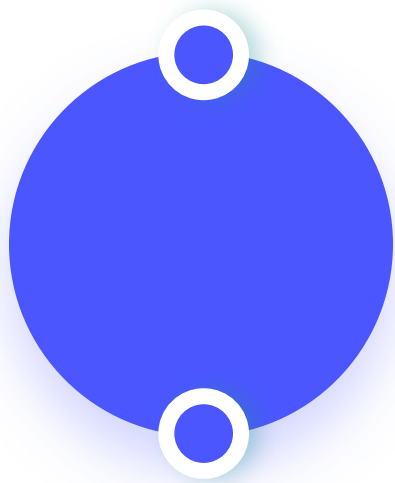
Horizontal gradients may not make as much sense in the 3d space, but they are gaining popularity because they look a little less generic than their vertical counterparts. We lose some of that convex style, but choosing the right colors can make the button look more dynamic and inviting.

DIAGONAL

Diagonal gradients help us achieve that organic, natural look. If you're trying them out for the first time, start safe, with a 45-degree angle.

BEST PRACTICES

You can pick nearly any color and create a beautiful looking gradient out of it. If you're starting your gradient journey, however, it's best to use the safe method that works every time.



STEP ONE

Create a linear gradient and use the same color on both ends. In our case, it's simply a transition from #4B56FF to #4B56FF.



STEP TWO

Check if your color palette is in HSB mode (Hue, Saturation, Brightness) and decrease or increase the Hue value of one of the gradient endings by 15-30. In our case, the color on top has the Hue value reduced by 30.



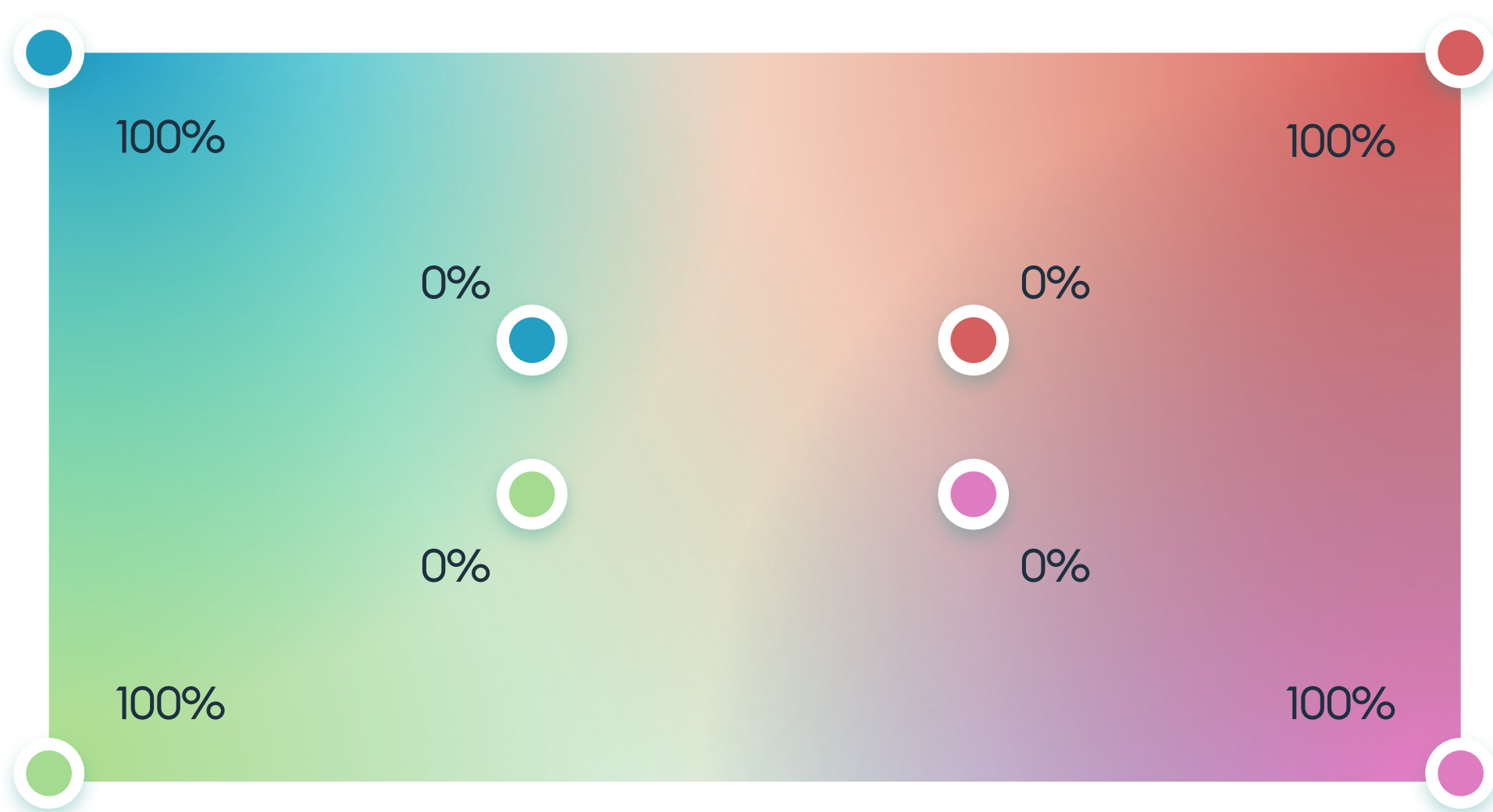
STEP THREE

For a more organic, friendly effect, you can try rotating the gradient (here it's at 45 degrees) or decreasing the Saturation by 10-15 points.

SOFT GRADIENT

Soft-gradient is a type of gradient that uses a couple, seemingly random tonal transitions. You can create it by applying a radial gradient with the same color on both ends, but with the outer edge having full transparency.

If you create this on a rectangle and move each radial-gradient instance into one corner, you can create an interesting effect of colors mixing into a very unusual pattern.



“

**Subtle gradients
between similar
colors are perceived
as the most beautiful.”**



12.

Typography

Typography

INTRODUCTION

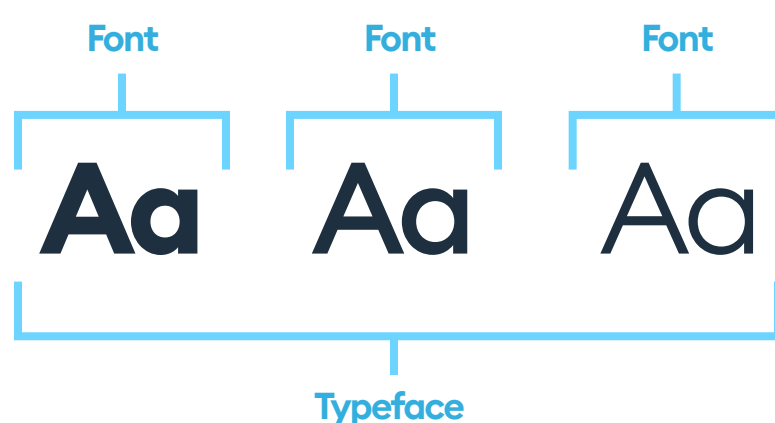
Typography is one of the broader subjects in design, and there are many books dedicated solely to this matter. There are, however, some basic typography rules for UI that can help bring your designs to the next level. The fonts you choose profoundly influence the look & feel, style, and message of the entire project. That's why it's essential to know your fonts.

Aa Aa Aa Aa Aa

FONT OR TYPEFACE?

There is a distinction between a font and a typeface that is important to understand to communicate better. A Typeface is a collection of various styles of the same lettering. Each of those styles is a font.

So Helvetica is a typeface, and Helvetica Bold is a font. In general, however, a font is a digital representation of how letters look. That presentation is about all the individual letters and their relation.



Aq Ęę

Óó Ćć

Aa Aa

Aa Aa

Aa Aa

Aa Aa

Aa Aa

Aa

Aa

SPECIAL CHARACTERS

Many languages have a set of distinctive characters that don't exist in English.

If the product you're designing is going to be multi-lingual, check if your font supports all the letters like q, ś, ø or ē.

VARIETY IN WEIGHTS

Choose a font that has a variety of weights. It will help you set the right hierarchy between elements. Light fonts are not that readable, but starting at regular, make sure you at least have two thicker weights like bold and extra-bold.

SIMPLE FORM

Simple is good. Choose a font that's not overly ornamental or fancy. The main goal should always be readability. More natural, cursive, script or handwriting fonts can go well with logos or headings.

READABILITY

Check a font in all of its sizes. If you can still read it at a tiny size and it looks good while super-big, you should be safe. In general, it's best to use thicker weights while designing UIs.

FONT STRUCTURE

Each font has a set of characteristics, based on its structural definition. It's good to understand how precise and relevant these values are.



Type

BASELINE

The baseline is a flat surface on which our font is sitting. You can use it to align the type to other on-screen elements.

[Type

CAP-HEIGHT

Cap height is the height of the capital letter.

[Type x

X-HEIGHT

The height of lowercase x is as a default value for other, less uniform lowercase letters.

Type

ASCENDERS

All elements above our x-height are called ascenders.

Type

DESCENDERS

All elements below our x-height are called descenders.

KERNING

Kerning is the space between letters. All fonts have a default kerning value that differs between them.

Typography Wok

The safe way is to use the default value or a global value in UI as it may prove challenging to code an entirely custom set. When the kerning is too small, or too large, it can negatively impact readability.

If you're not sure how to set it, a safe range in which you can play around with the default value is between -0.5 and +0.5.

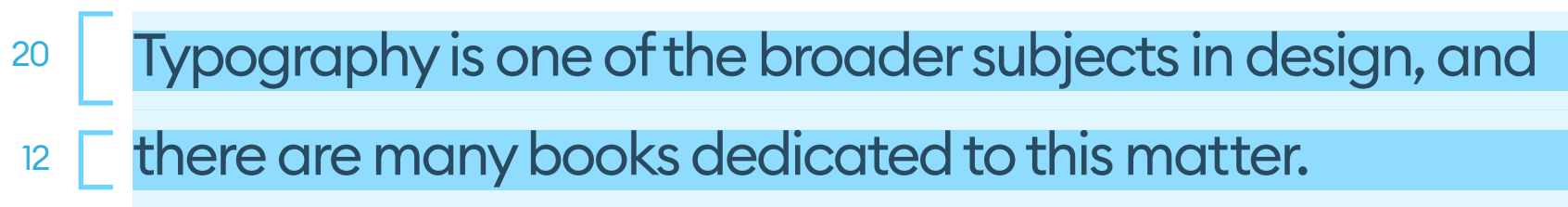


Sophisticated print designers often modify the kerning of individual letters for a better flow. It may be a good idea for illustrative typography but is nearly impossible to do in code, which makes it less relevant for UI design.

LEADING

Leading is the vertical space between lines of text.

In mobile apps, it's usually a slightly lower value than on the desktop. The default value can often be too small for the readability of larger blocks of text. You can try to increase it slightly, or use the golden ratio by multiplying the font size by 1.618 to get the best line-height value. The line-height name should be used instead of leading as this name also functions in most of the programming languages.



```
20 [ Typography is one of the broader subjects in design, and  
12 [ there are many books dedicated to this matter.
```

In the above example, we have the text size set at 12 points. Next, we multiply this by 1.618 and get 19.4. After that, we round the number to the closest value and get 19.

The problem here, however, is that our text size is an even number, while the line-height is an odd one. It means it likely won't distribute evenly inside the line.

We round it up to 20, which is the next even number. The same rules apply to other font sizes. If your size is an odd number, the line-height has to be odd as well.

“

**If you're not sure,
stick to the default
values for kerning
and leading.”**

Whitespace is the empty area between letters, words, lines of text, and everything else we see on-screen. Too small or too large whitespace value negatively impacts readability and hierarchy. Whitespace is essential to a good layout, and getting it right takes a lot of practice. Soft-grids can help you define whitespace between objects, but it's crucial to understand the typography values first.

Typography

Introduction

Typography is one of the broader subjects in design, and there are many books dedicated to this matter.

SAFE

This example shows large enough margins and spacing for all the elements to be readable.

Typography

Introduction

Typography is one of the broader subjects in design, and there are many books dedicated to this matter.

TOO SMALL

When the kerning is a very low number, and everything is very close together, it's a lot harder to read and process.

T y p o g r a p h y

Introduction

Typography is one of the broader subjects in design, and there are many books dedicated to this matter.

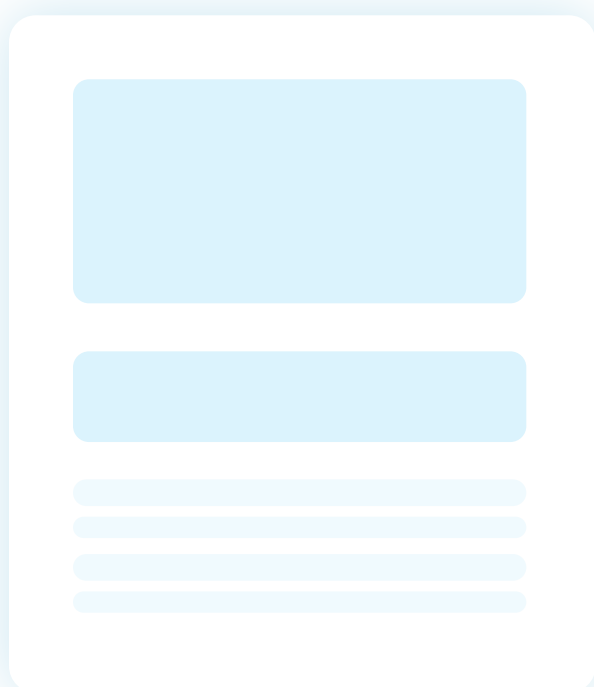
TOO LARGE

Uneven margins with very high whitespace make design look chaotic and hard to read. It seems as if there are no rules applied to the layout.

HIERARCHY AND GRADATION

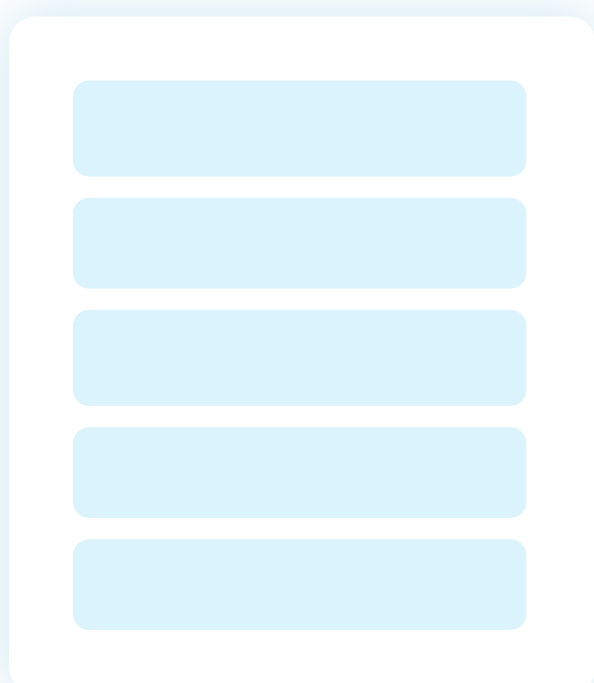
A clear, visual hierarchy helps with faster processing of the interface. But how do you achieve that?

Just as in books and magazines, the content we put into our UI can have a set of basic rules. Things like headings, subtitles, quotes, and body-copy all play a significant role in how easy the design is to understand. Gradation helps with building hierarchy by adding more visual impact to top-level, essential elements like headings and titles. You can use both the size and weight of your font to create gradation and visual hierarchy.



GRADATION AND HIERARCHY

This content has a large heading, slightly smaller subheading, and much smaller body copy. That gradation defines the hierarchy and makes it much easier to understand the purpose of every element.



NO GRADATION

Without gradation, there's no hierarchy. When every element has the same visual weight, it makes everything look the same. You don't know where the title ends, and the copy begins.

There are many ways to define text hierarchy. We will cover the most basic ones here. Once you get comfortable with them, you can try to experiment.

The Golden Ratio (1.618) has many uses in design. One of those uses is defining hierarchy. Let's assume your body text (also known as paragraph or `<p>` in HTML) size is 10p. Multiply that by the Golden Ratio, and you'll get 16 for the subheading (H2). Multiply it again by 1.618, round it up, and you'll get 26 for the main title (H1).

H2
16p



Good practices

The easy way out.

Typography is one of the broader subjects in design, and there are many books dedicated to this matter.

H1
26p

P
10p

You can also use your base-grid value (8 or 10) as a divider for your font sizes. In the case of having a base number of 8, you end up with 16, 24, 32, 40 and 48 possible font sizes that work well together. Of course, it all depends on the font itself, but this method works well for many simple, sans-serif typefaces. (For more on sans-serifs see page 153)

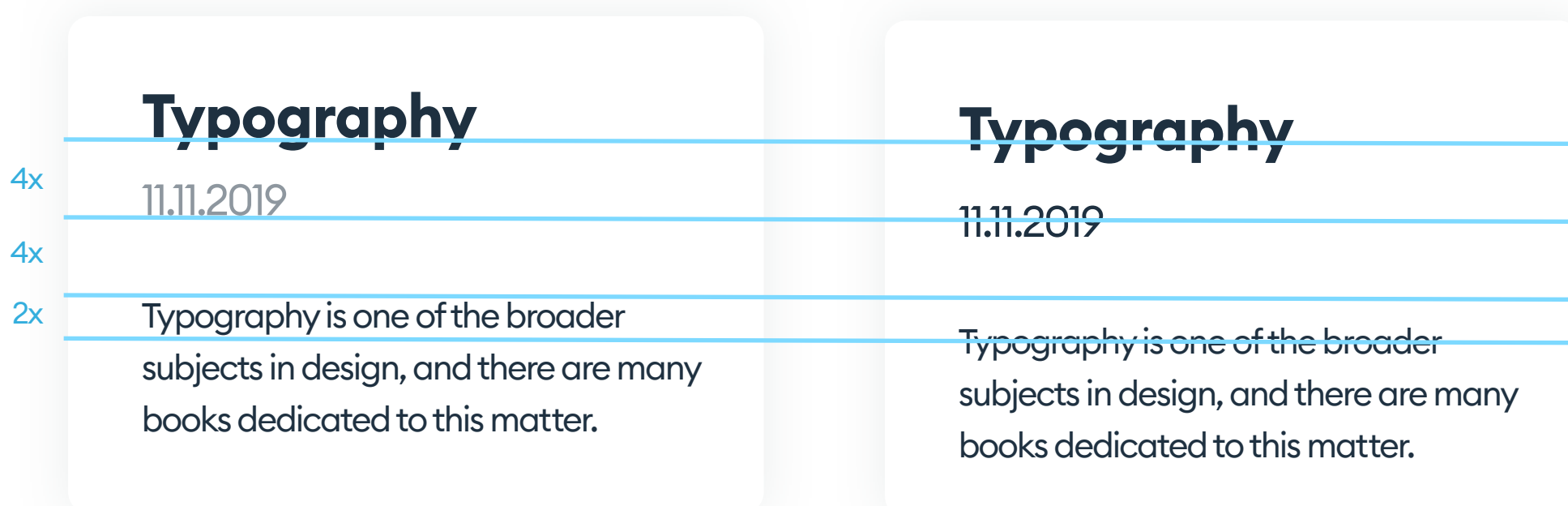
There are other, more sophisticated methods to explore after the mentioned two. But the main goal is to have higher-level elements (like titles) bigger than lower-level parts (like paragraphs).

GRID

While designing websites and apps, a grid of vertical lines is the most common way to layout objects horizontally. If you add a horizontal grid, you can use it to align text and create text rhythm.

It helps build the right hierarchy through a combination of font size and font spacing. This method works best, especially for text-heavy projects like news apps or websites, blogs, and wikis.

By aligning our font baselines to those horizontal guides, we enforce a hierarchy, which in turn makes the text a lot easier to scan and read.



HORIZONTAL GRID

You can use many rules (including the golden ratio) to set the right text rhythm. One of the easier methods is using multiplications of your font x-height. If you set the standard spacing to double the x-height, you can create more significant gaps by increasing that number again (to 4x).

NO GRID

The content is not laid out based on a grid. All the vertical spacing is either automatic or entirely random.

ALIGNMENT

LEFT ALIGN

Text alignment is a cultural thing and can differ between countries. Most of the western world uses the left-side alignment because we read from left to right. Irregularity of the right edge (called the jagged edge) of a text block shouldn't be a concern unless it's very significant. There are, however, cultures that read from right to left or from top to bottom, so make sure to accommodate that if your project will be multicultural.

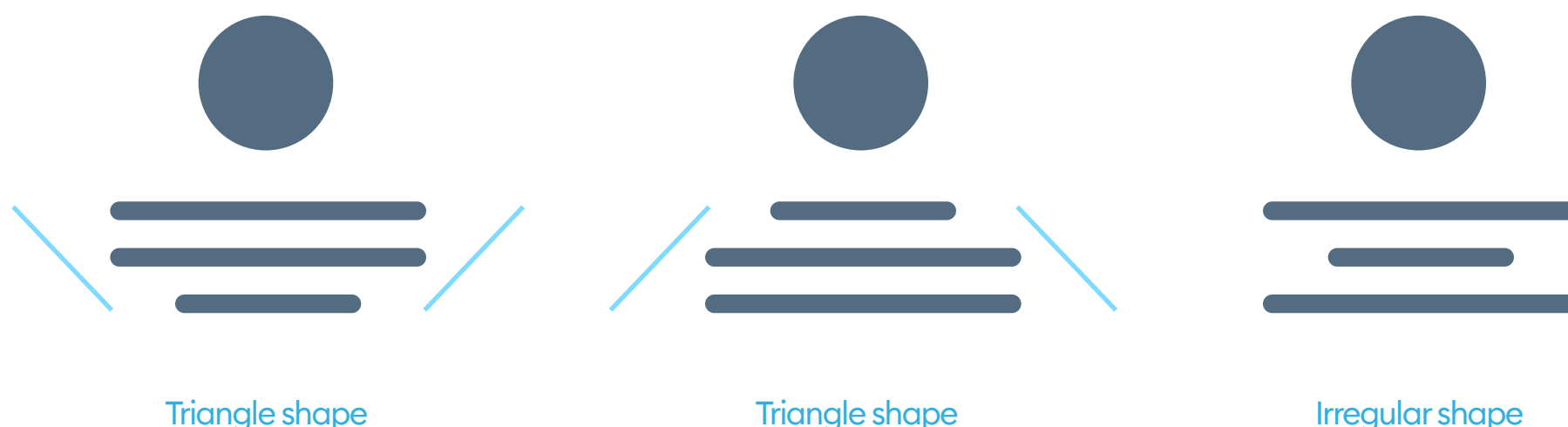


JUSTIFIED TEXT

Try to avoid justified text while designing digital products. Having both text edges uniform may seem friendly, but it happens at the cost of irregular kerning and varying spaces between individual words. Justified text is harder to achieve in code in a readable way.

CENTERED TEXT

Center aligned text works well when it's short. You can use center-align for two or three sentence blocks. Anything longer than that will be much harder to read. Centered text works well as a subtitle for icons or images. Try to remember to align it properly along the Y-axis of the figure. The best shape of the text is a triangle, as it makes it easier to read than an irregular shape.



READABILITY

LINE LENGTH

If you're designing text blocks for mobile devices, try to keep within 30 to 50 characters per line. That will help with achieving legibility on a small screen.

On larger screens (like tablets and laptops), the best range is between 6 and 9 words per individual line. This method should help you pick the right font size for a specific text block.

CONTRAST

Phones and tablets are often very close to our faces, with their backlights going right into our eyes. Very high contrast combined with shining light and a close-range can be very unpleasant. While you should always think about having an accessible design, try not to overdo it with ultra-high contrasts either.

As explained before, try to avoid pure-black (#000000) when designing text blocks. It's much better to use dark grey (like #222222) or better yet use a grey that's mixed with your primary color. If your app or website is mostly blue, it may be useful to have a blue-grey shade for the fonts to match. (#22224A in our example below).

Typography

Introduction

Typography is one of the broader subjects...

PRIMARY ACTION

The primary dark blue CTA defines a blue-grey shade for our fonts. That makes every part of the info card internally consistent and more pleasant to the eye.

“

**Try to keep between
30 and 50 characters
per line in mobile
designs.”**

SANS-SERIF FONTS

Sans-serif is a category of typefaces that do not use serifs - the small lines at the ends of characters. Sans-serifs are well suited to digital products because of their simplicity in shape and form. They look good even on lower-quality displays.

Aa Aa Aa Aa Aa

All font styles have their purpose. Sans serifs are more minimal, modern with the primary goal of avoiding distraction. The simplicity helps in focusing on the content itself without deciphering the twists and turns of the typeface itself.

WHEN TO USE SANS-SERIF FONTS?

Sans-serifs are the base font style for all digital interfaces.

They work especially well for both headings and mid-length text blocks, descriptions, labels, buttons, and forms.

If your text is longer than ten sentences, it may be better to try a serif font instead. The small lines of the serifs can guide the eye to faster jump between characters in a sentence. It's not necessary to always use serif fonts for longer text, however. Sans-serif will do just fine if your product doesn't use serif fonts anywhere in its branding.

SERIF FONTS

Serif fonts got their name from having little additional lines (called serifs) at the character edges. They also used to be called Roman fonts.

Aa Aa Aa Aa

While it may seem decorative or even ornamental to have those little lines in the font, they do have a specific purpose. A serif font is easier to read because the serifs help in discerning the individual lines of text. They guide the eye. Serifs are also considered more serious in tone, but the main benefit is simply their ability to be more readable in longer text.

WHEN TO USE SERIF FONTS?

Serif fonts are great for print media (books, newspapers, and magazines).

In digital products, they should co-exist with sans-serif fonts. Using only serifs in an app or on a website is not recommended unless the choice is branding related.

In digital media, Serifs are best for headings or body copy in blogs and articles. However, if you choose a serif font for the paragraph text, try to have a sans-serif title (H1) for establishing a better hierarchy.

DECORATIVE FONTS

Decorative fonts are script, calligraphy, handwritten, and all the other stylized typefaces. They're often neither serif nor sans-serif, but rather more complex in shape and form. Decorative fonts shouldn't be used in interfaces unless they serve a specific purpose. A good example is an author's signature under a blog post - a handwritten font will make it look more personal.

Aa Aa

Decorative fonts can also remove a layer of seriousness and make your product seem infantile or plain silly. It's especially a case for serious financial or banking products.

Using popular typeface choices is especially vital when starting work as a designer. Look around and see what fonts great designers are using and try to pick the same sets yourself. Remember, that simple, readable style will be a lot easier to fit into any kind of design scenario. The more complex the font, the harder it will be to make it work well in a design.

FONTS TO AVOID

Some fonts are simply not made for UI and high-density displays. Their style is either dated or was severely overused over the years. For that reason, try to avoid system fonts of the past decades (they were often not updated to modern standards).

Try not to use:

Arial Times New Roman Tahoma Courier
Verdana Calibri Comic Sans Trebuchet

Decorative, thin, or otherwise elaborate fonts are a poor choice for UI design. The more little lines and details a font has, the harder it is to process and understand. It takes attention away from the important UI elements.

*This font will be super hard to read
on a small screen.*

SMALL SIZES OF THIS FONT CAN BE UNPLEASANT TO LOOK AT.

*USING ONLY CAPITAL LETTERS WON'T MAKE
THINGS ANY EASIER.*

“

**Simple is always
the best choice.”**

Both Android and iOS have their dedicated fonts that should be your first choice for designing apps on these platforms. You don't have to use those fonts in your design, but unless the font you choose is free, you can face pretty hefty licensing fees. In general, app usage of paid fonts is the most expensive way to use them.

Try to use mobile system fonts in your first designs exclusively. The users already know those typefaces from all the other apps they are using, so your design will instantly seem familiar.

You can download and use both of these for free.

Try not to mix the fonts between the platforms, however. Using San Francisco (Apple font) for anything other than a Mac OS or iOS app can be against the licensing terms of that typeface.

San Francisco (SF UI)

Apple (iOS, macOS, watchOS, tvOS)

Roboto

Android, Material Design (Mobile + Web)

FONT SIZES ON MOBILE

Both Apple and Google also suggest the smallest readable font-size for body copy and inputs. Keep in mind that all fonts are different, and these values are only for the mobile system fonts like SF (for iOS) and Roboto (for Android).



It seems like 18 will be the safest, most readable option that also adheres to the WCAG 2.0 rules. Remember - people are not afraid to scroll. It's best to have larger, more readable fonts than to fit more lines of text on one screen.

Of course, it doesn't mean you shouldn't use fonts smaller than 16 in your designs. Just try not to use them for main body copy and inside your form fields. These smaller sizes work a lot better for descriptions, labels, and meta-data.

FONTWEIGHT

LIGHT, ULTRA LIGHT

Super thin fonts were at their peak popularity in 2012 and 2013. Around that time, iOS 7 marked a radical departure from Skeuomorphism towards minimalism in iPhones, and light fonts reigned supreme. It turns out, however, that they're not a great choice for readability - especially at smaller sizes.

For a thin font to be readable, it has to be the right size. In many cases, small, the light text ends up with stroke widths being lower than 1 pixel, resulting in antialiasing and even decreased legibility.

This size is enough for that weight.

This is too thin.

This will be super hard to read.

Remember that with thin and light fonts, you need to make them darker to achieve the right contrast with the background.

EXTRABOLD, BLACK

Super thick fonts are nowadays a bit more popular, but they're not problem-free themselves. Thick, densely packed characters in small sizes may achieve the right contrast, but can still be very hard to read.

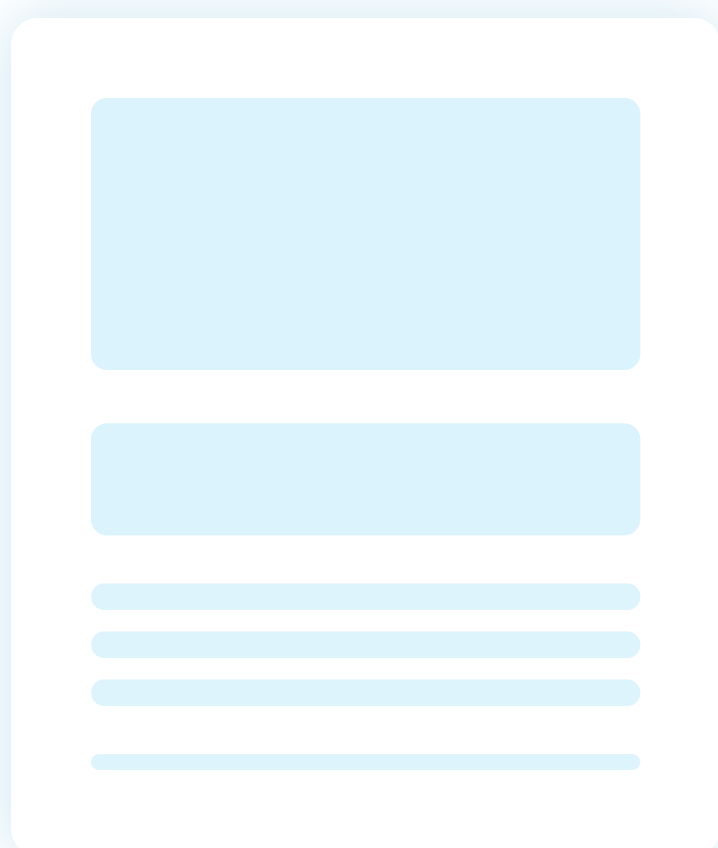
This size looks good.

This is starting to get harder to read.

Hard to distinguish individual letters.

CSS (Cascading Style Sheets) is a way to define the visual characteristics of objects on the web. Contrary to popular belief, it's not actually "coding," but rather a way to define attributes of on-screen objects.

CSS standardizes font definitions in a way that you should follow in your design. A typical UI often uses the four main CSS styles of H1, H2, P, and Span.



H1

H1 is the main title.

H2

H2 is a subtitle.

P

P is paragraph text or body copy - used for longer blocks of content.

Span

Span is often a way to specify all meta-data like publication date, comment count, author name, etc.

It's good to use those names in the design process, as it will make it a lot easier during developer handoff. The naming convention works excellent for both text styles or group or layer names.

FONT PAIRING

You can mix and match serif and sans-serif fonts. Starting with a base sans-serif typeface, you can add a serif counterpart for headings. It will make the text stand out more.

Interface design can also utilize just one typeface. In that case, it is essential to differentiate content types using font-weights (like Regular, Medium, Bold).

Here are the basic rules to follow with font pairing.



Pairing fonts

The basics of font pairing are quite simple.

FONTWEIGHT

This is an optimal ratio of heading to paragraph font weights.



Pairing fonts

The basics of font pairing are quite simple.

The difference in weight can't be too extensive. In this example, the header is visually heavy and makes the paragraph text disappear.



Pairing fonts

The basics of font pairing are quite simple.

Headings should nearly always use a ticker font than the paragraph text. In this example, the hierarchy is hard to understand.

Work Sans



Pairing fonts

The basics of font pairing are quite simple.

Montserrat + Work Sans

TWO SANS-SERIFS

Two distinct sans-serif typefaces can work well together. The secret is choosing ones that are different enough from each other.



Pairing fonts

The basics of font pairing are quite simple.

Muli + Brandon Grotesque

When two typefaces are too similar in their level of ornamentation, the result can be messy.



Pairing fonts

The basics of font pairing are quite simple.

Playfair Display + Work Sans

SERIF AND SANS-SERIF MIX

While trying to pair a serif typeface with a sans-serif one, try to choose the simplest possible sans-serif font. The serifs are usually complicated enough visually.



Pairing fonts

The basics of font pairing are quite simple.

Playfair Display + Brandon Grotesque

A more decorative serif font combined with a decorative sans-serif makes the text block complicated and hard to read.

Pairing fonts

The basics of font pairing are quite simple.

Montserrat + Georgia

Pairing fonts

The basics of font pairing are quite simple.

Montserrat + Georgia

Pairing fonts

The basics of font pairing are quite simple.

Georgia

Pairing fonts

The basics of font pairing are quite simple.

Playfair Display + Georgia

If you plan to use a serif font for paragraph text, try matching it with a thicker sans-serif heading for best results.

Choosing a thinner sans-serif can severely impact the readability of the entire text-block.

Entirely serif-based designs should be avoided unless that choice is brand-related.

Two different serif fonts are also hard to do right. In most cases, the result will be too complicated visually.

THE RIGHT NUMBER OF FONTS

When it comes to the number of typefaces, weights, and sizes, try to keep it simple. If it's possible, try to choose one or two fonts, with two weights (regular and bold) and between 3 and 5 sizes. The next step is to use those rules consistently across your design.

Mixing and matching sizes and weights help to differentiate headings, paragraphs, and interactive elements without creating chaos.

The example below uses Work Sans with three weights (bold, medium, regular) and size gradation of 24,16,12 and 8.

Heading (h1)

Work Sans Bold (24px)

Subheading (h2)

Work Sans Medium (16px)

Paragraph text / body copy (p).

Work Sans Regular (12px)

Smallest text, used for meta-data and labels (span).

Work Sans Regular (8px)

Typographic consistency is essential for all the interactive parts of the interface, such as buttons and forms. A button label should use the same font size and weight across all the interface buttons.

PRIMARY BUTTON

Work Sans Medium (8px)

SECONDARY BUTTON

Work Sans Medium (8px)

GHOST BUTTON

Work Sans Medium (8px)

“

**Try to limit your
fonts to one or two
per project.”**



Icons

Icons Icons

13.

Icons

Icons

Icons Icons

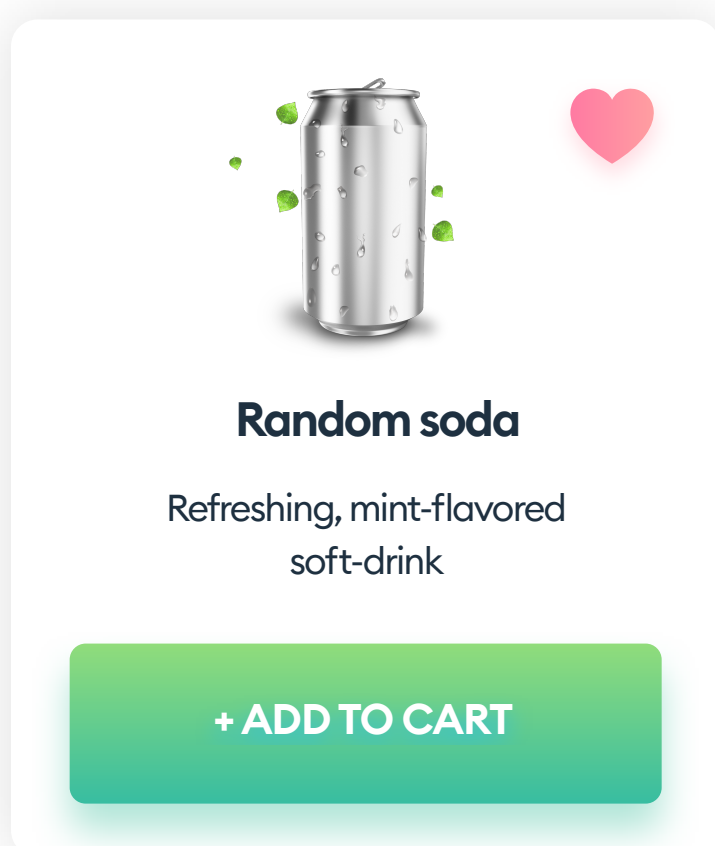
Icons

INTRODUCTION

An icon is a small pictogram that either symbolizes a function or status. Most icon shapes come from a simplified version of everyday objects and shapes.



Keep in mind, however, that the meaning of an icon is not always universal, and even the most obvious shapes can be interpreted the wrong way by some users. The simpler the icon, the better chance of understanding it, but in many cases, it's good also to include a text description.



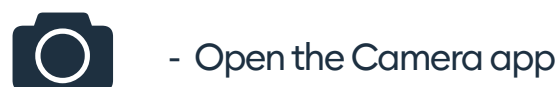
Icons can mean different things in different cultures. Do your research and pick the best shape for your target group. One of our apps made for farmers was tested for icon understanding a few years ago. When asked what the heart icon does, 67% responded with "it means the product is healthy," and only 21% said it means "add to favorites."

“

**Even the most obvious
icons can still be
misunderstood.”**

ICONS AND LABELS

Should we use text labels with icons? And if so, then when is the best time to use them? While having a label always helps, you can safely skip it for universal icons that mean the same thing in many other products. A small portion of users may still get them wrong, but given the ubiquity of the shapes, they will work in most cases. These include:



You can also skip the labels with icons related to top-level navigation like the hamburger menu and the back arrow. If your target audience is a little older or less tech-savvy, it's actually good to keep the labels here.

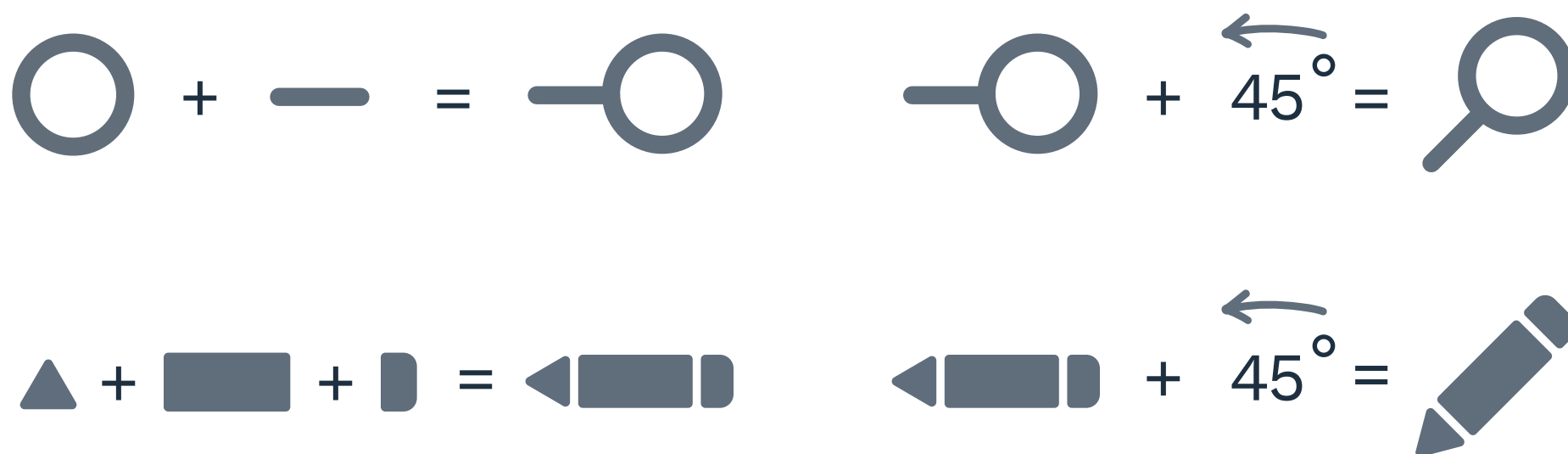
Icons are a way of setting the tone and style of the product, making it more user-friendly. But we need to remember that they are mostly there for aesthetic or space-saving purposes. Having an icon with a label will make it a lot easier for the users to understand what it does right away.

The longer or harder to understand the label, the longer it will take the user to understand its meaning. It's best to use just one simple word for the label.

The more visually complex shape, the harder it is to understand its purpose. Try to make the icons simple enough to convey the right message.

WHAT ARE ICONS MADE OF?

Most icons are simple shapes that you can create yourself in any design tool. Of course, there are plenty of beautiful, free libraries of icons to be found online. But if you're learning design, it's an excellent exercise to draw some of the simple shapes yourself first.



For example, a magnifying glass icon, which is synonymous with search, is a simple oval that's empty inside and a small line attached to it.

We combine the shapes and rotate them 45 degrees for the final form.

You can mix and match simple rectangles to create the + button, an arrow, or an X. The easiest way to draw them is without rotation, so start with a vertical or horizontal shape, and rotate it when the entire form is complete. It's a lot easier than drawing it already turned. An excellent example of this is the pencil icon shown above. By drawing some initial, simple icons yourself, you'll also learn how to build them so you can adjust downloaded icons to match the style of your product. You can use the stroke of the shapes to be able to switch the thickness of the icon easily.



ICON STYLES

It is essential to have all the icons of our product done in a consistent style. There are a couple of main factors to consider when aiming for consistency.

LEVEL OF DETAIL

The best interface icons are usually simple, but you can do a more realistic, very detailed icon when necessary.



Simple / abstract shapes



realistic, detailed shapes

FILL VS OUTLINE

The icons can be a full shape, filled with a color, or outlined, empty inside with a specified stroke width. You can also add more attributes like shadows or multiple colors.



solid icons



outline icons

ROUNDNESS

The edges and corners can be straight and sharp or rounded, which is considered more user-friendly. Sharp corners work well for very serious products, while rounded ones are perfect for friendly, consumer-facing designs.



Rounded edges



Sharp edges

CONSISTENCY

Keep your icon styles consistent throughout the product. Having icons from "different sets" will create a chaotic, less defined look.



Consistent corners



Varied corners



Same type of fill



Different types of fill



Similar Level of Detail



Various Levels of Detail



A simple, small sized icon



More detailed, larger icon

ROUNDNESS

All icons should have the same type of edge and corner rounding. Avoid mixing rounded and sharp corner icons.

FILL

Unless it's intentional for setting the state (on vs. off), avoid mixing outline and filled icons on one design.

LEVEL OF DETAIL

The smaller the icon, the less detailed it should be. As the shape gets smaller, the complexity of lines will blur and make it harder to understand. You can make larger icons slightly more detailed, but keep in mind that the more detail you add, the longer it will take to process.

CONSISTENCY



Same stroke width



Different stroke widths



Friendly, outline icons



Friendly, filled icons



Serious, outline icons



Serious, filled icons

WEIGHT

If you decide to use outline icons, remember to have them all use the same stroke width at all sizes. The thicker the stroke, the more visual impact (weight) the icon has.

FRIENDLY VS SERIOUS

The look and feel of icons can be either friendly, professional, or somewhere in between.

Friendly icons tend to be more rounded. Outline icons to be considered friendly should have a thicker stroke.

Serious, (professional) icons are either very subtly rounded or made with sharp corners. They're often more minimal and use a much thinner stroke.

LEGIBILITY

When picking the icon style, try to test various icon sizes to see how they scale down and how much detail they lose in the process. Lower quality displays can also show some of the more detailed ones as blurry.



SIMPLE + FILLED

Simple, filled shapes with rounded corners look pretty good at any scale.



SIMPLE + OUTLINED

Outlined icons with a thick stroke are pretty clear at most sizes. They lose some detail when very small, so it's crucial to test if they're still fully visible.



COMPLEX + FILLED

Filled icons with sharp corners and more detail, lose some of their legibility when scaled down to a smaller size.



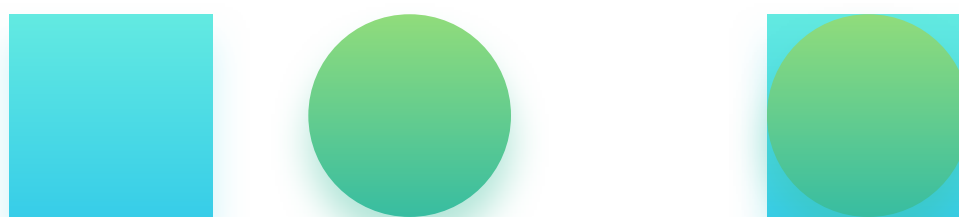
COMPLEX + OUTLINED

Thin, detailed outline icons lose a lot of the detail when we scale them down. Some lines start to disappear when anti-aliasing is applied, making them blurry and hard to see.

ICON SIZE

Shapes can have a different optical weight. It means an object can take less visual space, even if it's the same height or width in pixels as the other one.

The best way to see this is by making an oval and a square - both at 100 x 100 pixels. When you put them side by side, you realize that the square seems to be bigger than the oval. It is because it takes a lot more space on the sides, which makes it look more substantial.



All icons (and especially icons used in the same context) should be proportionally scaled. We can't allow for one icon to feel bigger and stronger than the rest, as it will make it stand out more and break the visual harmony.

Having all icons be the same size in pixels does not guarantee they "look" the same size. In many cases, you need to adjust the shape by hand to match.

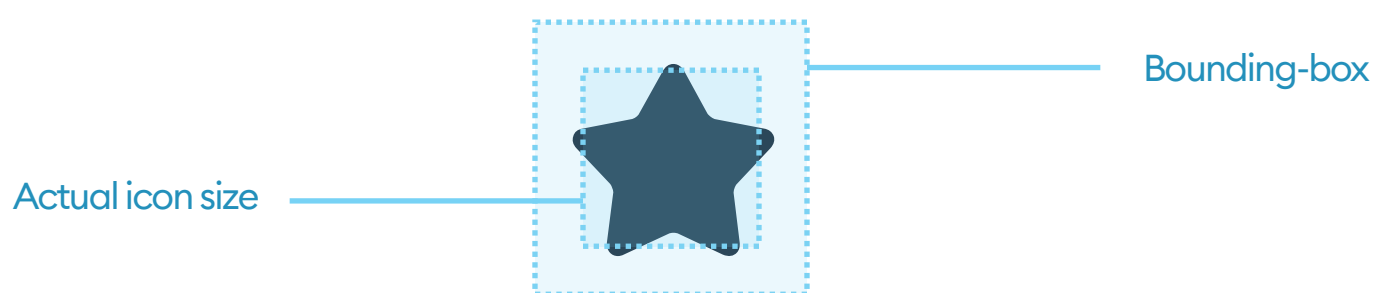
The optically smaller icon can be slightly bigger to make up for the lost space on the sides. The goal is to have both shapes have a similar visual weight.



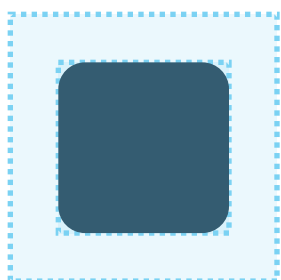
SAFE AREA / BOUNDING BOX

A bounding box is a safe area around the icon that we can use for irregular shapes to take a bit more space.

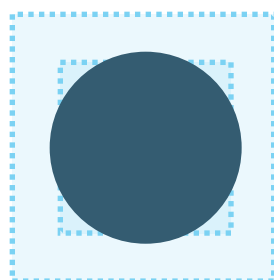
To make it easier to adjust the icon weight, we create a safe area around the most square-looking shape of all. Then use the same frame to adjust other icon shapes to fit.



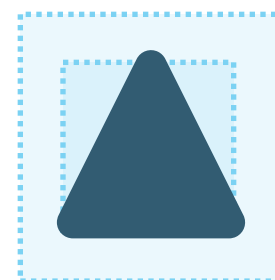
Then you can adjust the size based on the shapes:



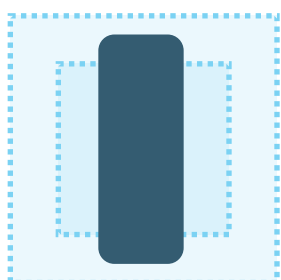
Square icons are the easiest ones to start with. They're for setting the bounding box for other shapes.



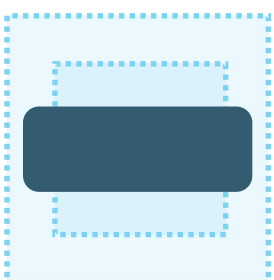
We need to make oval based shapes slightly larger to compensate for the missing space.



Adjust triangular shapes the same way, so that their weight matches the square.



Narrow, vertical icons need to be much taller than the square to have a similar visual weight.



Wide, horizontal icons need to be even wider to compensate for their lower height.



Irregular icon shapes are the hardest to scale right. Test them close to the other icons to see if they're big enough.

Buttons

Buttons

14.

Buttons

Buttons

Buttons

THE BASICS

A button is an interactive element that results in an action described in it. You can bet that if it says "save" on a button, clicking it will most likely "save" something. It's also an essential interactive element of a digital product. It's crucial to get buttons right.

It can lead to a purchase, downloading, sending, and many other vital actions. Digital buttons are also descendants from real-world buttons like in the tv remote, record player, or a game controller.



BUTTONS SHOULD LOOK LIKE BUTTONS

The most important rule while designing a button is for it to stand out enough so it won't be confused with anything else.



By removing elements from a button, its function starts to dissolve and disappear. It becomes decoration or text, losing its actionable qualities.

“

**A button should
look like a button.”**

FAMILIAR SHAPES

We are used to specific shapes and forms that are typically associated with an action. The more our button looks similar to what we associate with buttons - the better. It is the main reason why a rectangle (or a rounded rectangle) is always the safest choice for a button.



These objects will be identified as a button right away.

An exception to the rule is a text link. A link has the same general function as a button, only lacking the shape. Due to its online popularity, it is also really straightforward for the users. Blue and purple text links - dating back to the beginning of the world wide web are one of the most common UI patterns imaginable.

DOWNLOAD

DOWNLOAD

Blue and purple text links are one of the fundamental design truths. Most users understand them right away.

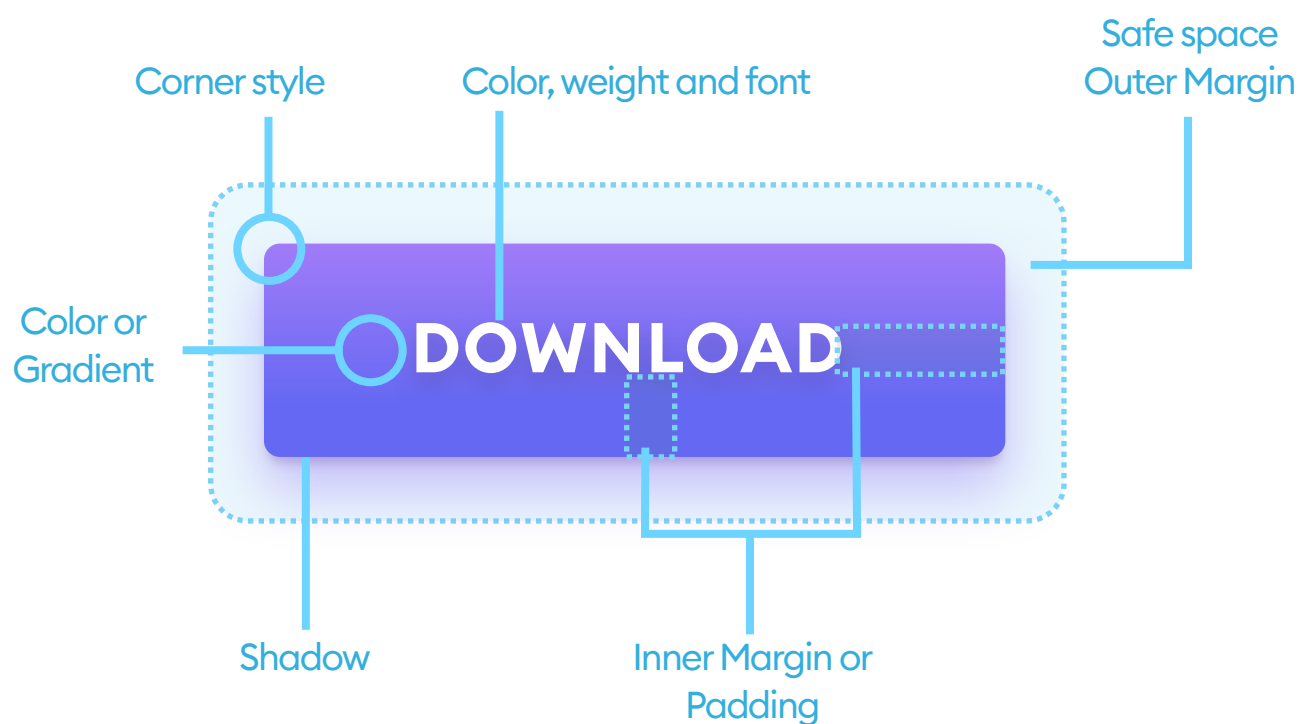
Other shapes and forms (triangle, circle, organic) are not as recognizable to the user. Please proceed with caution and use them only when the general style of your product requires deviating from the norms.



Elements that can take much longer to understand they are actionable.

BUILDING A BUTTON

While designing buttons, remember about all of these elements and choose them wisely. Using the brand book as a baseline, think about what kind of buttons will match the brand and fit well within the interface.



You should set the padding and the safe space using your grid base numbers. (See the Layout and Grid Chapter and "The Red Square Method")

In the above example, the left inner margin is twice the size of the vertical one, which is a safe choice for increased readability.



ALIGNMENT

Poorly aligned and unevenly spaced buttons are one of the most common problems of all interfaces. Double-check if your button labels are centered both horizontally and vertically. Create guides if you need to be sure.

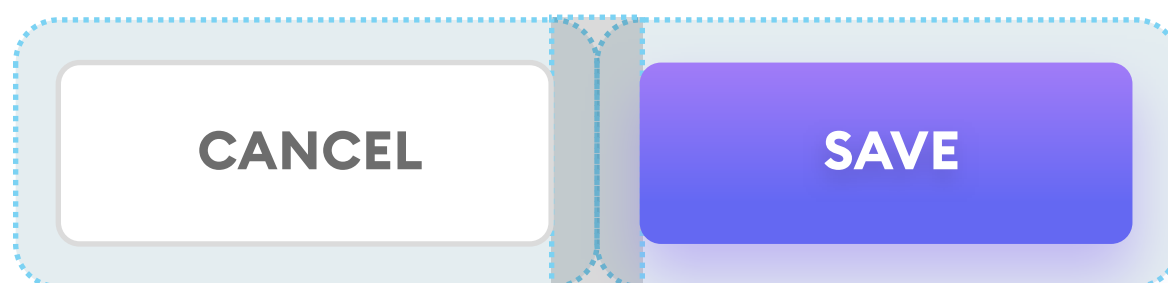


Aside from grid-based methods, there's a safe way of choosing button spacing with multiples of capital W. If at least 1 W fits on each side of the button label, you're safe.

On the sides, it's even better to use $2 \times W$ for increased readability. In general, the smaller the padding inside the button, the harder it is to read and understand.



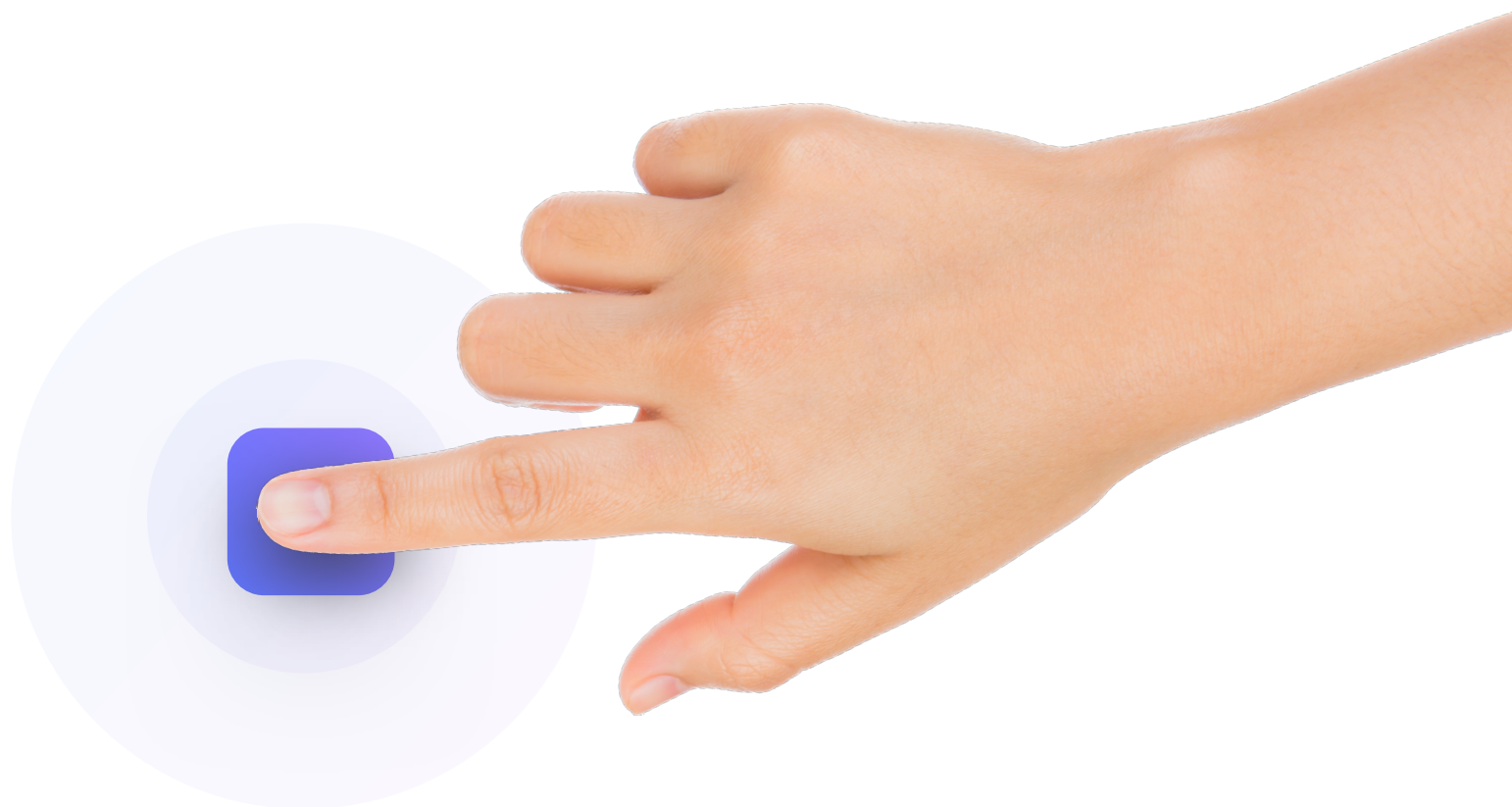
Don't forget about the space around your buttons. If you have a group of them, the safe area should be individual for each one - try not to overlap it!



SIZE MATTERS

Both web and mobile buttons should also have a set minimum size. If your buttons are too small, it will be difficult to tap or click on them. That results in frustration and can lead to users uninstalling your app. The best way is to start with 44 by 44 points for all interactive elements on mobile devices. The sweet spot is somewhere around the 50p height for mobile buttons.

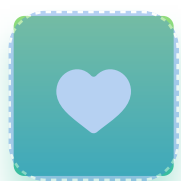
The width (including the Double W minimum spacing) can be any number equal to or larger than the height. In the case of cursor-based devices, 32 by 32 should also work. Remember that even on desktops, the larger the button, the easier it is to use.



This button is too small. The inability to quickly tap or click on it will frustrate your users.



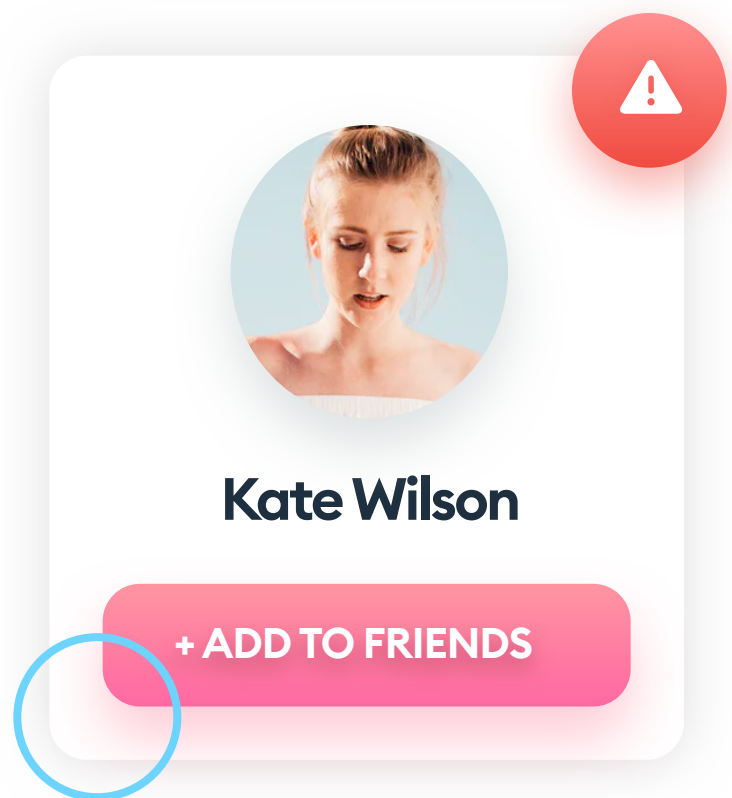
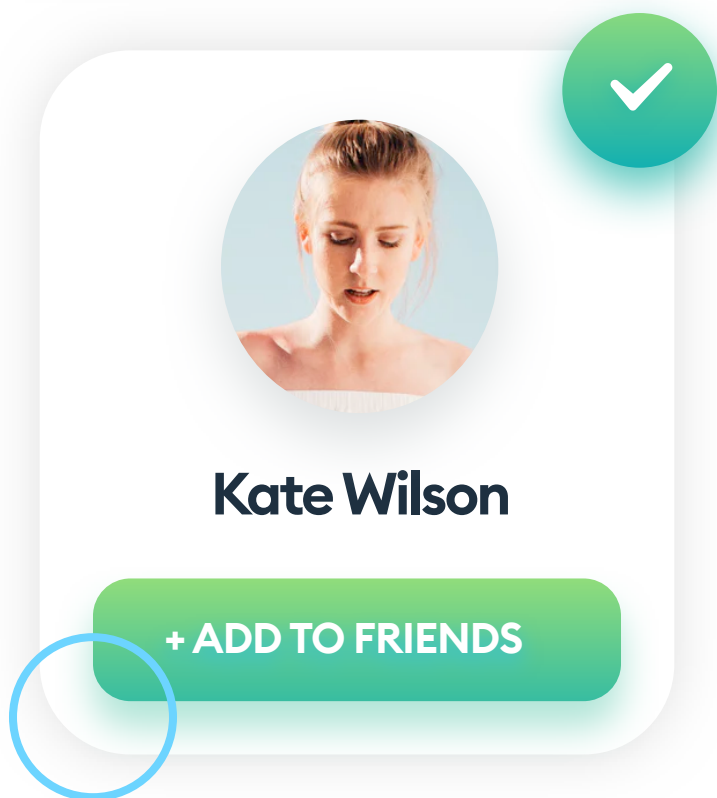
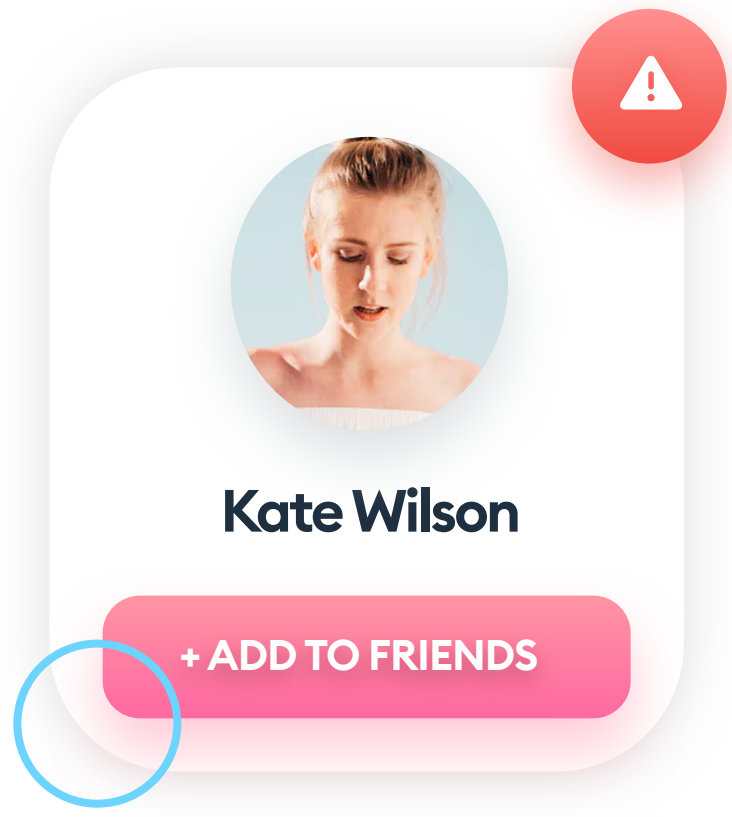
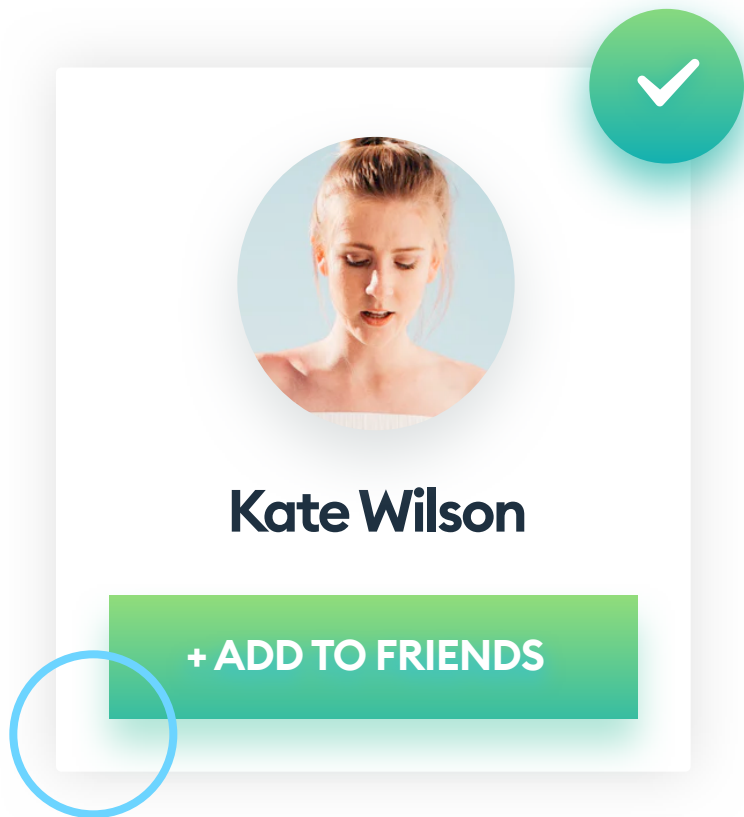
This button is still a bit too small for mobile devices. It probably will be fine on laptops.



This button is big enough for both computers and mobile devices. People should be able to find and activate it with ease.

FIGURE TO BACKGROUND

If you're using rounded buttons, keep in mind to have the right rounded corner ratios towards other on-screen elements. Using the same one for everything will create imbalances in the margins.



The diagonal spacing is the same here like the one on the left side and the bottom. That results in a better looking, cleaner, and faster to process outer edge.

The diagonal spacing is larger (bottom) and smaller (top) than the side spacing. That makes the edge stand out too much and take the attention away from the button itself.

CALL TO ACTION

CTA or Call to Action is an essential, actionable element of a website or app. It's usually closely tied to the sales process.

While designing your CTA button, remember that it has to stand out. The screen's on should not have any other buttons that compete with it visually or catch attention. The best way to achieve that is to make it the only element with a unique color or style on the screen. Don't forget to think about the contrast, so your button doesn't blend with a background too much.

Size also matters. If it's too small, people may miss it while scanning the page. If it's too big, it can suffer from "banner blindness," which is ignoring visual elements that resemble an advertisement.

As with other buttons, the sweet spot is between 40 and 60 points of height, with proper spacing on both sides.



DOWNLOAD THE BOOK!

THE RIGHT MESSAGE

Remember that your CTA needs to "call to action" through wording as well. It needs to be convincing and inviting from the start. It also has to be precise - things like "Sign up for X" or "Download the book" work best.

Try to avoid labels like click here, next, or back.

“

**The optimal height
of a CTA button
is between 40 and
60 points.”**

TEXT LINKS

Text link

Hover



Active



Disabled

A text link was the first way to have an actionable element on a web page. A blue link, underlined is thus the fastest identifiable element of the page. Nearly everyone associates it with action and tries to click it.

With that in mind, remember not to underline non-active elements (especially if they're blue) as an inability to take action can confuse and frustrate the users!

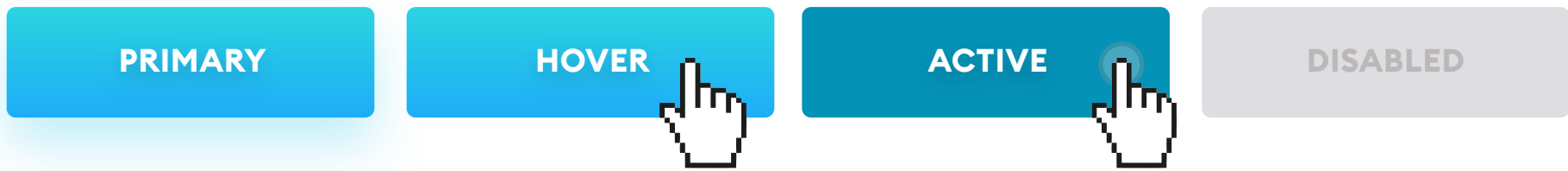
41 SHADES OF BLUE

Link Link Link Link Link Link Link Link Link Link Link
Link Link Link Link Link Link Link Link Link Link
Link Link Link Link Link Link Link Link Link Link
Link Link Link Link Link Link Link Link Link Link

When Marissa Mayer worked at Google, she tested over 41 different shades of blue to find the best combination available. The goal was, of course, for people to click more, and one particular shade won the race. You can now see it in Google's search results. 41 Shades of Blue may sound like a cheesy novel title, but it also shows the dedication to research that makes a difference.

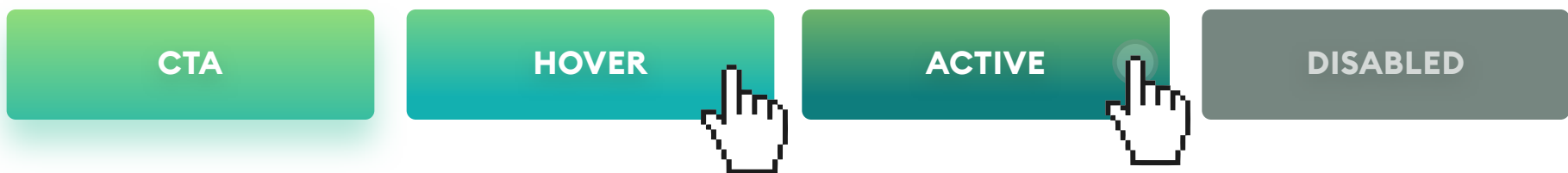
In truth, however, if your link has enough contrast and differs from its surroundings (through color and underline), you can be sure it will be understood.

BUTTON TYPES



The primary is the button for the main, positive actions on each screen – things like OK, Save or Download.

Unlike the CTA button, you can use more than one instance of this button on the screen, but try not to group it with other primary buttons or the CTA.

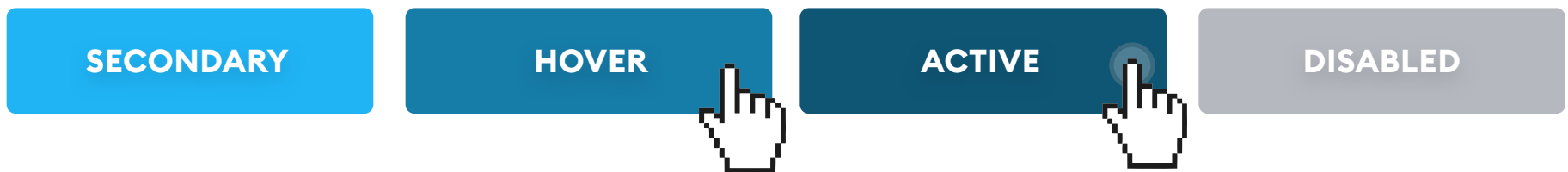


Design your Call to Actions to be visually different than all the other buttons. You can use colors or sizes to achieve that. The goal is for the button to be visible right away and to catch the attention. Because of that, you should only use one instance of CTA on one page. If you are designing a long page, you can duplicate the primary CTA near the bottom as a "catch-all" type button.

If you try and use more than one "strong" button, your page will be converting less, as the user's attention divides between many elements.

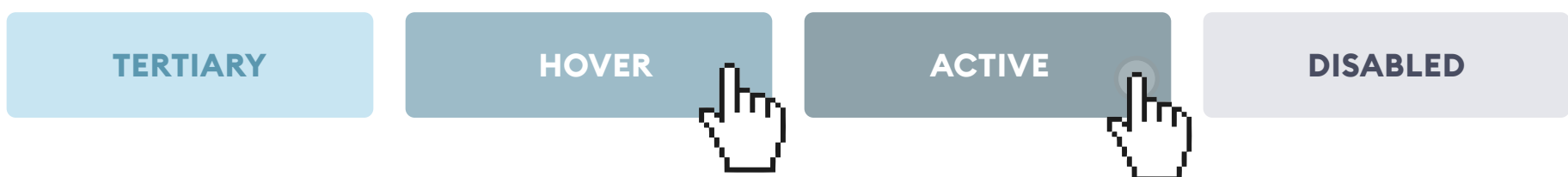
CTA should also deliver the most important actions of your interface like "Buy now" or "Sign up."

BUTTON TYPES



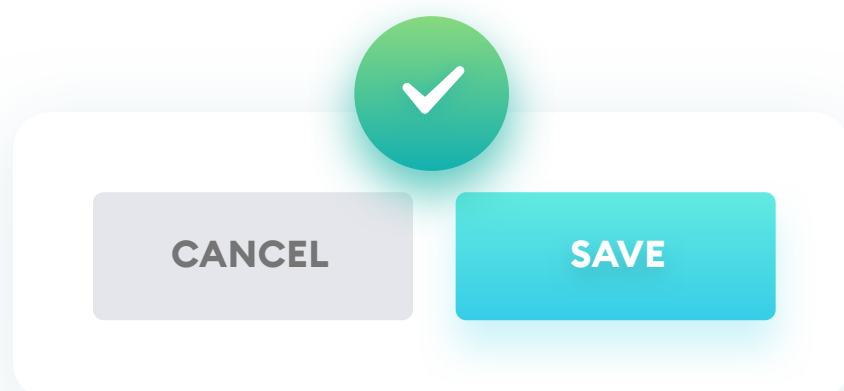
Secondary buttons can be used for all the less critical actions or if we have many buttons with similar importance on one screen. If you're unsure which button to choose, this should always be your safe choice.

Make sure this button does not compete visually with either the primary or the CTA, and you're good to go.

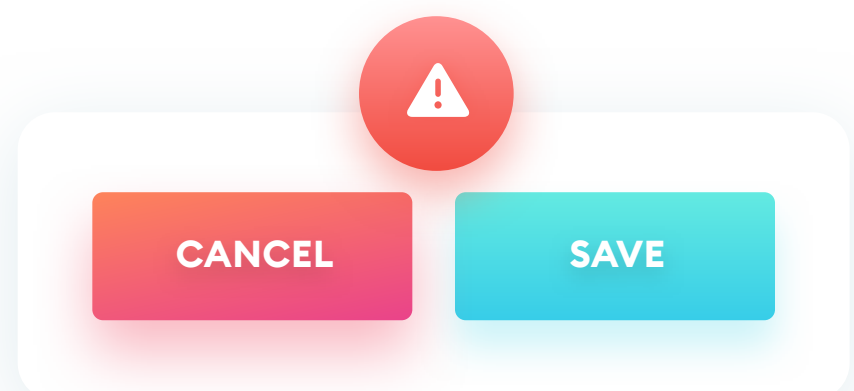


Tertiary buttons are best for negative actions like "cancel" or "revert." You can also use a text link or a ghost button as your tertiary, instead of a rectangle.

There's a notion that all negative actions (like 'remove') should be a red button. In most cases, the tertiary will do the trick. If you're unsure how to design negative actions, think about it like this: Are those negative actions desired? If they're not the core of what your project has to accomplish, then making them red would take attention away from other, more critical on-screen elements.

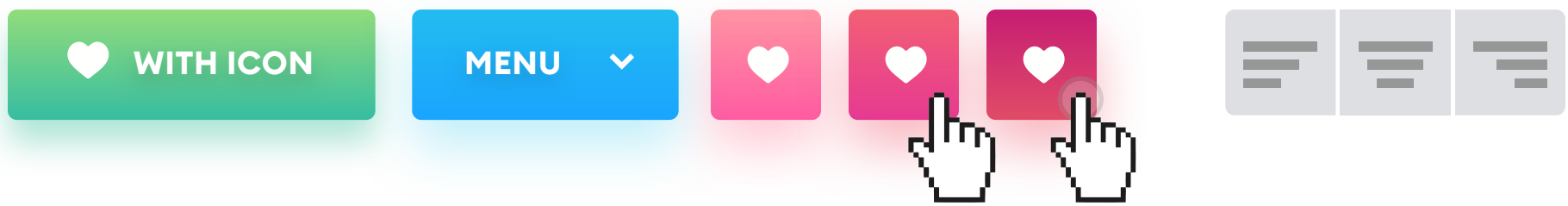


The negative action is quickly identified as less important than the positive one.



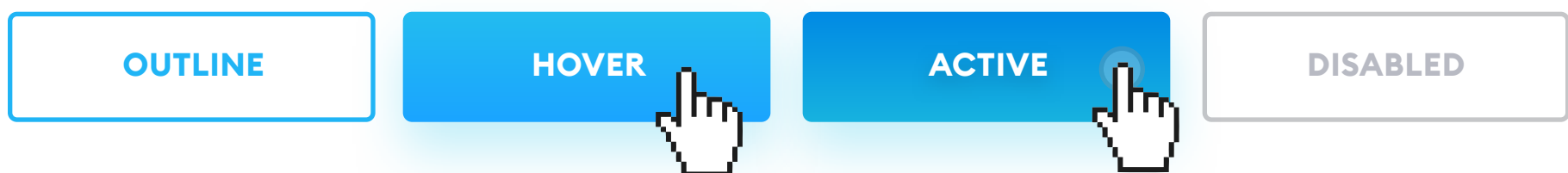
The negative action is more eye-catching than the positive one.

BUTTON TYPES



Icons on buttons are mostly used in apps to condense more functionality within limited space. If there's a text label next to an icon, the perceived importance of that button is higher. In general, however, it's best to avoid having too many buttons with only the icon. Even the most well-known symbols can be interpreted differently by your users and confuse them.

One of our most famous examples is the heart symbol. In one particular target group, we researched most of the users thought the heart meant it was a healthy product.



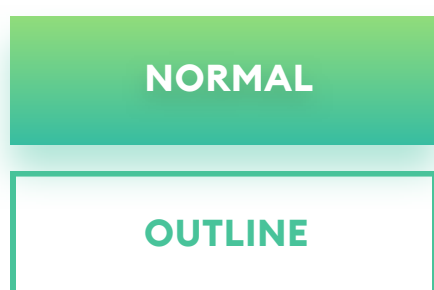
Outline buttons, also called ghost buttons, are usually reserved for less important actions. They're often used as a tertiary button as they don't compete as much visually with primaries and CTA's.

You shouldn't use a ghost button as your Call to Action, because it captures less attention than a button that's filled. Because of that, it takes slightly longer to process, which may lead to lower conversion.

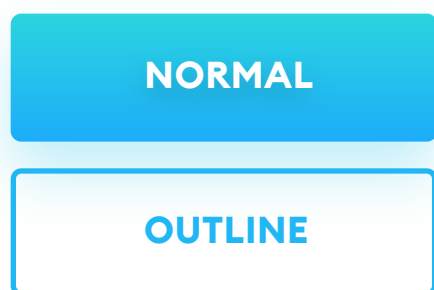
Their active states (hover, active) work best when filled with color.

CORNER RADIUS

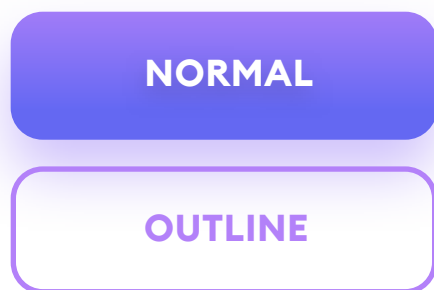
Buttons can also vary through their corner-radius. Remember that each project is unique - if your brand image has only sharp corners, having rounded buttons will make the UI inconsistent. Mixing a couple of different corner radiuses within one interface can also lead to consistency and readability issues unless done on purpose.



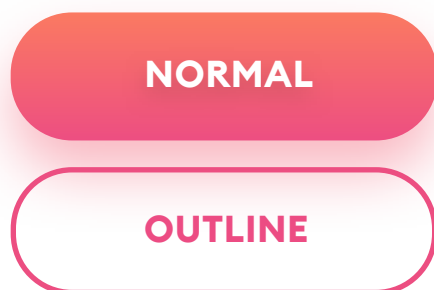
Sharp corners fit brands that have similar shapes in their logo and other elements.



Slightly rounded corners are likely the most popular and also the most user-friendly choice.

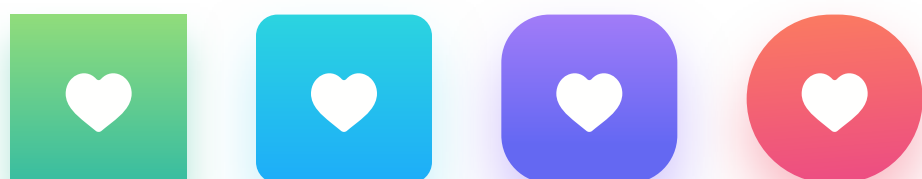


Larger corner-radius can make it harder to align the text above and below the button.



"Pill" or the highest possible border-radius works well for CTA's.

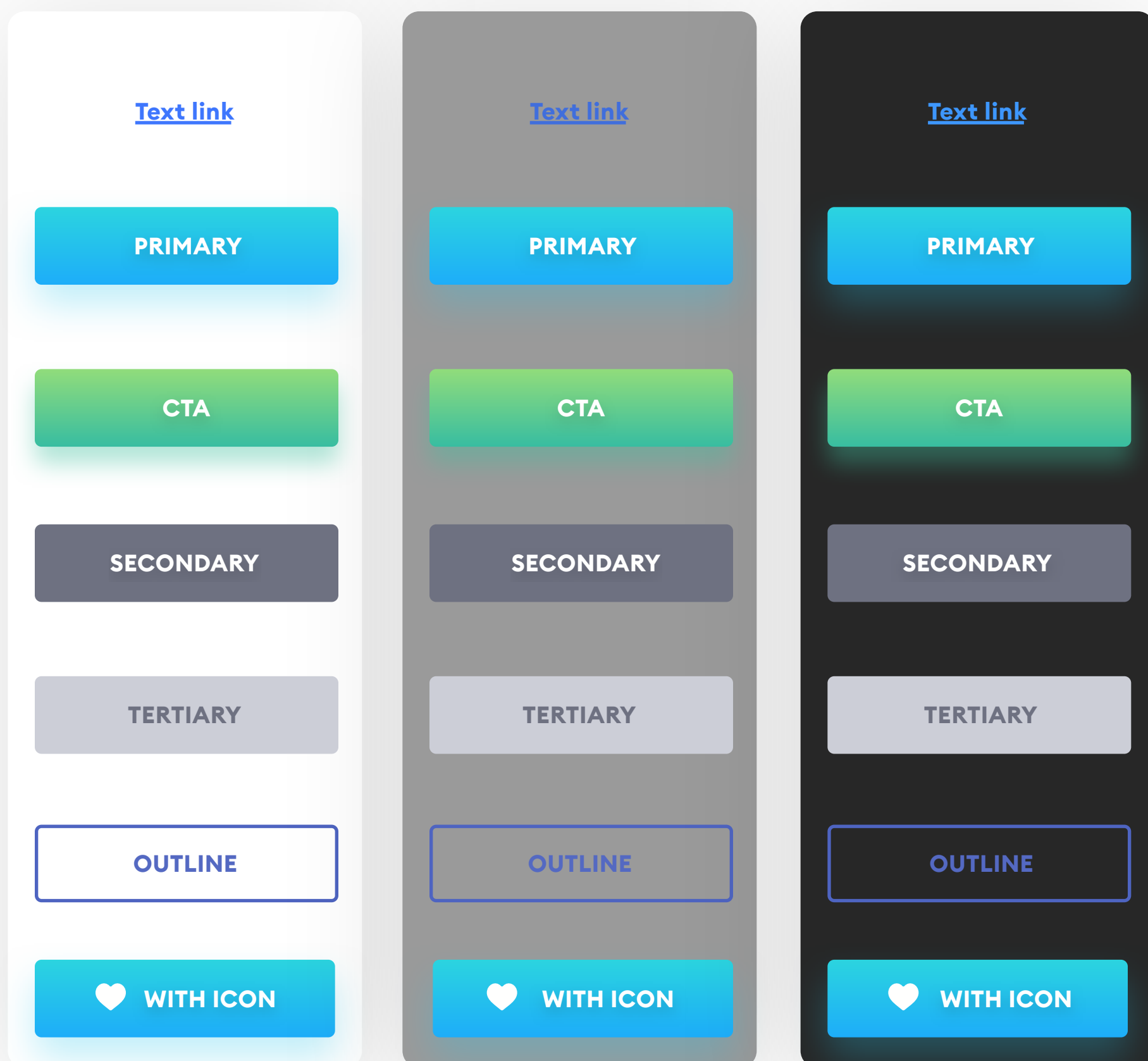
The same rules apply to your icon-buttons.



VISIBILITY

It's good to design your buttons in a way that they work well on various backgrounds. A lot of apps and websites are adding "dark modes" now, which results in at least two completely different versions of the product.

The colors and saturation of the buttons should work well with most of the backgrounds. You can create a simple tester like below and always aim for high contrast and visibility.



BEST PRACTICES

Important buttons also work well with icons. A basket or a cart icon quickly identifies a checkout. However, that is the case only as long as the word checkout is used as a label.



A right-facing arrow or chevron placed after the button label makes the resulting message stronger. The user is more inclined to click and "proceed." It works well if you'd like to make your CTA stronger.



Buttons with shadows are also more "clickable" and noticed much faster than flat ones. Add a subtle drop shadow to make it stand out from the background.



The button on the right is more visible and will be a stronger Call to Action.

BEST PRACTICES

Rounded buttons are considered more friendly and positive than sharp edges. At the same time, however, they make it a lot harder to design content around them. If you have left-aligned text just above the button, the more rounded the corner, the less that text will visually fit. It makes you feel as if the left margin is in two places at the same time.

If the button is only slightly rounded text above, it seems to fit within the perceived left margin.



The text over a more rounded button begins to "bounce" away to the right visually.



Text over a completely rounded button loses visual alignment and seems to be outside the grid.



Pill type buttons work the best when the text above them is centered or removed altogether.

BEST PRACTICES

Aligning icons on buttons is one of the hardest tasks a designer can face. In many cases, it depends on the font-weight to icon weight ratio and their "optical size."

However, you can try and use this simple solution that will work fine in most cases.



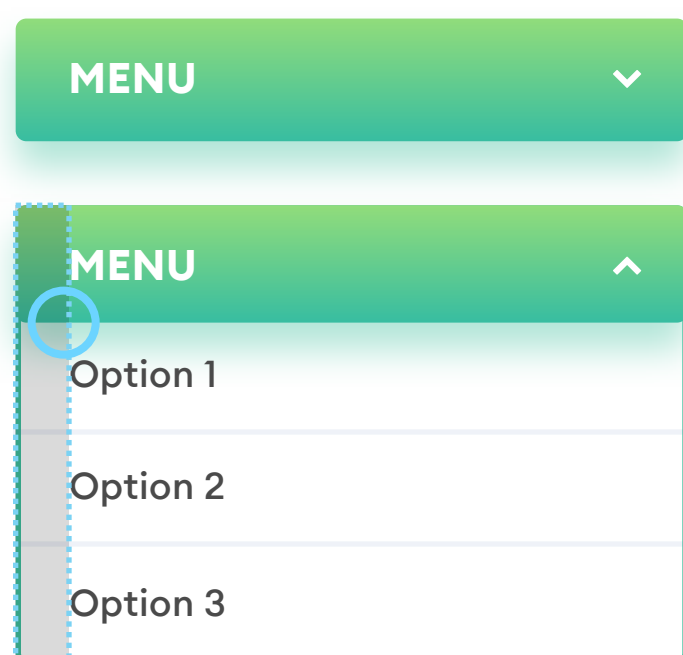
Create a square or a circle the size of our button height. If it's a pill or rather sharp-cornered button, just align it to the right side of the shape. Then create another smaller object inside of it with its spacing being the height of our text. Then fit the icon right in the middle and try not to make it larger than the smaller circle. In the case of chevron, text-height works for chevron height as well.



If our border-radius is larger than 8 points (but not quite a pill), it's best to align our square to the place where the border starts to bend. If you need to use a more complex or larger icon (like a shopping cart), make sure it's not bigger than our smaller container.

BUTTON STYLE DROPDOWNS

Rounded corners are also problematic while designing a button/dropdown combo. It works the same way as a normal dropdown, but its initial state looks more like a button with a full color or gradient fill. When you expand the button, it creates a shape that distorts the left edge, even more, making it harder to process the text. You can see that clearly in the second image.



If the corners are sharp or only slightly rounded, we end up with a natural, easy to read expanded version.

The text margin allows the user to go down in a straight line without shifting focus away.



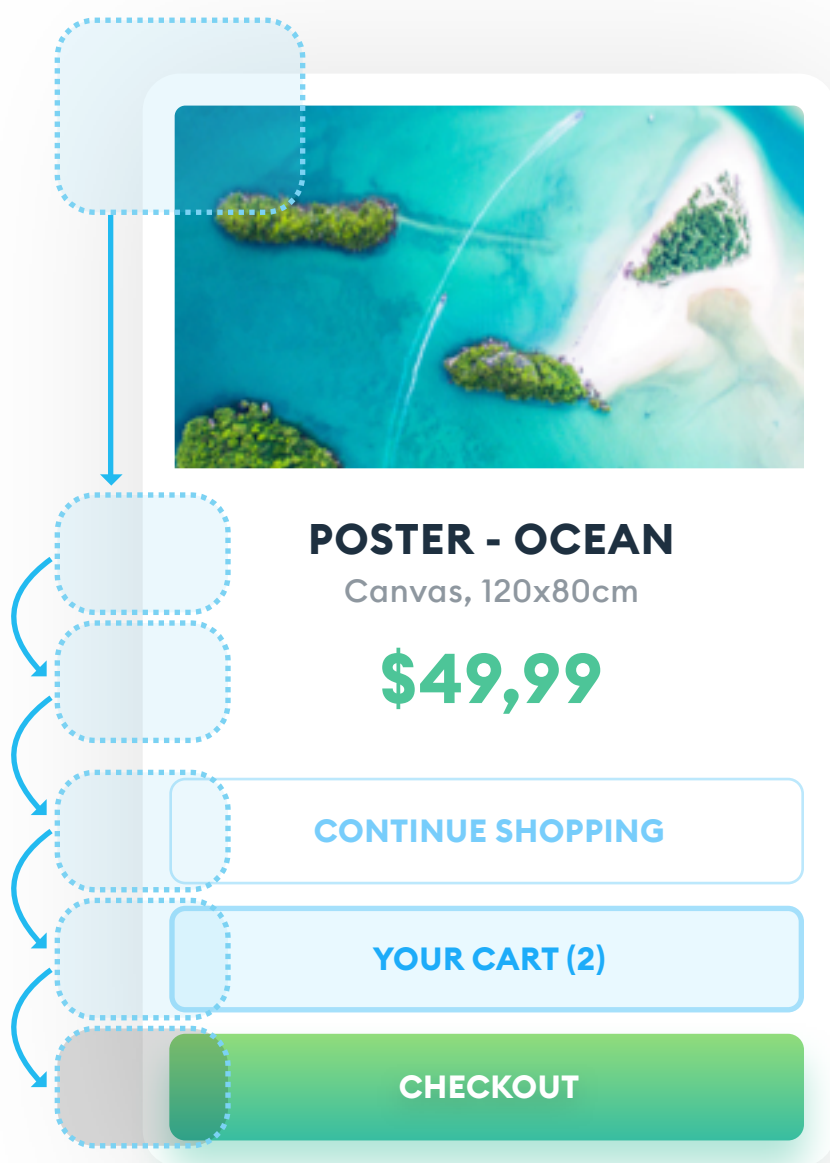
If the corners are completely round, we end up with a shape that captures our attention and swings it between two separate grid lines.

That impairs readability as our focus shifts from edge to edge while scanning the image.

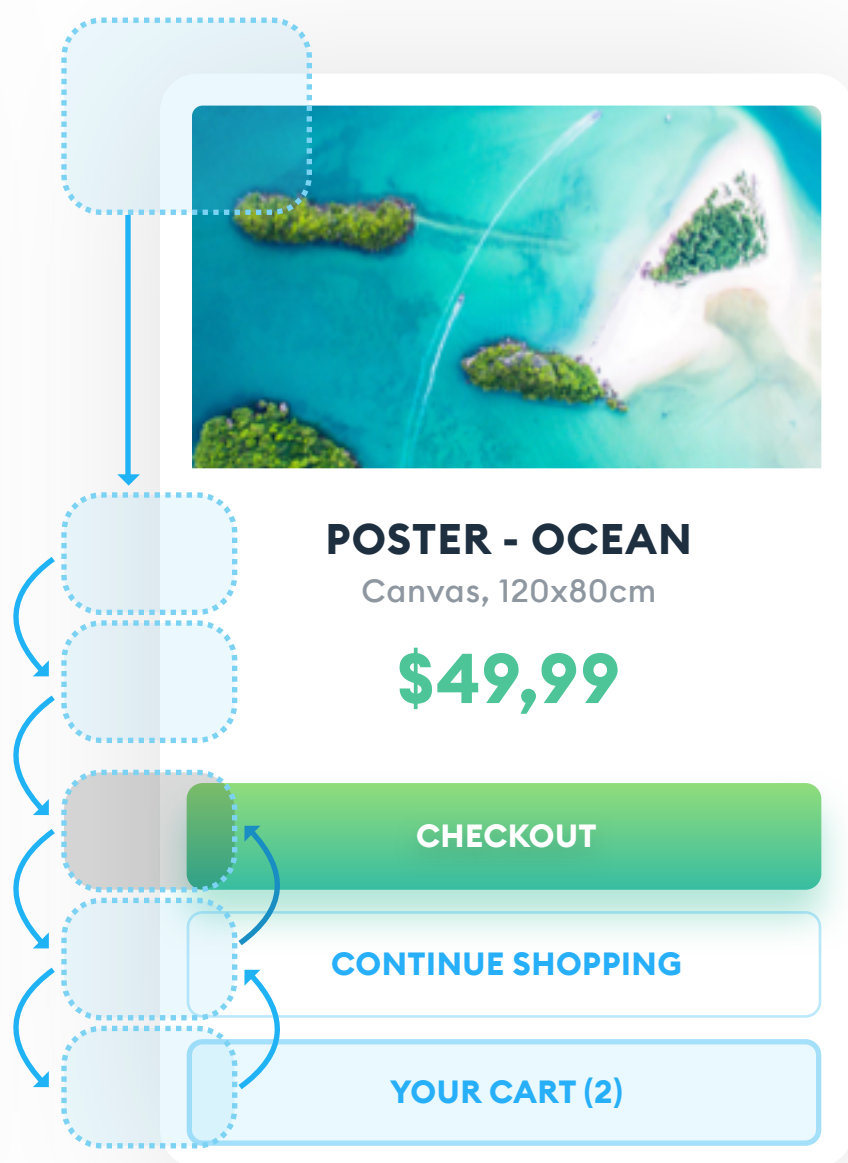
HIERARCHY

Buttons should be in places where the user expects them to be. That usually means below the descriptive elements and often aligned to one of the sides.

That matches the way we scan things - from top to bottom along the left edge (which is also called an F-pattern). If you want to stay safe without bending the rules, pick the first image.



By scanning the card from top to bottom (along the left edge), we naturally end up on a strong element, which in this case, is our CTA button.



If the most pronounced visual element is not at the bottom, our eyes travel back and forth along the edge, which makes for slower processing.

AUTO-ALIGNMENT IN DESIGN TOOLS

If we want to center our label on a button correctly, we need to make sure that both the label and button height are either even or odd numbers. If one value is an odd number and the other one is an even one, you will never be able to center them correctly.

We can try to use auto-alignment tools like auto-center, but sometimes even they end up missing a pixel or two.

Sadly entering an even number for both the container and label height is often not enough. Fonts are interpreting the numbers differently (depending on the font construction) and render the resulting text slightly larger or smaller than we anticipated. That's why try and use little squares to measure the height from the top and bottom (and then from the sides) to be sure it's the same on each side. The buttons below both have an even number as their height and an even number for the font-size. One of them had to be modified by hand to fit perfectly instead of missing one pixel.



We used auto-alignment in both buttons. The example on the right, when checked manually, shows that there's a one-pixel gap on one side. In a case like this, you need to adjust the button height to match the label manually.

“

**Don't trust numbers
alone. Test your main
buttons manually.”**

CONTRAST

The text to background contrast ratio is fundamental to get right. If it's too small, the user can skip it entirely while scanning the screen. If the contrast is too high, it can become an eye-strain - especially with high brightness displays.

Most design tools allow you to add a plugin (often free) to check the contrast ratios between elements. Aim for at least AA, but also use your judgment as to what will work OK. We'll talk more about the contrast in the chapters on colors.

Remember that some of your users may have impaired vision. Even if they don't, why make their lives harder?

Which one do you think will get noticed first?



The one on the left can be barely visible on low contrast screens or by visually impaired users.

TEXT CASE

A label written in all-caps is slightly slower to process than title-case one. It of course, depends on the number of words, their length, and even individual letters.

Doing buttons ALL-CAPS is not wrong, but try to limit them only to short one or two word labels. Words like "DOWNLOAD" or "SAVE" are safe to use either way. It's also closely tied to how the brand itself communicates in its marketing materials and branding. If it includes ALL-CAPS, it's good to be consistent with those rules.

The title case is slightly more readable, especially with more wordy labels. Try it for things like "Sign Up For Our Newsletter" or "Download the Free Ebook."

Try to avoid entirely lowercase button labels, though, unless they're brand defined.



The label on the left will be slightly faster to read and process. The one on the right will be easier to align correctly inside the button. The choice is yours.



Cards Cards

Cards

15.

Cards

Cards Cards

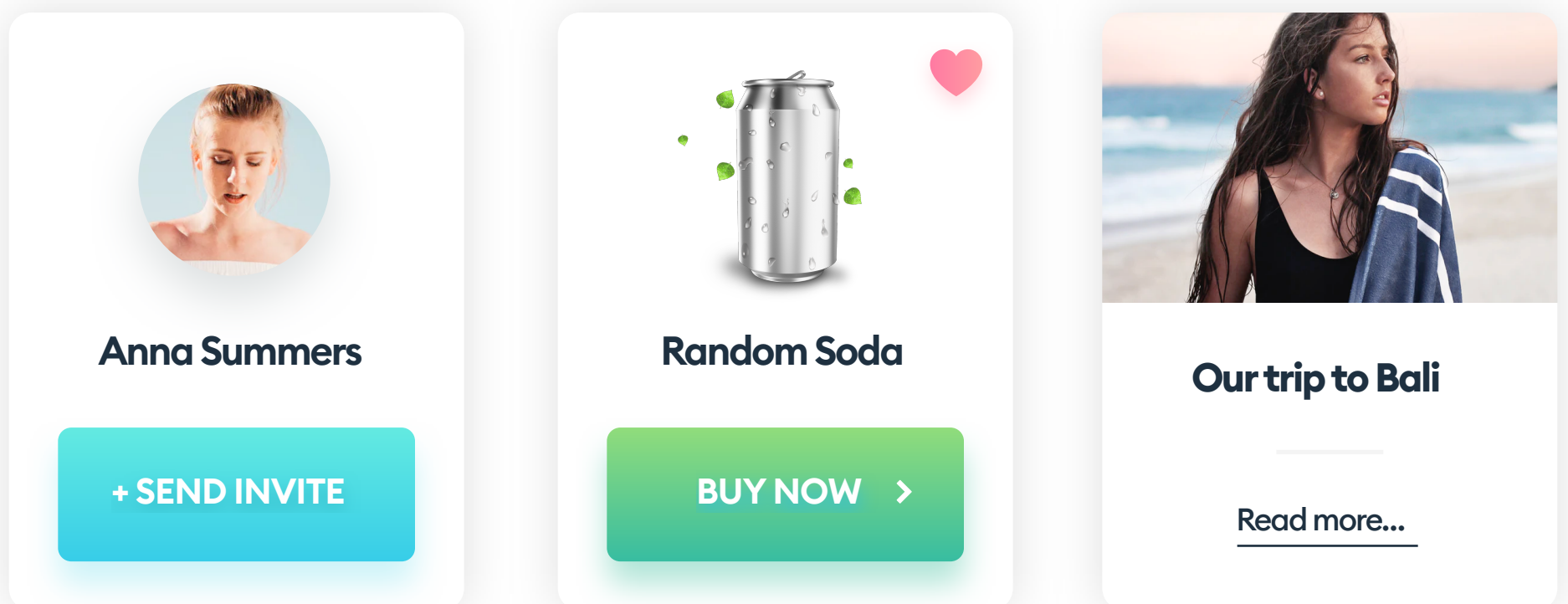
Cards

Cards

INTRODUCTION

Cards are one of the most popular ways of showing content in an interface. They can display products, information, people, or actions. Most of the products out today, especially after Material Design, use some form of card-based UI.

A card is usually an excerpt from the full detail page it leads to after tapping or clicking. It can contain text, buttons, icons, and photos to help the user better decide which one to choose.



When designing a card, start with adding the essential content to it first. After you have all the elements, you can start finding the best place to position and style them.

Avoid using non-essential information on cards (filler content) - they should be concise and to the point. Stick with just one button and three lines of text, moving everything else to the following detail view.

HOW TO SHOW CARDS

Usually, cards exist within a horizontal or vertical scrolling carousel. We can also stack them on top of each other, or use a grid. It can either be one with equal heights or a more organic "masonry" grid, which can have different heights for each card, while still keeping them in a column.

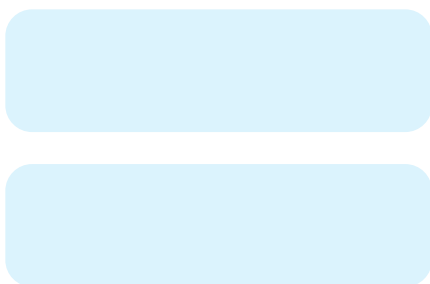
Try to make your cards accessible to all popular input types (cursor, touch, or arrow keys).

Here are the most popular ways to display cards:



HORIZONTAL

Horizontal cards are the base for side-scrolling carousels.



VERTICAL

Vertical cards are best for catalogs and other data-heavy presentations.



STACK

Stacks work well for simple, action-based sorting (like swiping them away to the left or right).

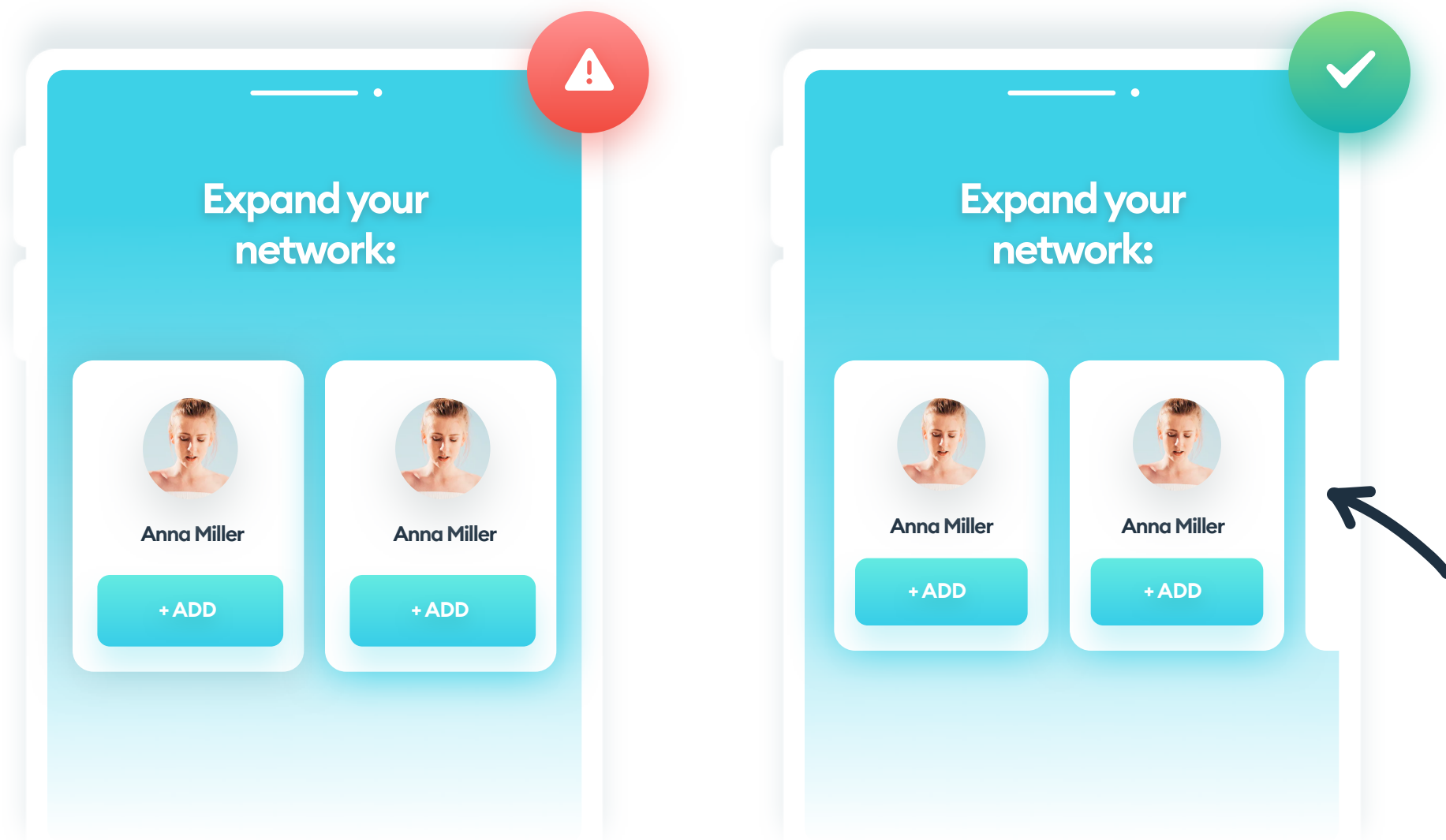


GRID (MASONRY)

Both the masonry and even grids are suitable for news portals and e-commerce sites.

SCROLLING CARDS

If your cards are in a side-scrolling carousel, try to have the last one slightly cut-off by the edge of the screen. That will help the user understand that there is more content behind the screen to access.



NO CUT-OFF CARD

This kind of card layout is not suggesting the user can move them horizontally.

ONE CARD CUT-OFF

With one card sticking out from behind the screen edge, we automatically know there's more content there to scroll.

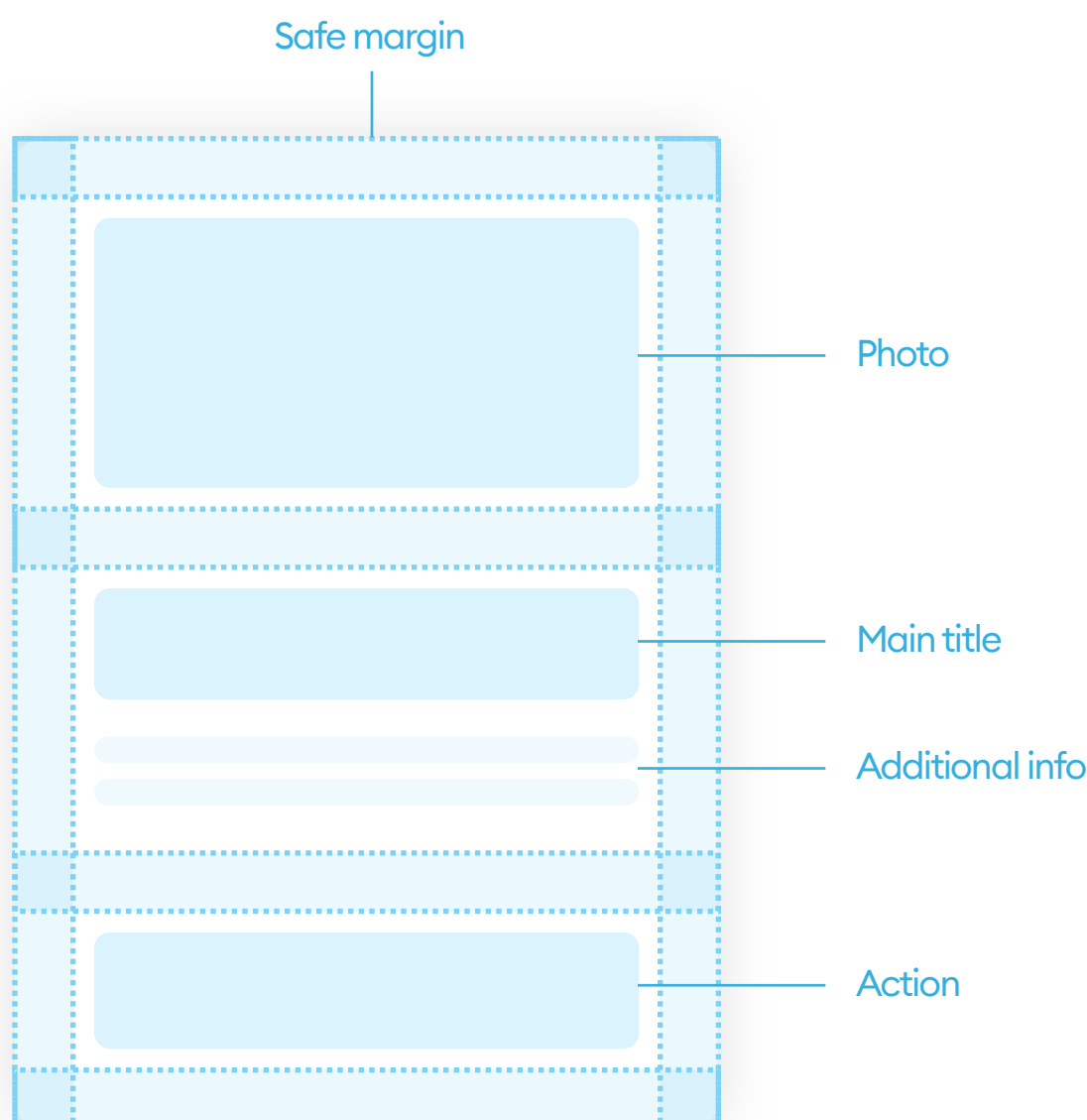
DISSECTING A CARD

Layout and hierarchy of our card directly influence how easy and fast it will be to understand their purpose and take action.

Identify the most critical information (like the price), the secondary info (like the product name), and a label (category) and design them in a clear, hierarchical way by grading the emphasis (like using a bigger, bolder font for the important info).

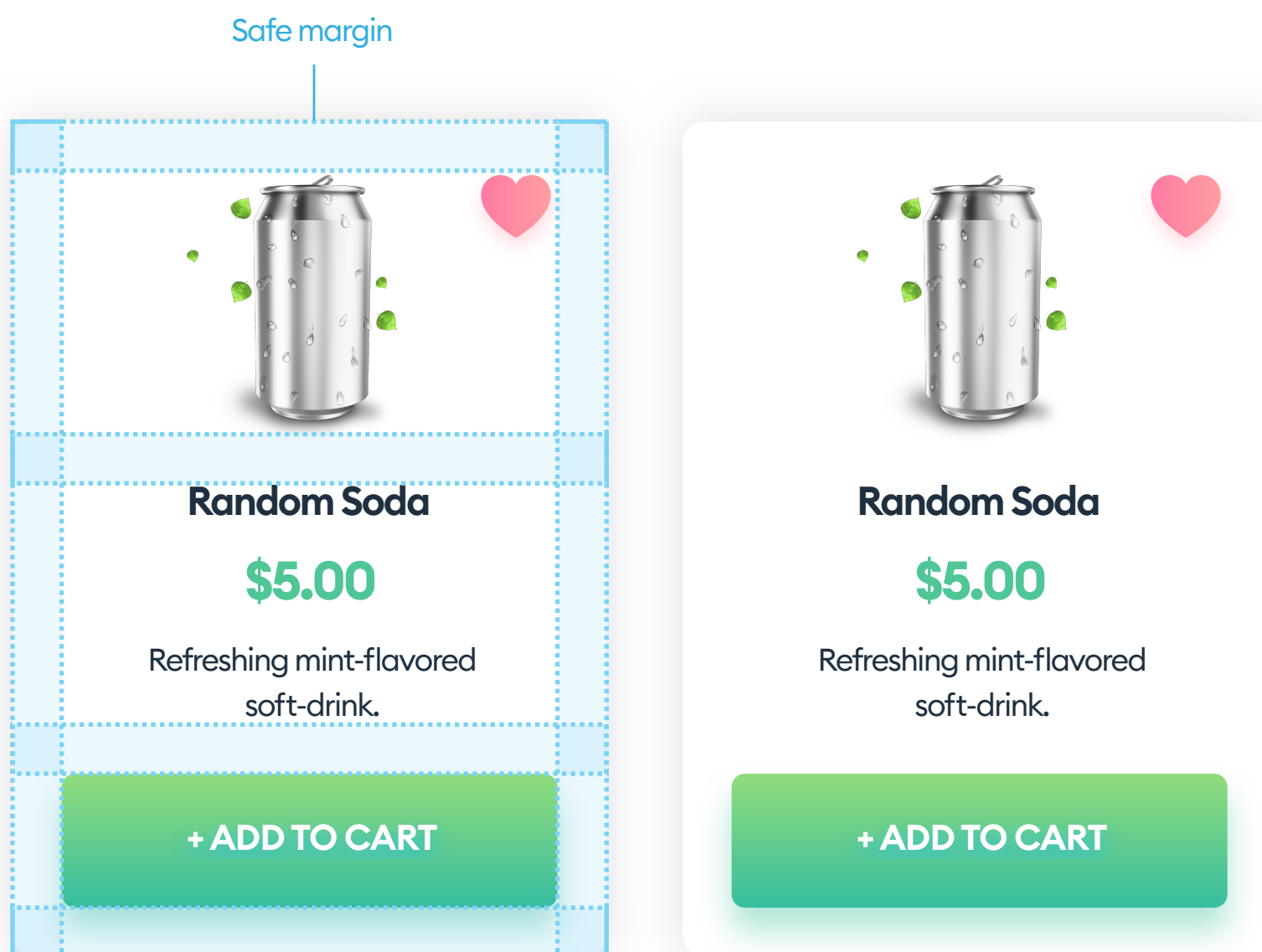
When designing a template card, do it with the longest possible text in every field (like the longest product name, highest possible price, or the longest last name on a user card). That will help you plan the best way to display all kinds of data.

Keep your margins consistent and uniform, stacking related information together for clarity.



CARD STRUCTURE

Once we get the content hierarchy right, you can start adding the visuals. These can include product photos, icons, or a Call to Action button.



While we don't recommend it, cards can have additional action buttons (secondary and tertiary). They can also contain action icons (usually in the top corners).

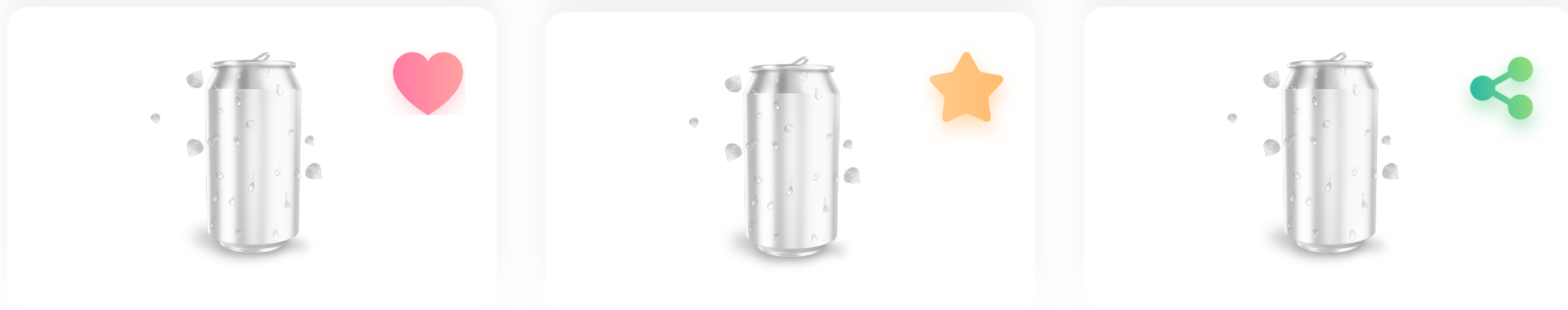
Keep your cards uncluttered and straightforward. Remove all non-essential information for higher clarity.

Unless they're the main object of the card, photos and illustrations shouldn't take more than half of the vertical space of the card. The exception being photo galleries where the images are an essential element.

ICONS

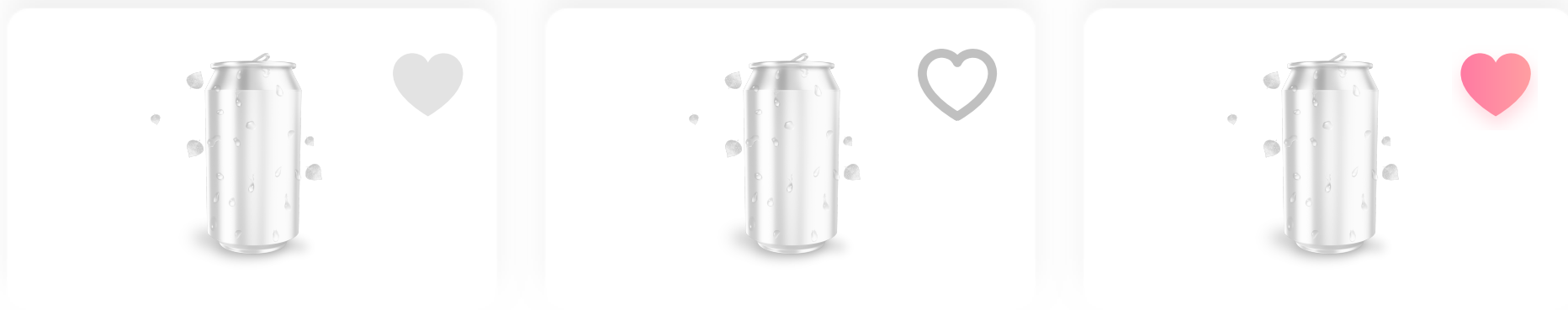
Cards can use icons for non-essential actions that won't compete with the CTA too much. These include things like "add to favorites," "share on social media," and similar.

Icons usually occupy the top right portion of our cards, because the bottom is the best place for the primary CTA which should be catching the most attention.



Action icons can either be external (showing a new screen, like a share-sheet for social sharing) or internal, where the status of the icon changes once we activate it (add to favorites).

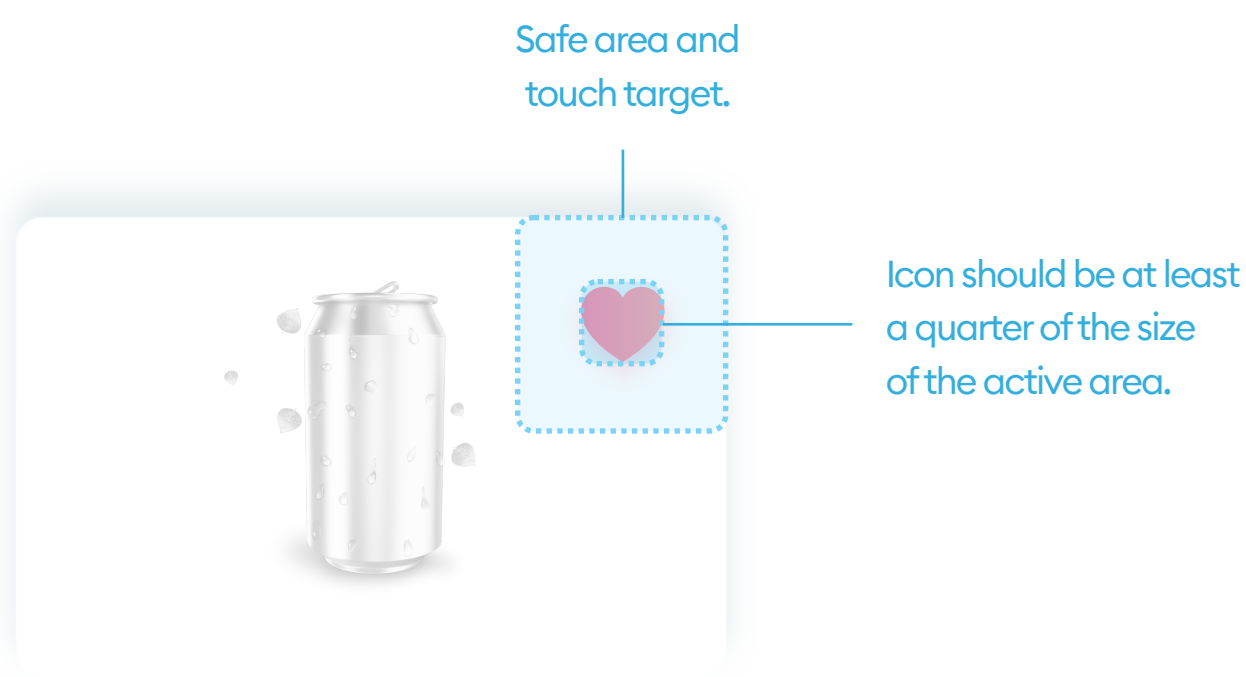
All internal icons need to show which state they're in precisely. If the product is already in favorites, the icon needs to show it by a colorful fill. If it's not, it can be greyed out or outlined.



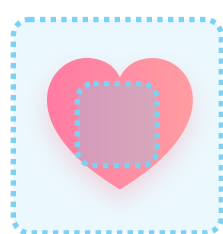
ICON ALIGNMENT

The icon should be the right size, with a proper margin from the sides and big enough active area, also known as the touch target.

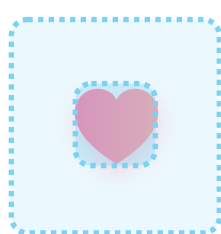
The size of the icon should come from a grid base number we set up beforehand. If we're not sure about the size, the safe choice is around a quarter of the touch target. With the typical TT of 44x44 pixels, the smallest, yet still, a good looking icon should be about 20x20.



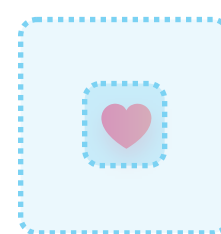
If too big, the icon can grab way too much attention away from the CTA. If too small, it can disappear or look like a decoration, and not something you can use.



Too big



Just right



Too small

CARD SHADOWS

Because of the assumed position on top of the background, cards should cast a shadow. The best choice for a realistic-looking shadow is assuming the light is coming from the top, which translates to a positive shadow offset on the Y-axis.

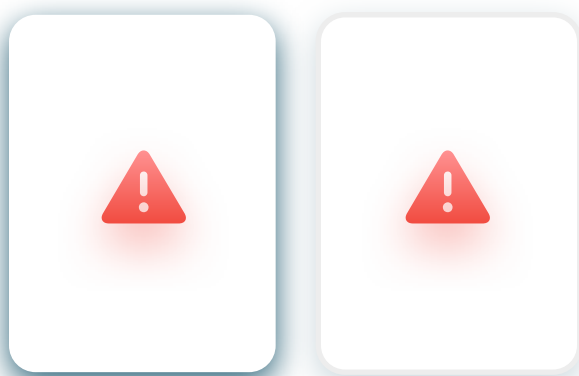
The horizontal offset (the X value of our shadow) should stay at zero in most cases. We usually look at the screen from the front, so a side-leaning shadow can negatively affect the perception.

Another way is to skip the offset altogether ($x=0$ and $y=0$) and focus on blurring the shadows equally on all sides.



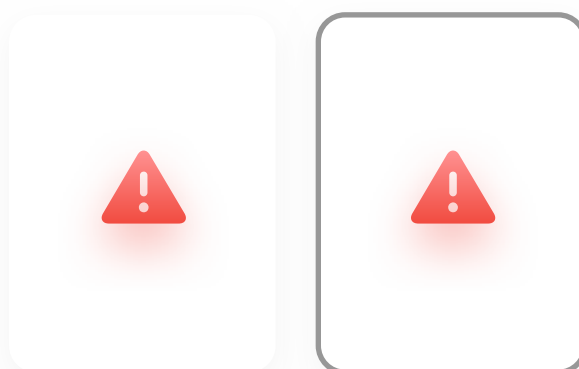
PLEASANT, GOOD LOOKING CARDS

A soft, subtle shadow that makes the card appear as if it's floating over the background.



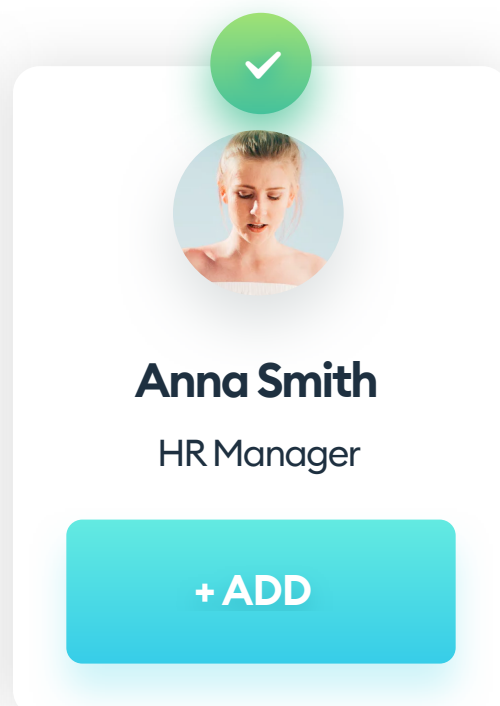
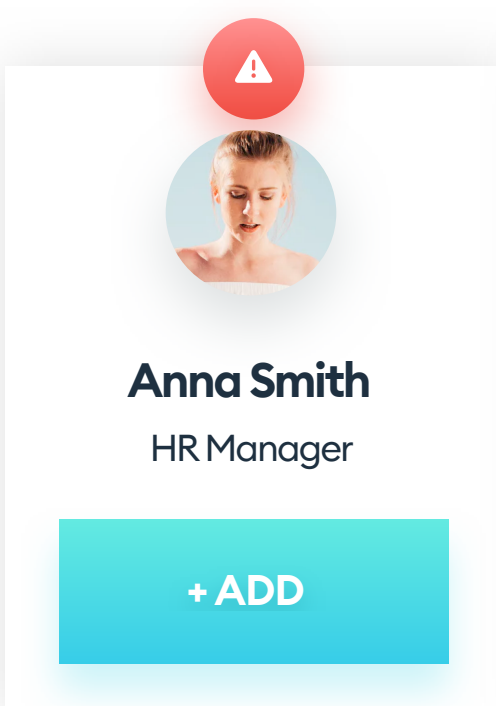
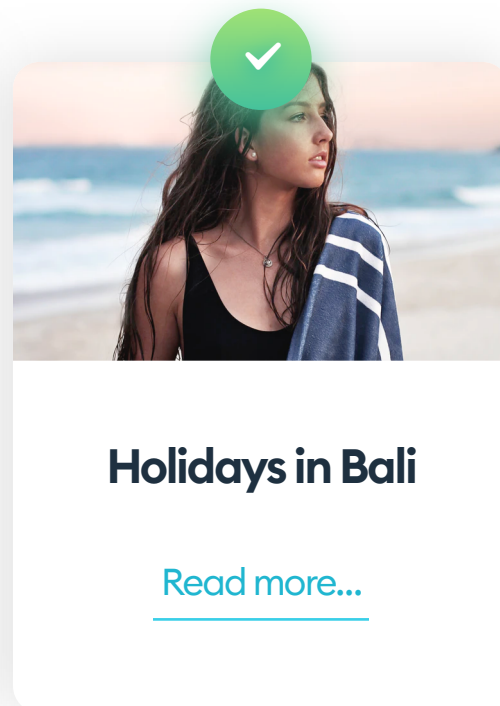
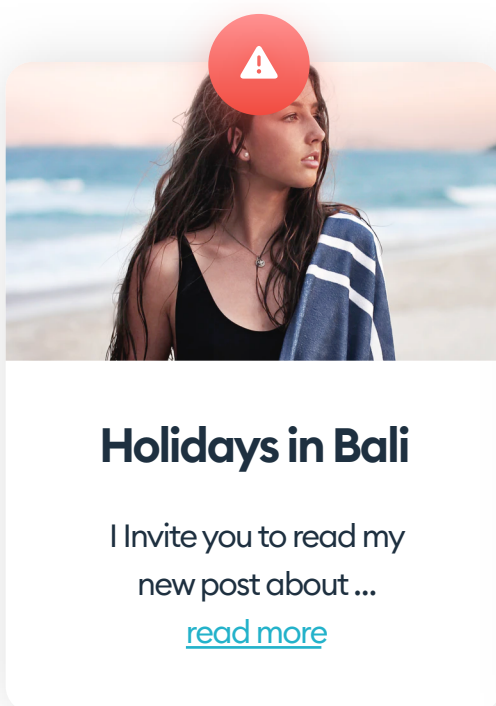
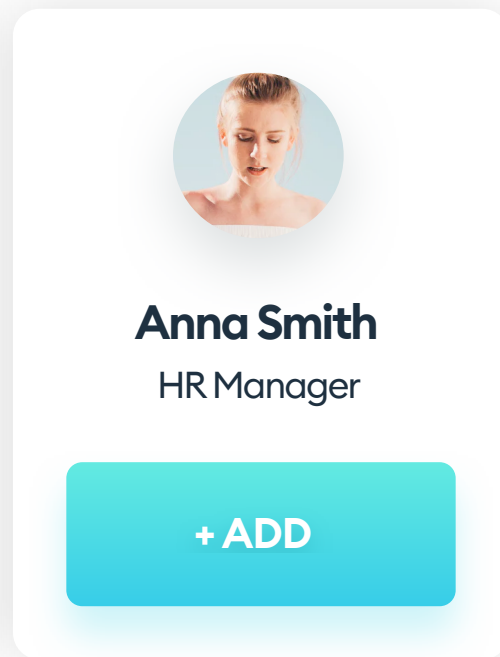
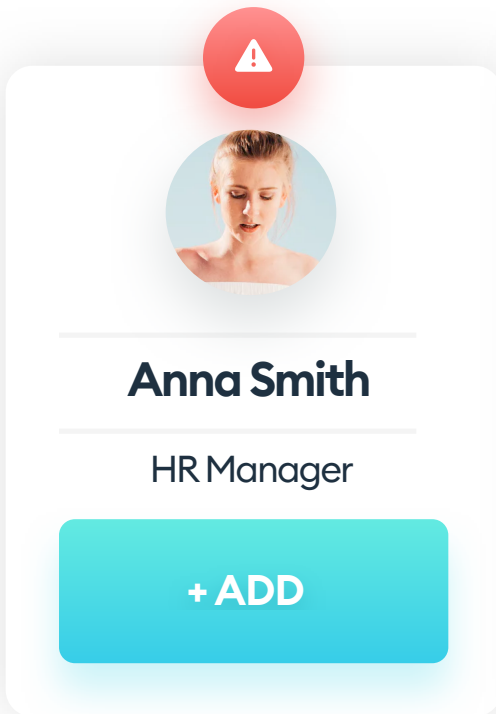
MOST COMMON PROBLEMS

If the shadow or stroke is too strong (high contrast), it takes the attention away from the contents of the card.



Too little contrast will result in a blurry border that makes the entire card harder to discern.

GOOD PRACTICES



LESS IS MORE

Use the right margins and font sizes to set the layout and hierarchy in place. Avoid unnecessary decoration (like separators or other decoration).

TEXT LINKS

If you plan on using a text-link instead of a button, make sure it has enough space for a comfortable touch.

ROUNDED CORNERS

Our eyes prefer rounded corners to sharp edges. Make your cards and buttons more friendly with a small border-radius.



16.

Tables

and Graphs

Tables and Graphs

SHOWING DATA

We are living in the information age. Data is collected all the time and about everyone. To better understand and process all that information, we need to make sure it's clear and readable.

Tables are a popular choice for showing large sets of data in a structured form. It's good to know the basic rules for a readable, user-friendly data presentation.

Even a very dense, complex table can be easy to scan and look good if we design it the right way. Here's an example of a typical data table in a digital product.

Coffee	Size	Price
Espresso	S	6,00
Doppio	S+	8,00
Cappuccino	M	10,00
Latte	L	12,00
Flat white	L	12,00

A table is a grid of columns and rows, allowing for easy scanning, comparing, and analyzing large data sets. While designing a table, keep in mind that the essential part of it is the actual data and not the decoration.

All interface elements that are not data should only be used when necessary. Avoid decoration and ornamentation without function. These include shadows, backgrounds, icons, colors, and in some cases, even divider lines.

“

**Data is the most
important part
of tables, so avoid
unnecessary
decoration.”**

BUILDING A DATA TABLE

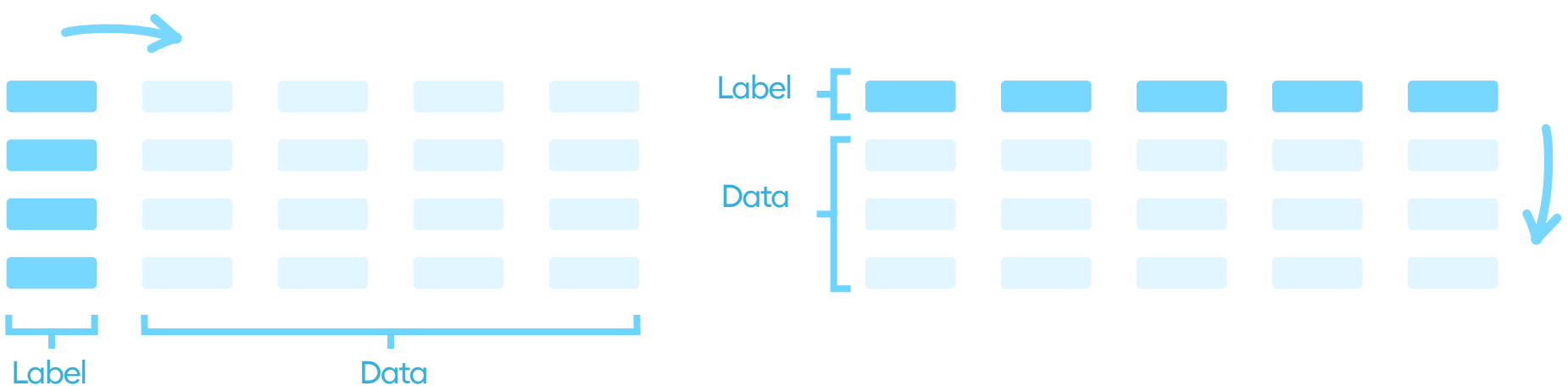
A typical table has two main elements: a label (for either the rows, columns, or both) and the actual data (text, numbers, or images).



A diagram showing a table with three columns and three rows. The first row is a header with labels: 'Coffee', 'Size', and 'Price'. The second row contains 'Espresso', 'S', and '6,00'. The third row contains 'Doppio', 'S+', and '8,00'. A bracket on the left labeled 'Label' spans the header row. Another bracket on the left labeled 'Data' spans the two data rows.

Coffee	Size	Price
Espresso	S	6,00
Doppio	S+	8,00

We can present the data in columns or rows, depending on the type and amount of information. More complicated data sets can have labels in both the rows and the columns for clarity.



If our column has a lot of rows or columns, and scrolling is unavoidable, it's good to have the label part (column or row) stay on the top layer, with the content scrolling under them. It will help the user understand the context when moving across the table.



SORTING

Sorting is a process of re-arranging the data in a row or a column based on the desired criteria (ascending, descending, etc.). When sorting is possible, it's essential always to make that fact clear and visible. The best choice is a small arrow showing the sorting direction (either up or down).

Coffee	Price ^
Espresso	6,00
Doppio	8,00

Coffee	Price v
Doppio	8,00
Espresso	6,00

In case we have more than the main two sorting options, we can use a simple dropdown with all the options to sort the specific row or column.

That intricate sorting pattern should only appear when necessary, however, as it makes the table a bit harder to use because of the added interface complexity. The more options, the longer it will take your users to understand the data and take action.

Coffee	Size	Price ^
Espresso	S	6,00
Doppio	S+	8,00
Cappuccino	M	10,00
Latte	L	12,00
Flat white	L	12,00

When a column is defining the sorting of the entire table, it's good to highlight it with either a bold label or a background color to make it stand out.


“

Clarity is more important than the number of rows you can fit on one screen.”


READABILITY

Table readability comes from a couple of factors like typography, density, color use, and separators. Keep only the essentials, remove the rest, make sure everything is aligned and have enough space to be easy to scan.

In the example below, we removed column centering, unnecessary (and attention-grabbing) borders, and simplified the typography to just one font with two weights. Adding more whitespace between the rows allows for easier scanning and increased readability.




Coffee	Size	Price
Espresso	S	6,00
Doppio	S+	8,00
Cappuccino	M	10,00
Latte	L	12,00




Coffee	Size	Price
Espresso	S	6,00
Doppio	S+	8,00
Cappuccino	M	10,00
Latte	L	12,00

ALIGNMENT

Text data should be left-aligned (to their labels as well) as we use the F-pattern to scan them. Numbers, on the other hand, should nearly always be right-aligned. It allows the user to quickly find much larger numbers in the set, as they would stand out along the left edge.



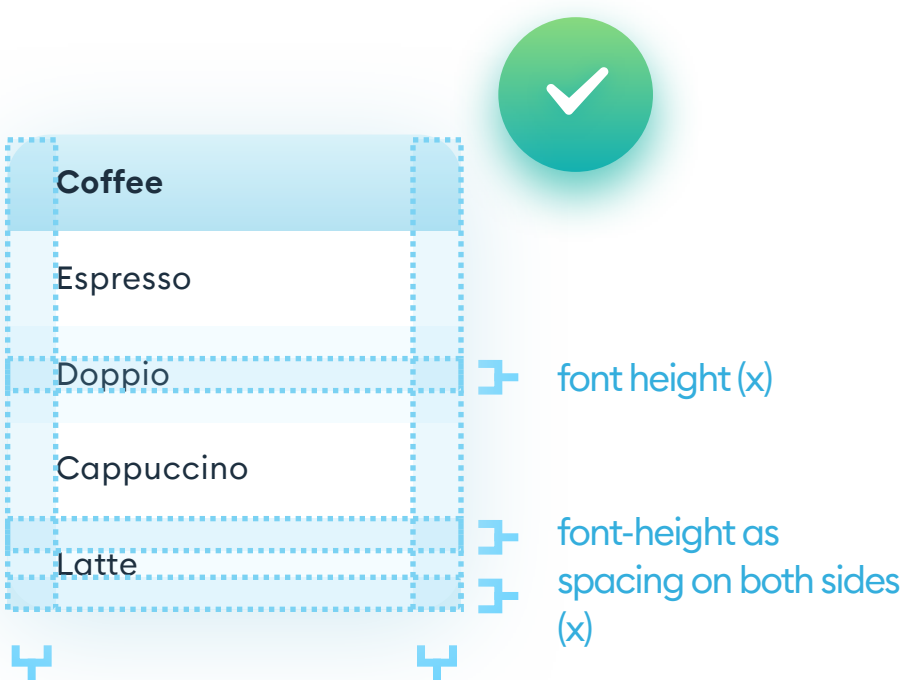
Coffee	Price ^
Espresso	6,00
Doppio	8,00



Coffee	Price ^
Espresso	6,00
Doppio	8,00

WHITESPACE

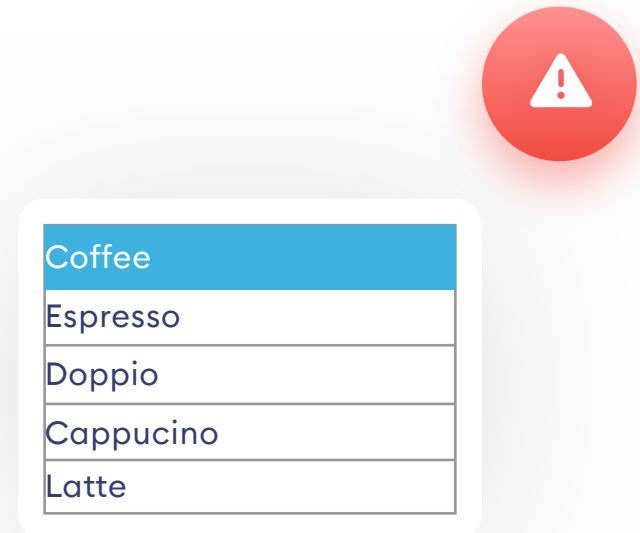
Try to treat a table cell like a button. That means it should have enough space on all sides of the data. The text inside a table cell should have enough "breathing room" to be easy to read. A lot of left-aligned tables are very dense, with data super close to the vertical dividers. That dramatically decreases readability.



A diagram showing a table with a header row 'Coffee' and four data rows: 'Espresso', 'Doppio', 'Cappuccino', and 'Latte'. The table is enclosed in a dashed blue border. A green checkmark icon is positioned above the table. To the right of the table, there are three blue arrows pointing to the right margin, with labels: 'font height (x)', 'font-height as spacing on both sides (x)', and 'font-height as spacing on both sides (x)'. Below the table, there are two blue arrows pointing to the left and right margins.

Coffee
Espresso
Doppio
Cappuccino
Latte

Minimal side margin should either come from the grid base numbers (8 or 10) or fit at least one capital letter.

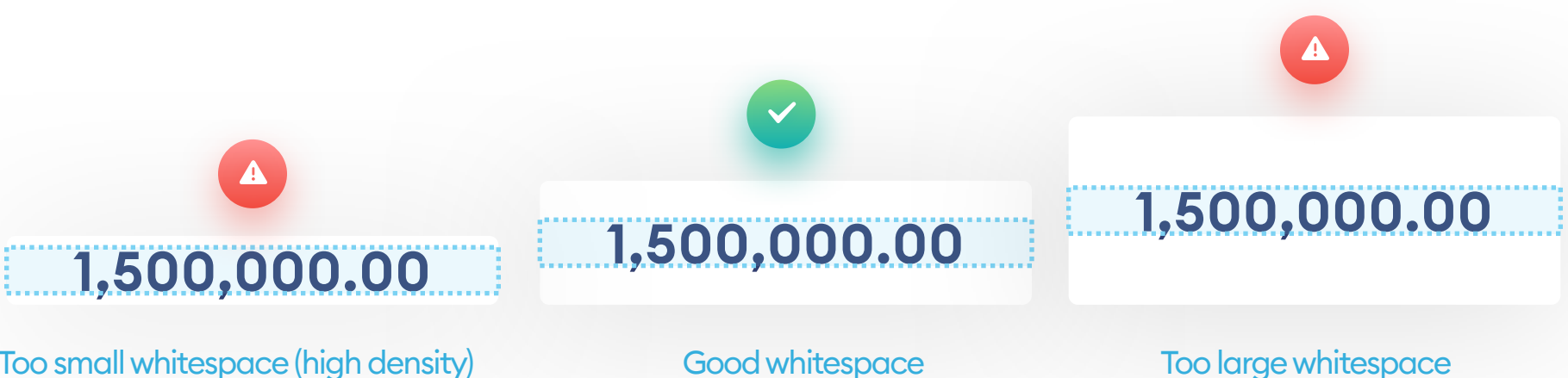


A diagram showing a table with a header row 'Coffee' and four data rows: 'Espresso', 'Doppio', 'Cappuccino', and 'Latte'. The table is enclosed in a solid blue border. A red warning icon is positioned above the table. The text is positioned very close to the left border.

Coffee
Espresso
Doppio
Cappuccino
Latte

With dense, low rows text positioned close to the left border will become hard to read. No left margin makes it a lot harder to scan as our eyes jump from text to the border and back.

Despite popular opinions, people are not afraid to scroll. Creating a dense, data-heavy table just to fit more rows on one screen is usually not the best idea. There are, of course, exceptions in products for professional data analysts, but for most people dense = hard to read. While adding whitespace to your cells, try not to overdo it. If the space is too big, it also decreases the legibility of the data.



A diagram comparing three examples of the number '1,500,000.00' displayed in a table cell. The first example has a red warning icon and is labeled 'Too small whitespace (high density)'. The second example has a green checkmark icon and is labeled 'Good whitespace'. The third example has a red warning icon and is labeled 'Too large whitespace'.

1,500,000.00
1,500,000.00
1,500,000.00

Too small whitespace (high density)

Good whitespace

Too large whitespace

SEPARATORS

We can separate rows or columns in three major ways. The most common one is, of course, a line separator. Two other types are whitespace based separation and cell backgrounds. A background that only affects every other row is called zebra-stripes.

THREE TYPES OF DATA SEPARATORS IN A TABLE

Coffee
Espresso
Doppio
Cappuccino
Latte

Whitespace

Coffee
Espresso
Doppio
Cappuccino
Latte

Separator (line)

Coffee
Espresso
Doppio
Cappuccino
Latte

Background

If our table only has a couple of rows or columns, you can easily remove most line-separators and backgrounds, and use whitespace. With more complex, larger data tables, you may want to consider separating the data with horizontal lines, or backgrounds.

Vertical dividers are rarely necessary. When doing line separation, avoid thick, high-contrast edges. Use either light grays or a lighter hue of your primary color.

Coffee
Espresso
Doppio
Cappuccino

Coffee
Espresso
Doppio
Cappuccino

Coffee
Espresso
Doppio
Cappuccino

TABLE WIDTH

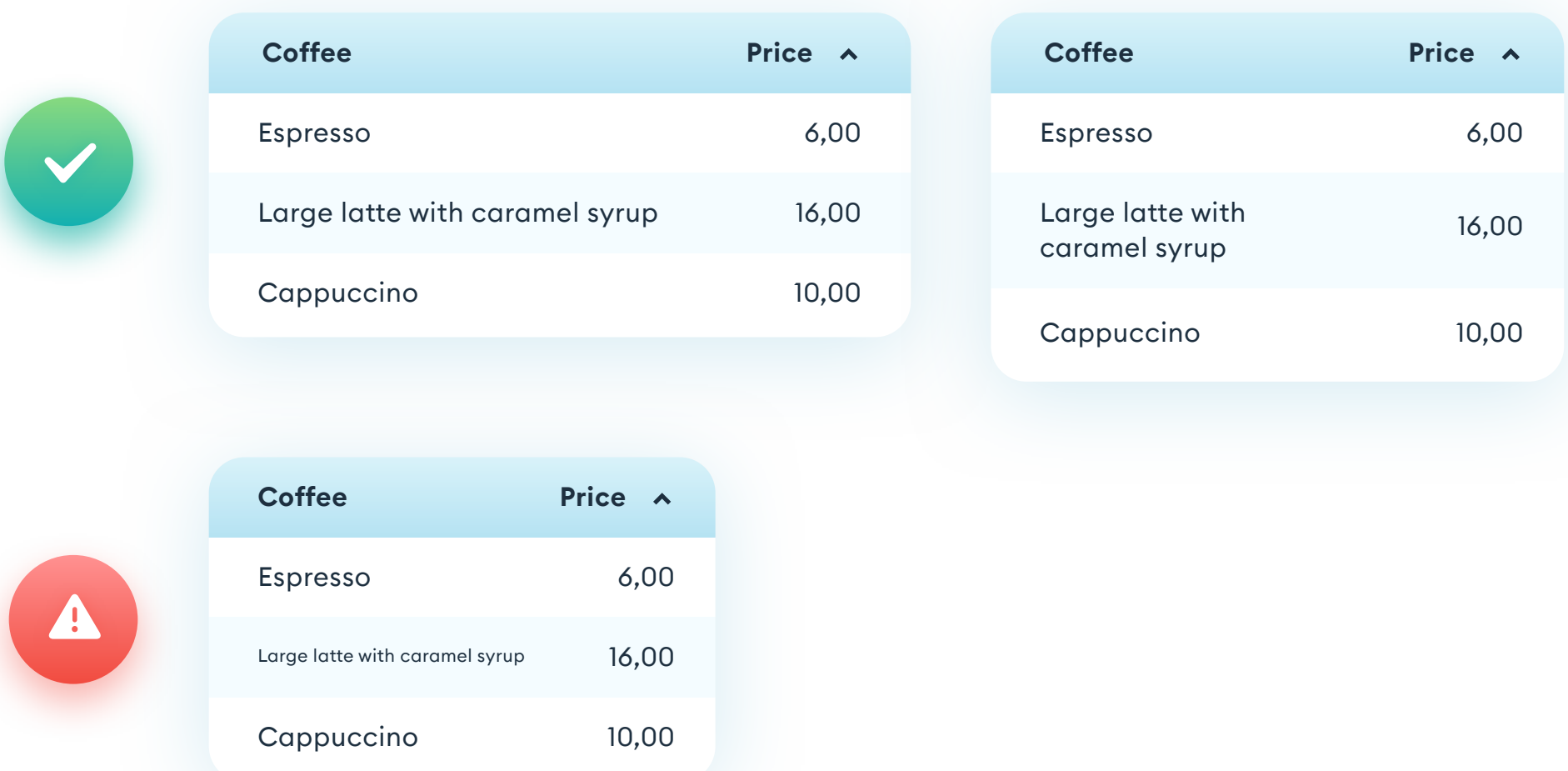
Simple, one or two-column tables should never stretch to fill the screen width. When our columns are too far apart, it dramatically decreases readability. Make the table narrower, and you'll be fine.



Coffee	Price ^
Espresso	6,00
Doppio	8,00

Coffee	Price ^
Espresso	6,00
Doppio	8,00

Designing the table width, however, we have to plan for the longest possible data entry in a row. Selectively decreasing the font-size in just one cell is very bad for legibility and should be avoided. Keep your cells consistent and straightforward throughout the entire design. With longer text it's completely OK to break it into two (or more) lines.



Coffee	Price ^
Espresso	6,00
Large latte with caramel syrup	16,00
Cappuccino	10,00



Coffee	Price ^
Espresso	6,00
Large latte with caramel syrup	16,00
Cappuccino	10,00



Coffee	Price ^
Espresso	6,00
Large latte with caramel syrup	16,00
Cappuccino	10,00

PHOTOS IN TABLE CELLS

Photos in data tables are pretty rare. However, there are times when you need to show a list of users or photo uploads, and adding a photo to the cell is necessary.





In that case, always use a thumbnail with enough whitespace. Photos should also be centered vertically against all the other cells in the row. In the example below, you can see that whitespace + vertical centering makes the table a lot more user-friendly (left).

Students	Grade
 Anna Smith	A
 Jenny Wilkes	B

Students	Grade
 Anna Smith	A
 Jenny Wilkes	B

If the photo in the table cell is supposed to expand, create a popup with the full version that can be activated by clicking or tapping on the thumbnail.

ICONS IN TABLE CELLS

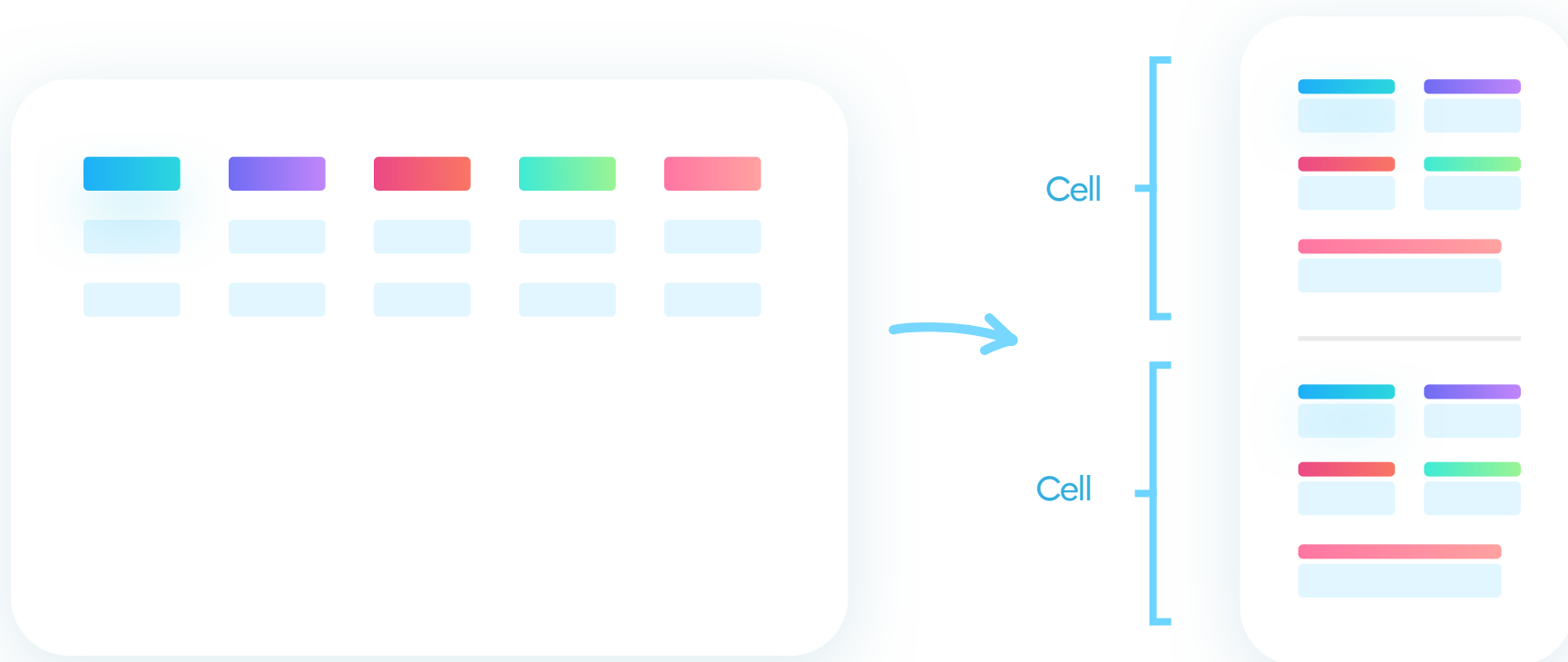
Coffee
 Espresso
 Doppio
 Cappuccino
 Latte

Avoid using icons in data tables (especially decorative ones). All graphical elements capture our attention while usually being less important than the actual data it illustrates.

TABLES ON SMALL (MOBILE) SCREENS

Wider tables work well on desktops, but can be a problem to recreate well on a mobile device with a much narrower screen. There are two main ways to make mobile tables work. One of them is a simple side-scrolling table (remember to keep the labels over the scrolling data). It can cause some accessibility issues and is harder to use, but for large data sets, it can be the lesser of two evils.

The other way is turning our entire table into a set of cells - basically a mini table for each row of our wider counterpart. Each cell should contain labels for every data entry, and the layout can become more vertical.



A wide table with two rows

The same table adjusted to a smaller screen.


This approach works best when the table doesn't have that many columns (up to five). That way of showing data is more natural on mobile devices, as it only allows for vertical scrolling.

Keep in mind that in this setting, it's a bit harder to compare data between rows, as they are split into individual cells.

COLORS

Avoid using too many colors in your tables. Shades of grey and soft pastels should be enough, with an optional darker shade used to highlight the active row. When designing a table, ALWAYS start with just shades of grey, however, as it keeps the emphasis on the data and not coloring. You can add colors in the next step when you feel they're necessary.

It will help you avoid visual overload and focus on data clarity. You can add a subtle color background to the label row, to better differentiate it from the rest of the table, but that's it.



Coffee	Size	Price
Espresso	S	6,00
Doppio	S+	8,00
Cappuccino	M	10,00
Latte	L	12,00
Flat white	L	12,00



Coffee	Size	Price
Espresso	S	6,00
Doppio	S+	8,00
Cappuccino	M	10,00
Latte	L	12,00
Flat white	L	12,00



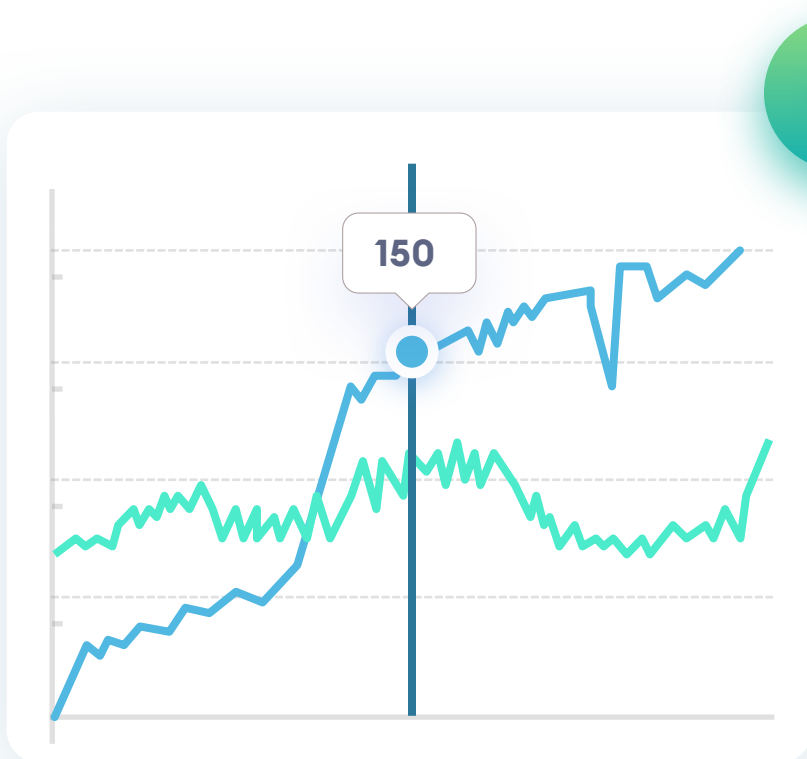
Coffee	Size	Price
Espresso	S	6,00
Doppio	S+	8,00
Cappuccino	M	10,00
Latte	L	12,00
Flat white	L	12,00

GRAPHS

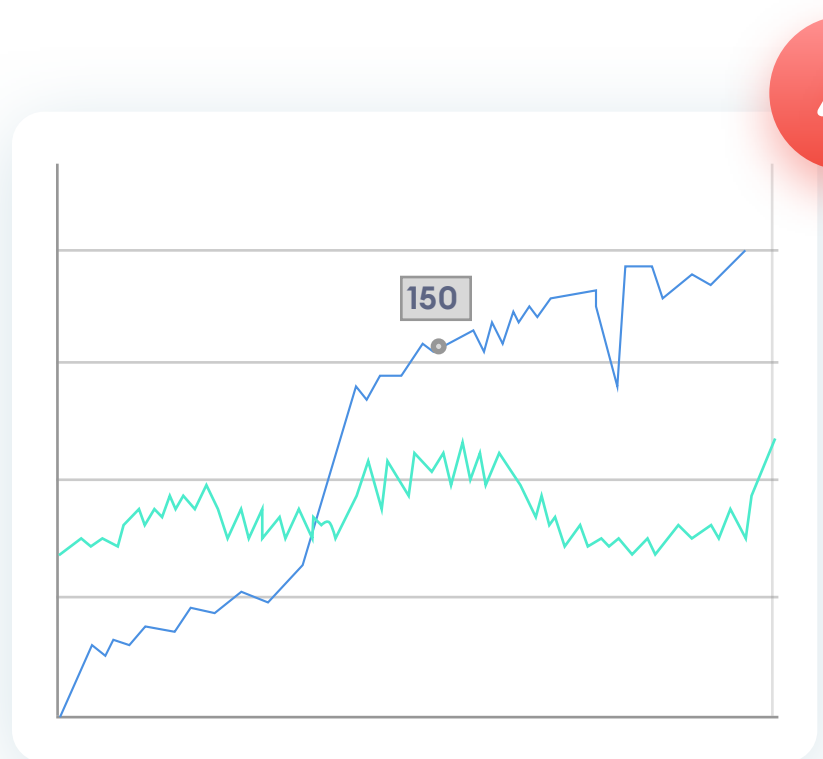
Graphs are another way to show a large amount of data visually. You can use them on both websites and mobile apps. Keep in mind that they're pretty tricky to code (or modify), so most graphs we see in products are not the best quality.

LINE GRAPH

A line graph, also known as a line chart, is a type of graph used to compare two values (often it's growth over time). It's the most popular type of chart, and most often done in a non-aesthetic way. You can easily bump up the visual quality of the line graph by using three simple rules: adjusting the line thickness, choosing the right contrast, and fixing the tooltip typography and alignment.



Clear tooltip with enough whitespace, thick, contrasting graph lines against much subtler informative lines.



Lines are too thin, which makes everything blend together. Tooltip is hard to read due to poor contrast and whitespace.

TOOLTIP

A tooltip is one of the essential parts of any graph. It's that small popover window showing a precise value that appears once we hover (or tap) on the graph line.

GRAPHS

Just as with data tables, when designing a chart, try to remove all the unnecessary clutter and only keep the essentials. You can do it by eliminating the guides (help lines) or making them more subtle. The graph itself should be front and center at all times, with the highest contrast and visibility.

FILLED GRAPHS

Pie-charts or bar-graphs are another popular type of data visualization. Keep in mind, that differentiating them by color alone can prove difficult to understand by people with visual impairments.

Unless our graph is interactive (clicking or hovering will visibly show the selected part), consider using an additional pattern over the color just to be safe.



You can test it by making all your colors greyscale. If you can still see the difference between the objects, it means they will be more accessible for your users. Another good practice is starting to design all types of data using just a single color, and differentiating them by a pattern. You can always choose complementary colors at the end.

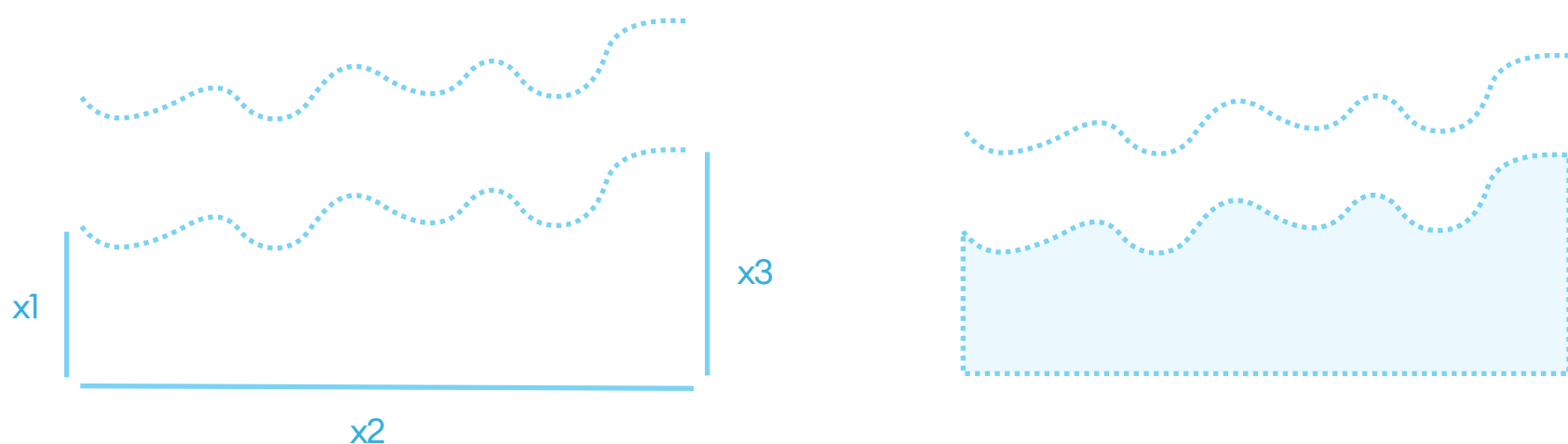
"UNREAL" GRAPHS

Many portfolio shots make their line charts super-smooth for a higher visual impact and a more beautiful result. It's good to know how to do them, to understand better the compromise between real data, and one that looks well in presentations.

It bends the honesty rule (see rules of design) a bit, so try not to overdo it. Most real data doesn't flow in a perfect, symmetrical sine-wave, even if it looks fantastic in your Dribbble shot.



To create a chart like this, first, create a curve with a couple of up and down waves. Then duplicate it and add the three closing sides, making it a full shape. (x1,x2,x3)



Add a subtle gradient to the resulting shape, starting from the same color as your line, and going down to the same color, but with full transparency. You'll end up with a fashionable and good looking (albeit not "real") effect.



17.

Forms

Forms

INTRODUCTION

A form is one of the most common ways a user can interact with a digital product. It helps with making a purchase, creating a profile, or signing up for a newsletter. Done right can boost a business. Done wrong can break it. In simple terms, a form is a set of labels and fields in which we manually enter information, with a button at the end, to save everything in a database.

name... Select option ▼ Your e-mail *****

CONVERSION

Conversion is a percentage of completed form entries. Every user that "drops off" before clicking that final "save" button is a decrease in our conversion rate.

ATF - ALWAYS TEST (YOUR) FORMS!

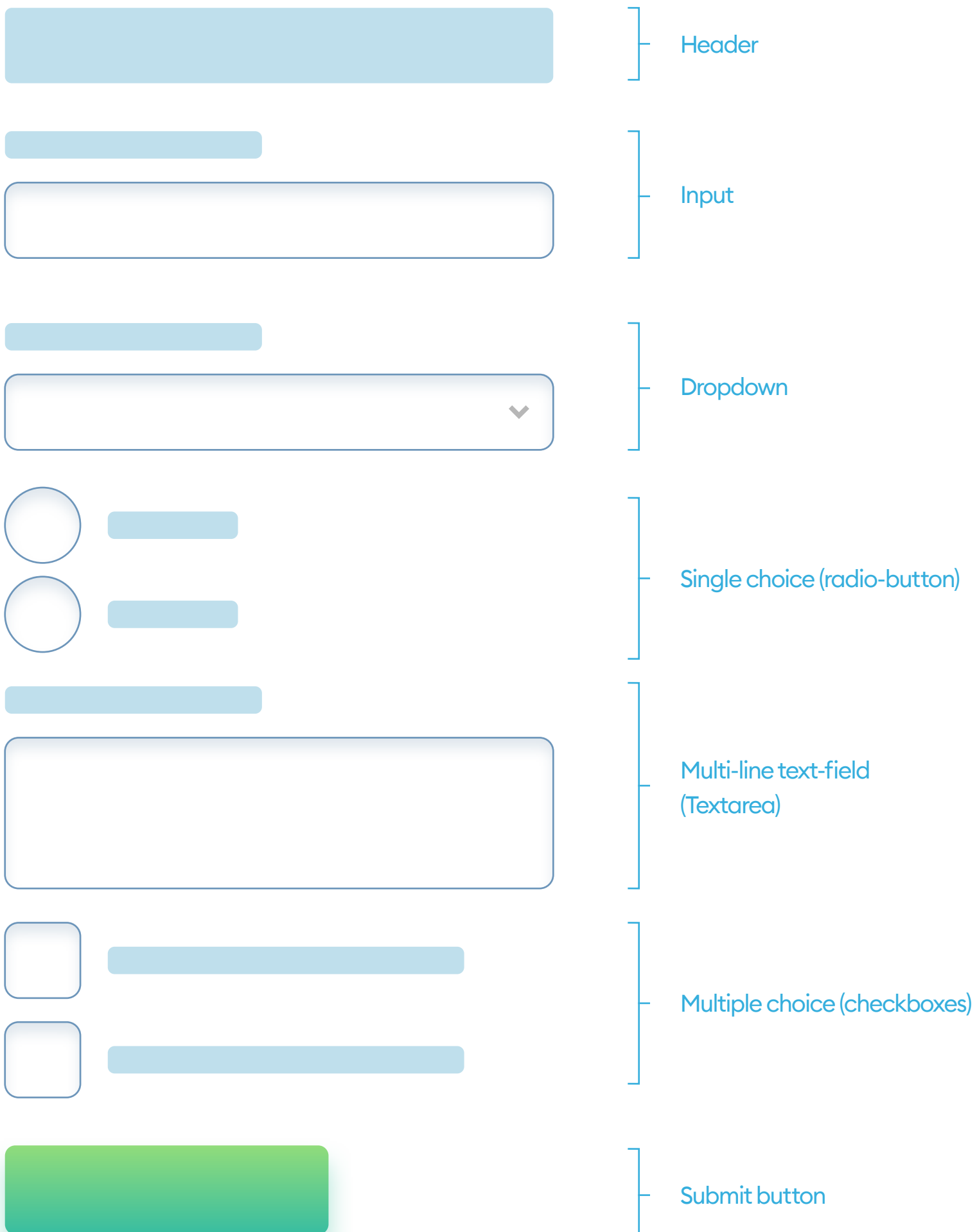
We can influence conversion with good form design. However, all users and target markets are different, and your form needs to be tested extensively with real users. In some cases, to make the form work better for a specific set of end-users, you need to bend or outright break some "good design" rules.

“

**Always test your
forms on real
users. ”**

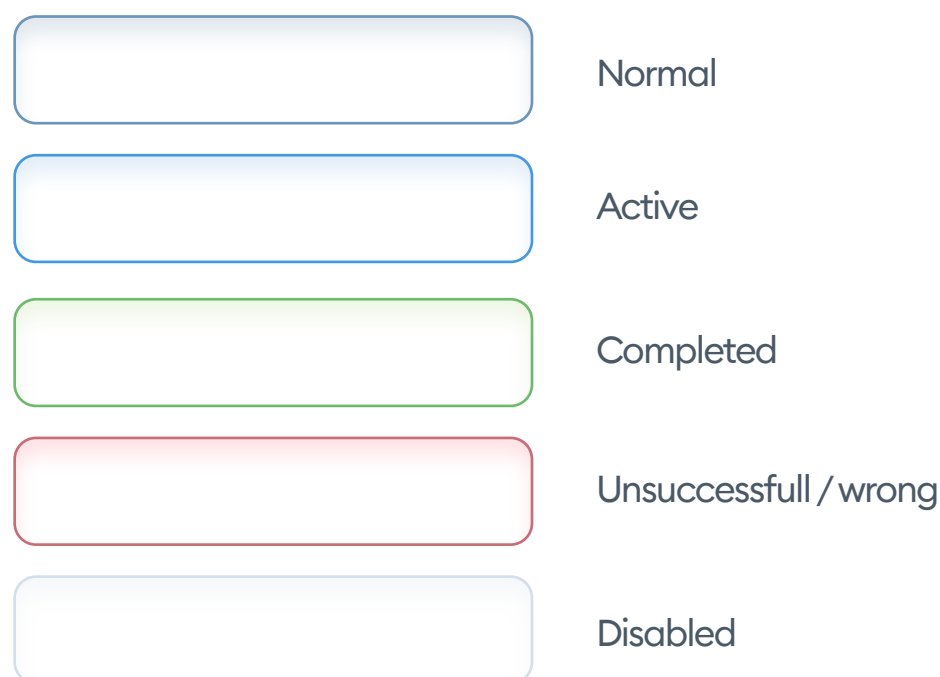
FORM ELEMENTS

The basic set of elements in a form includes a text field, a dropdown, and a button. We also have multiple-choice fields (checkboxes) and single choice ones (radio-buttons), text-areas for longer text entry. Other types (like sliders) are less common.



FIELD STATES

A form field can have one of five possible states. Try to design every form element (field, checkbox, radio, etc.) in all of those states. The easiest way is starting with the generic text-field (input) and then move the shadow, border, font, and color styles over to the other elements.



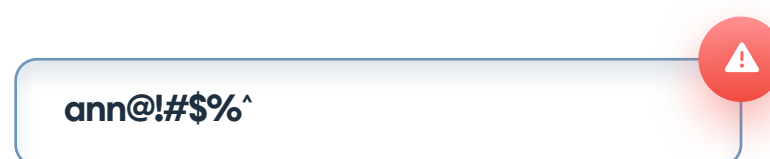
You can set field states by either a border-color, by using icons inside (like a checkmark for successfully validated fields), or different text-weights.

Pay especially great attention to designing the negative, invalid states of your fields. A short description of what went wrong should go directly under each field for better context and ease of use. You can color that text red for increased urgency.

Avoid having all validations displayed under the entire form, or inside popups. Keep your descriptions brief and to the point. It also makes sense to show the proper way to fill the field as an example.



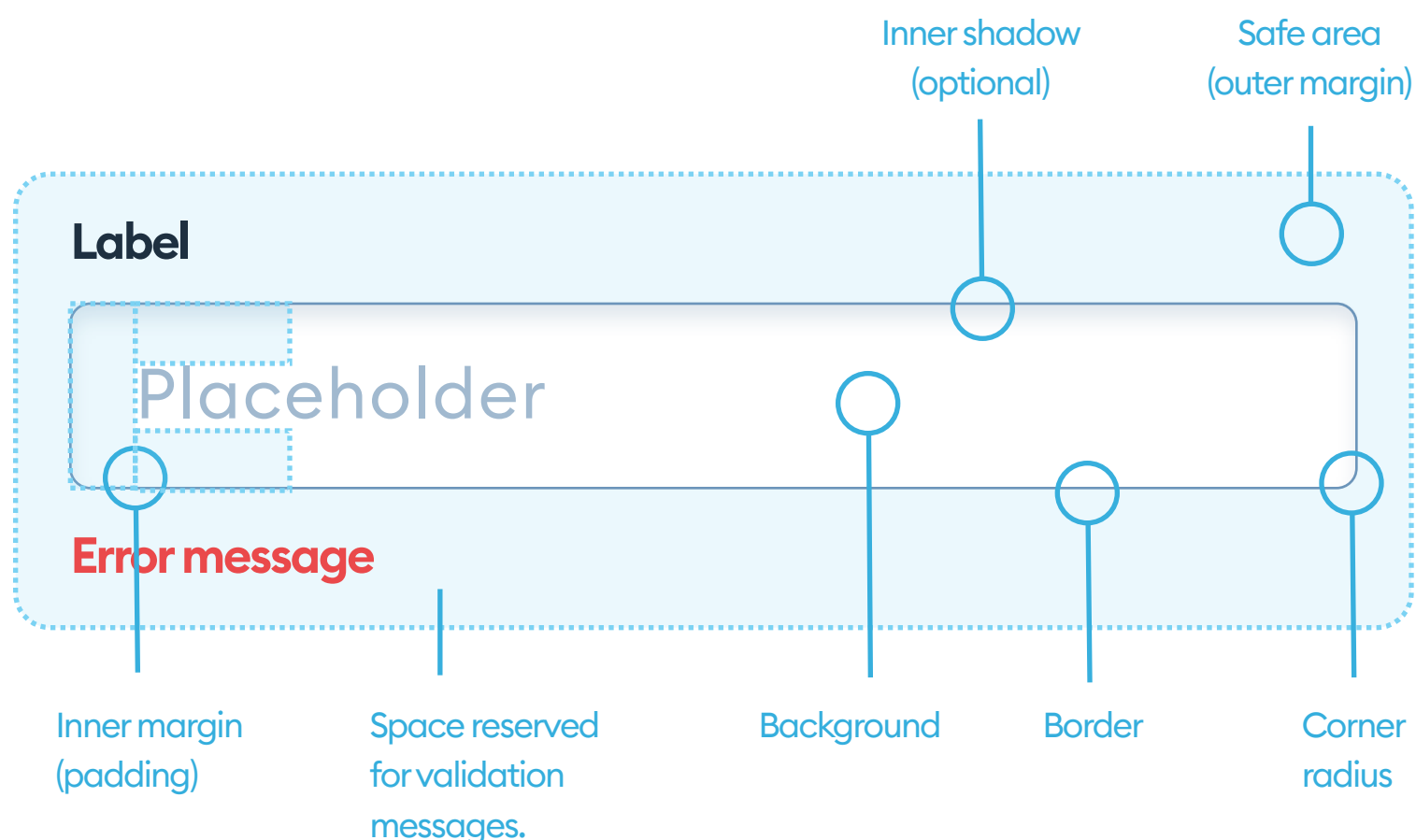
The entered text is not a valid e-mail address



TEXT-FIELD / INPUT

When designing a form, always start with the humble text-field. It's the most common form field, and you can use it as a visual guide for all the other fields. When designing for mobile devices, keep your placeholder text larger than 16 points; otherwise, the device can sometimes zoom the form in, thus losing the context.

Use Title-Case for your labels for faster processing. Labels in UPPERCASE take slightly longer to process.



TEXT-FIELD TYPES

There are two main text-field types: The standard one, and one that hides the text, which is ideal for password entry. As you type, it only shows the current letter and hides the rest under asterisks or dots. If you're designing a password field, remember to add a little eye-icon at the end so that the user can preview the entered password.

E-mail

Standard text-field

Password

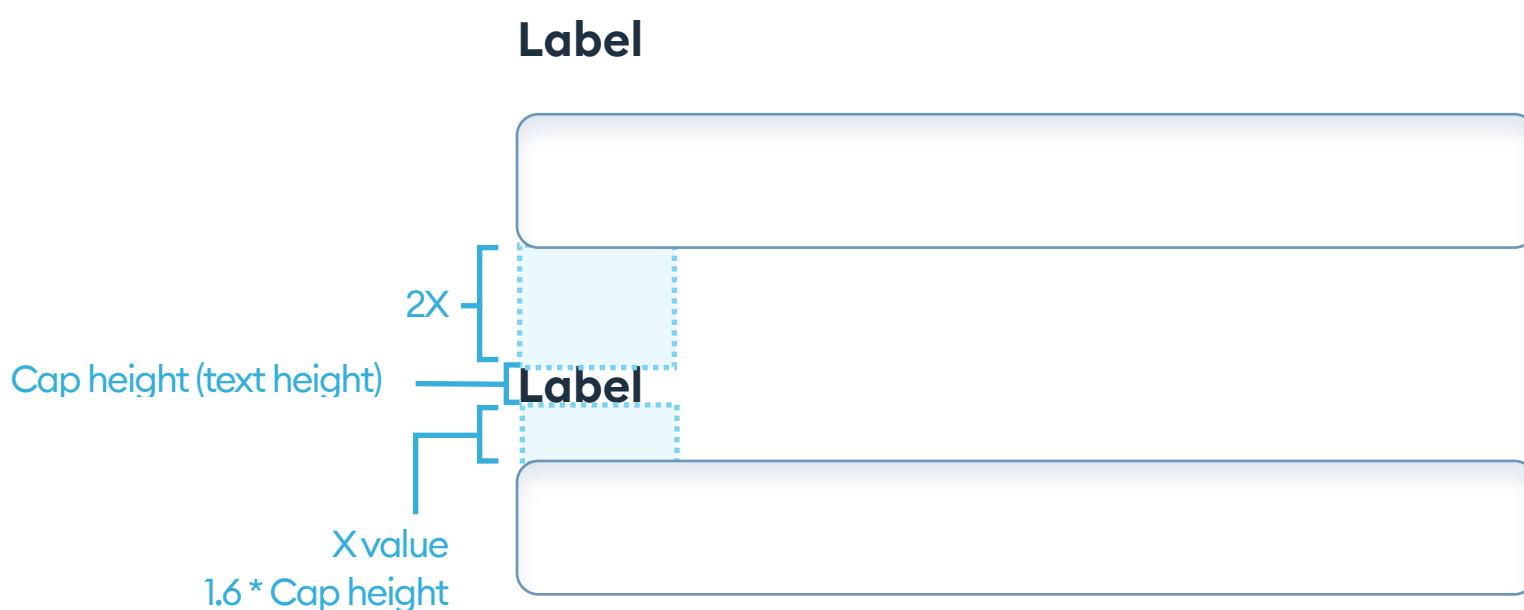
Password field - hidden

Password

Password field - shown

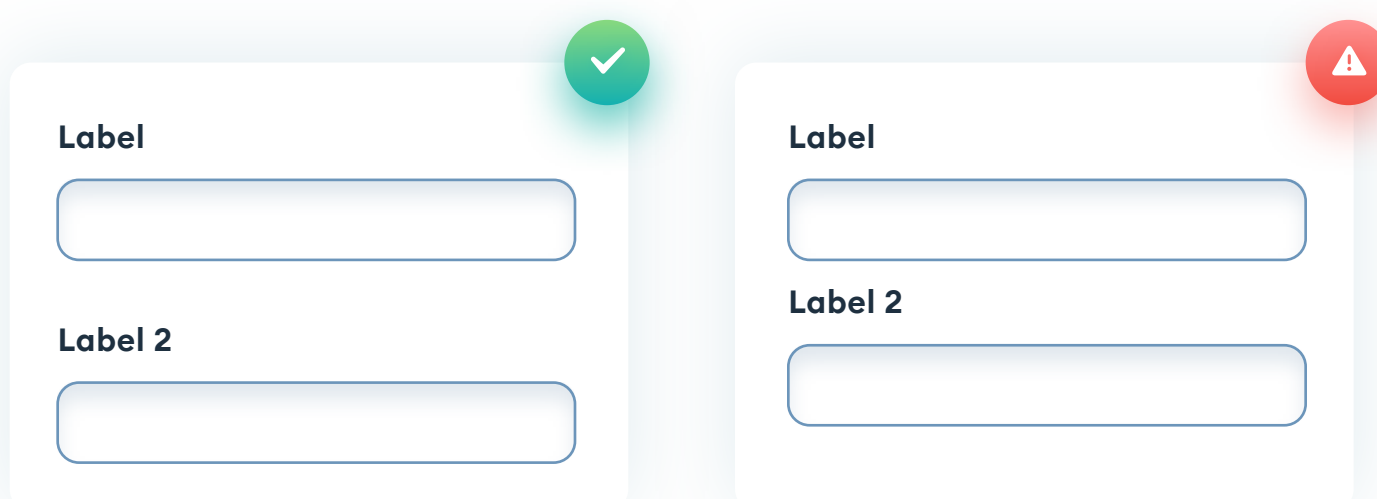
LABEL

A Label is a small, explanatory text accompanying the text-field. The main goal for it is to provide a semantic connection to the field and improve understanding. In most cases, it's simply the description of what we require from the user (for example, "enter your email or username" in login forms). In HTML, the labels are also processed by screen readers for improved accessibility.



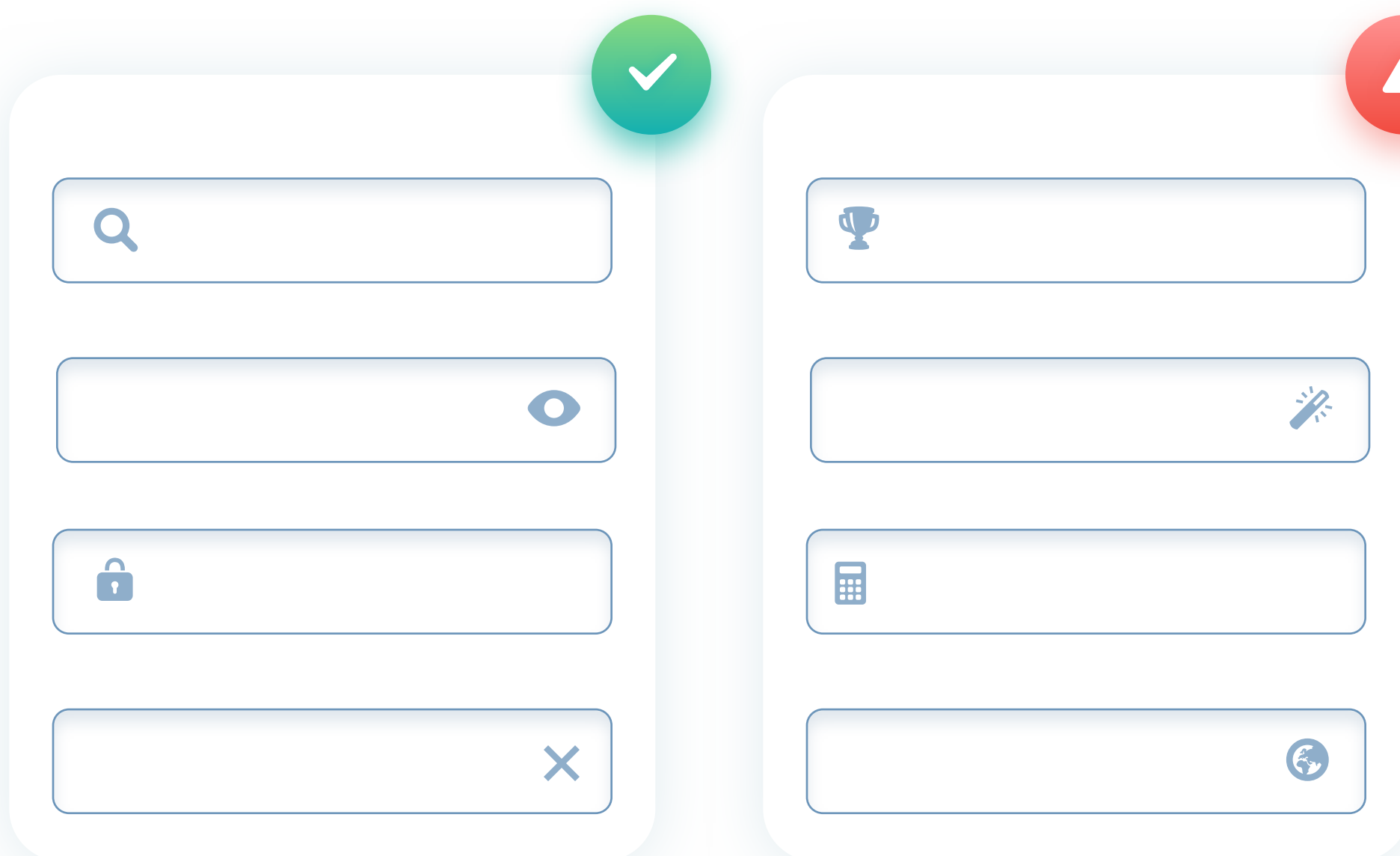
If your form has a lot of fields, remember to space them out far enough, so they don't get confused with each other. Proper spacing is using the cap-height (height of the capital letter) or 1.6 times that value.

The margin from one text-field to the label of the next one should be at least twice that height, so it's clear when one field ends, and the other begins.



USING ICONS AS LABELS

In some cases, an icon can also work as a label - for example, a magnifying glass is often used as the only indicator of a search field. But this is an exception that confirms the rule of not using icons alone as the labels. Most other icons should have a description next to them, to make sure all of your users understand what they do.

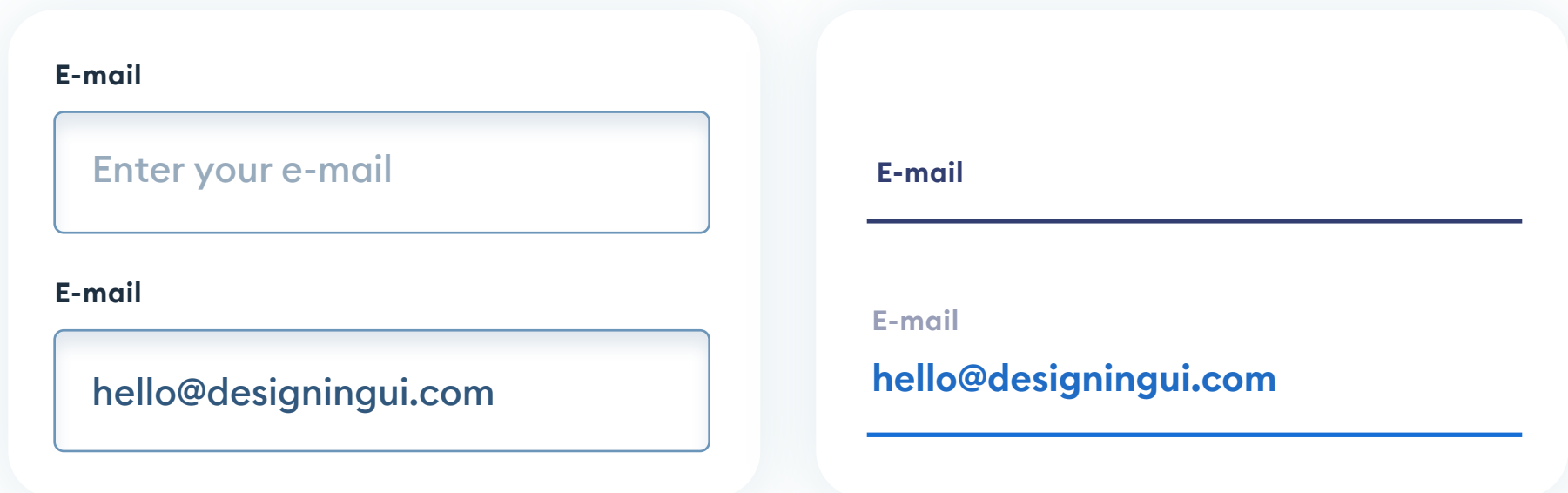


Some additional icons work great with a general label, but don't have to be described themselves. These include an eye icon (to show a hidden password), a lock (that indicates a password field), or an X used to close or remove the text-field content.

Using more abstract icons (like in the example on the right) is not recommended as they're not clear enough. In general, icon use should be minimal inside the form fields.

TWO MAIN STYLES

There are two main types of inputs you can use. The standard rectangles, according to most studies, are faster processed and better understood by users, than the Material Design horizontal lines.



Good text-fields should have a slightly lighter shade used for placeholders, with an easy to read, contrasting color for the actual data entry. In Material Design inputs, the label moves up as you start typing, and the line below changes color to indicate it's active.

CONSISTENCY

If you prefer Material style inputs, that's OK. The difference between those two types is not that big in terms of understanding, especially since Material Design has gained so much popularity. But what you should always do is to make sure you only use ONE TYPE of inputs in your entire product. Don't ever mix Material labels with the classic, rectangular shapes!

Remember to keep the label font smaller than the text inside the field. This contrast and size change will help with setting the proper hierarchy.

INNER SHADOW

Inner shadows were trendy during the Skeuomorphic era, but are now very rare in UI's. Forms are one of the only exceptions, where it can make sense to consider an inner shadow.

Many studies have shown that adding a subtle inner-shadow to a text-field slightly increases conversion. The change is not dramatic, but it's worth mentioning.



DROP SHADOW

Adding an outer shadow to a text-field, while keeping an outline is a purely aesthetic choice and doesn't affect conversion in a meaningful way.

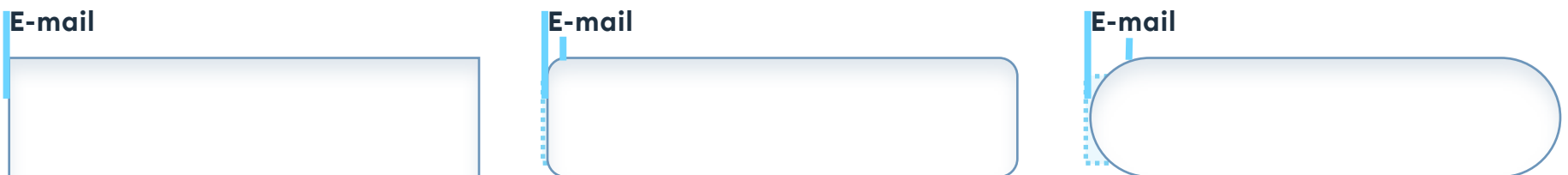


Many "portfolio shots" tend to remove the outline and leave the shadow alone. While it can look good visually, there's a significant chance that the text field will not be understood.

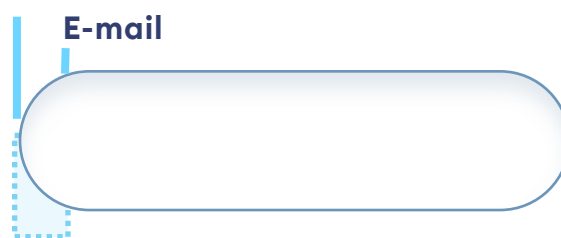
Try always to use a clear, visible outline in your form elements.

CORNER RADIUS

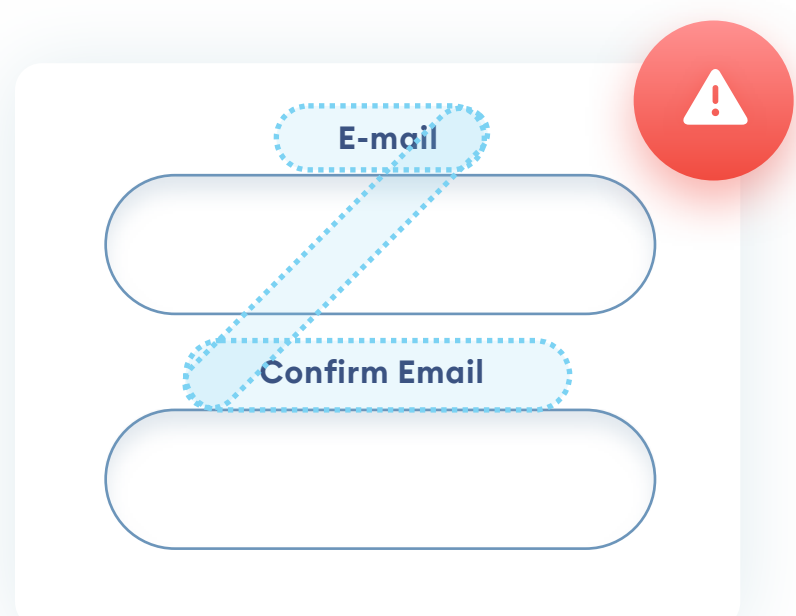
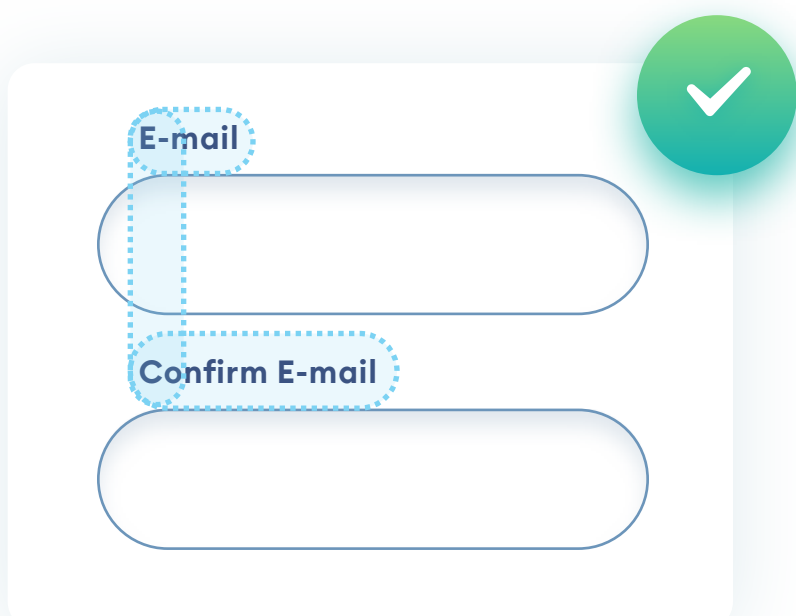
Just as with button designs, the corner radius can profoundly affect the readability of our labels. Plan ahead! If you're using a very rounded corner field (pill-shape), the text is visually floating away from the edge. This effect is non-existent with sharp corners of a very small roundness.



You can try to solve this problem by moving the text to where the rounding stops and becomes a straight line. It's not a perfect solution, and you should consider it only for short (1 or 2 fields) forms. With longer forms, it slows down our ability to quickly scan the text.



Avoid centering labels above your form fields, as it turns the natural F-pattern into a Z-pattern that's dramatically slower. The user is expecting to follow a smooth path through the fields, and with centered labels of various lengths, there's no pattern to follow, as each label starts at a different spot.

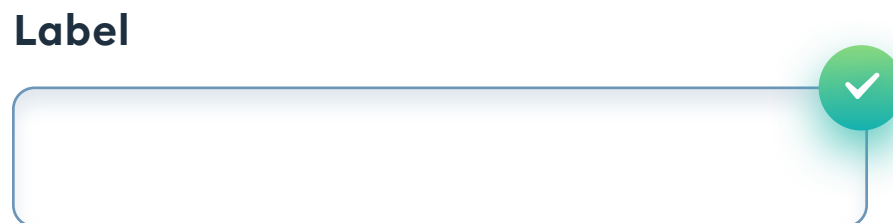


“

**Left-aligned labels
above text-fields
are the most
user-friendly
option.”**

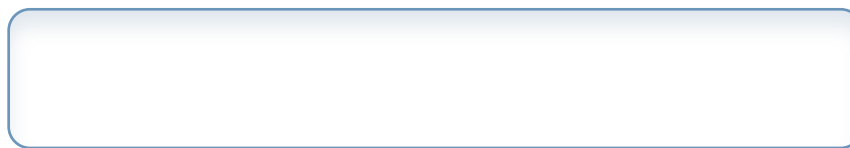
LABELS AND TEXT-FIELD ALIGNMENT

A left aligned text-field with a left-aligned label just above is always the best choice for both desktop and mobile applications. It's the easiest to scan, understand, and fill, as our eyes naturally go along the left edge.



On wider displays, you can try labels on the left side of your text fields. One way is to place the label left-aligned at a specific distance from the field.

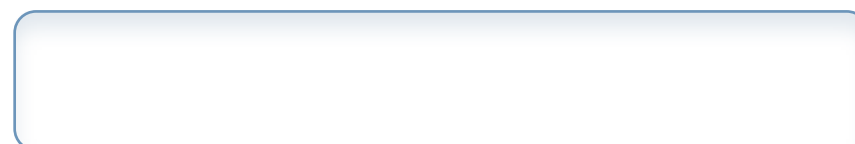
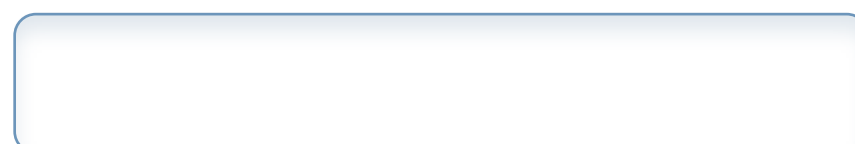
Label

A diagram showing a label "Label" positioned to the left of a rounded rectangular text field. The label is left-aligned with the left edge of the text field.

A slightly better approach is right, aligning the labels with a set margin to the input box. The main downside of this approach is that it's a lot harder to read a jagged left-edge, and with various label widths, that's the most likely outcome. Only use this if labels above the fields are, for some reason, impossible.

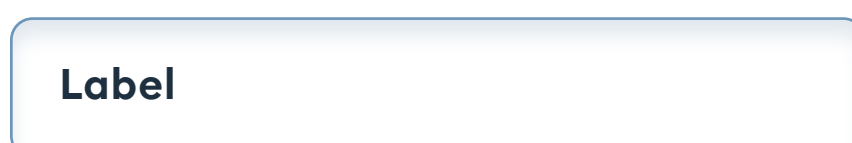
A Longer label

Label

A diagram showing a label "A Longer label" positioned above a rounded rectangular text field. The label is right-aligned with the right edge of the text field.A diagram showing a label "Label" positioned above a rounded rectangular text field. The label is right-aligned with the right edge of the text field.

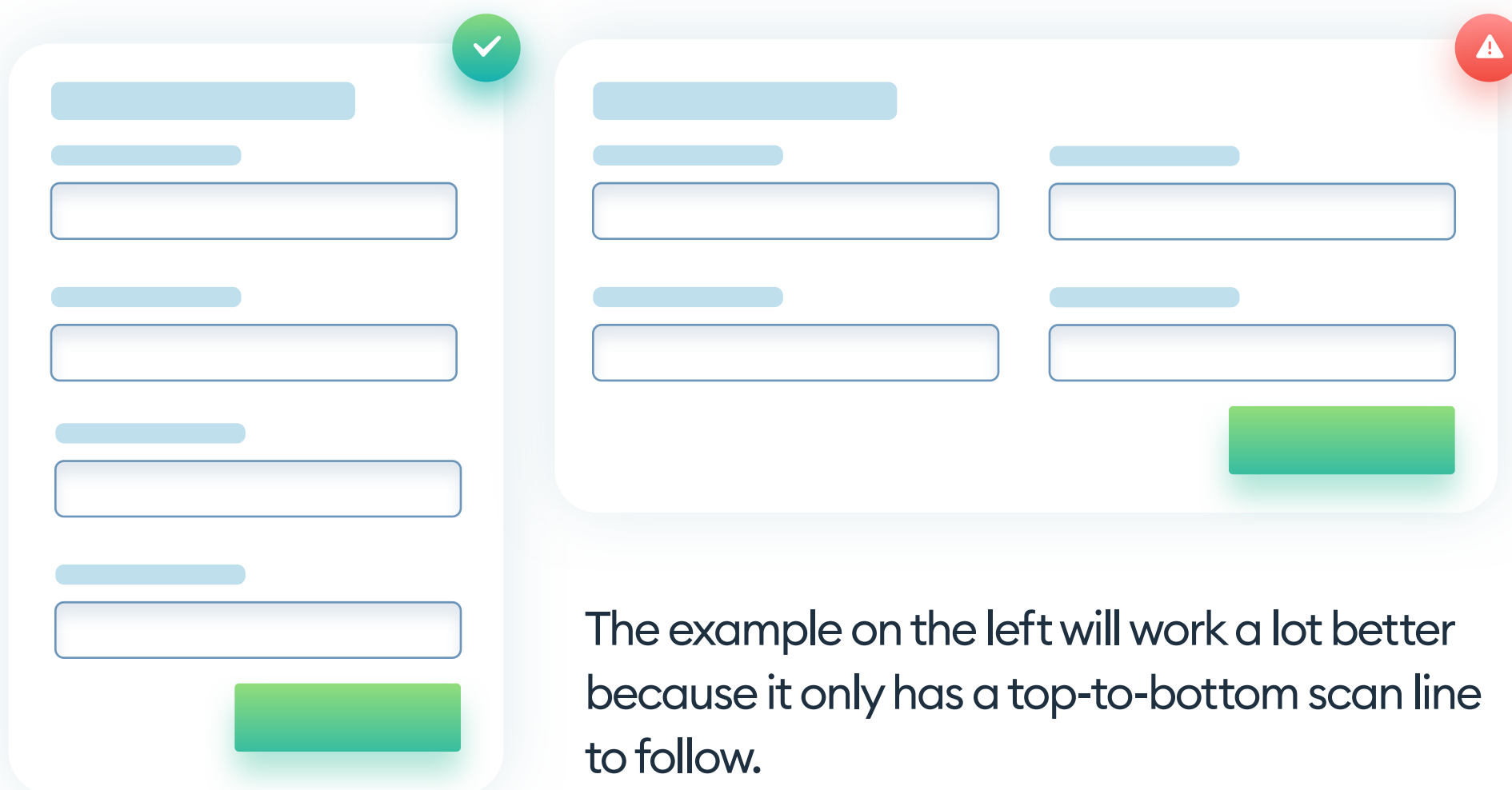
In rare cases, a placeholder text serving as a label can work (search bar), but avoid using it extensively. When entering the field to type, the label disappears, and the user can lose context. This approach can be a big problem with longer forms.

Label

A diagram showing a rounded rectangular text field with the word "Label" inside it, serving as a placeholder.

SINGLE VS MULTI COLUMN FORMS

Single column forms convert a lot better than multi-column ones because you only have a single path to follow with your eyes, instead of jumping between blocks.



Keep forms rather narrow. The best range is somewhere around 320 and 500p wide (mobile and desktop). The wider the fields, the longer it takes to process them - keep it simple!

Longer (6 fields and up) forms are better divided into separate pages, with clear information on which step you currently are.

Don't mind the empty spaces on the sides of your form. Don't fill it with placeholder images or decoration. Leave it empty! It's great for giving the form maximum attention and the best focus.

“

**Single column forms
convert much better
and are easier
to use.”**

FIELD WIDTH AND CONTEXT


If you can anticipate the typical length of the user-entered text, try to design the fields to match that. Concise information like age or area / zip-codes should use shorter fields to indicate the amount of data they require.

That way, the length of the field itself is helping the user enter the right information.

The image shows two side-by-side form designs for an email and area code input. The left design is marked with a green checkmark, indicating it is the correct approach. It features a wide 'E-mail' field with the placeholder text 'enter your email' and a shorter 'Area code (5 digits XX-YYY)' field with a placeholder '--- - ---'. The right design is marked with a red warning triangle, indicating it is incorrect. It features a wide 'E-mail' field with the placeholder text 'enter your email' and a wide 'Area code' field with a tooltip icon (an 'i' in a circle) next to the label.

In some cases (like the Area code), we can suggest the right format using a placeholder, or a short example like "XX-YYY." The user doesn't even need to add the hyphen, as it can directly jump to the third number after entering the first two.

Try to make your fields self-explanatory. If you need any additional description, try adding it to the label.

If it's longer, you can add it under a tooltip icon , but that approach is not the most optimal one, as the user can miss it and never activate the tooltip.

DROPDOWN

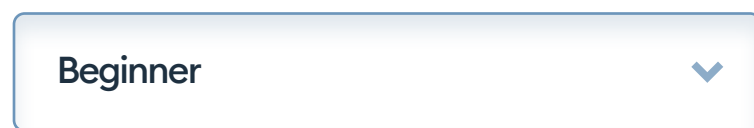
Dropdowns are another very popular form element. They allow for a single choice from a longer list, that expands on click. The primary indicator that a field is a dropdown is usually a down-arrow or chevron on the right side.

Skill level

A standard, generic system dropdown menu. It consists of a rectangular box with a light gray border. Inside the box, the text "Beginner" is displayed in a dark gray font. On the right side of the box, there is a small, dark gray square containing a white downward-pointing chevron.

Standard, generic system dropdown.

Skill level

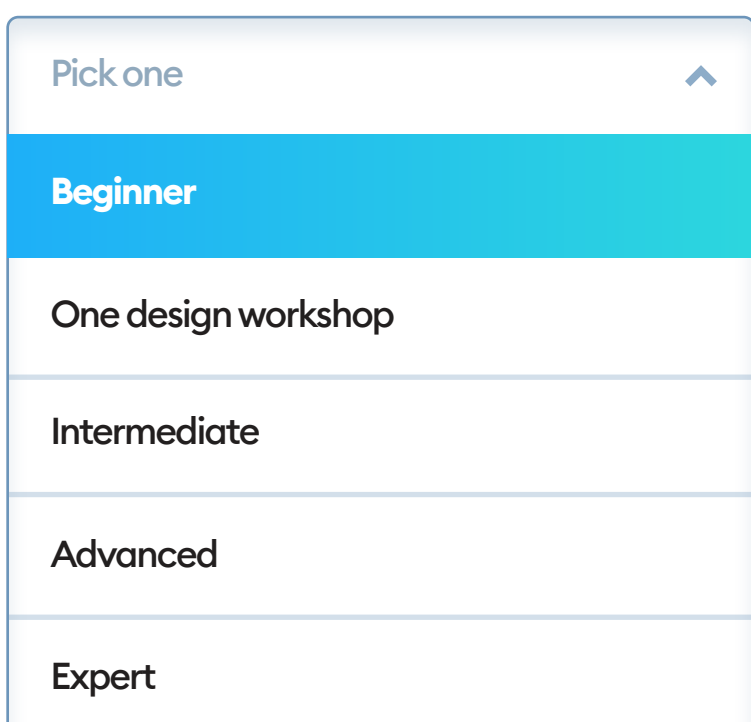
A custom dropdown menu with a light blue border. The text "Beginner" is displayed in a dark blue font. On the right side, there is a small, dark blue square containing a white downward-pointing chevron.

Custom dropdown, requires additional coding.

It's worth mentioning that an implemented dropdown will have an enforced system setting for how it looks. It means it will look differently on a Mac laptop, a Windows-based machine, or an Android Tablet. There are two ways of keeping consistency - either coding all our forms using only the platform native components or designing and implementing our very own style for all of them.

Remember, that styling all fields on your own also requires some additional development work, but the result is often well worth it.

Skill level

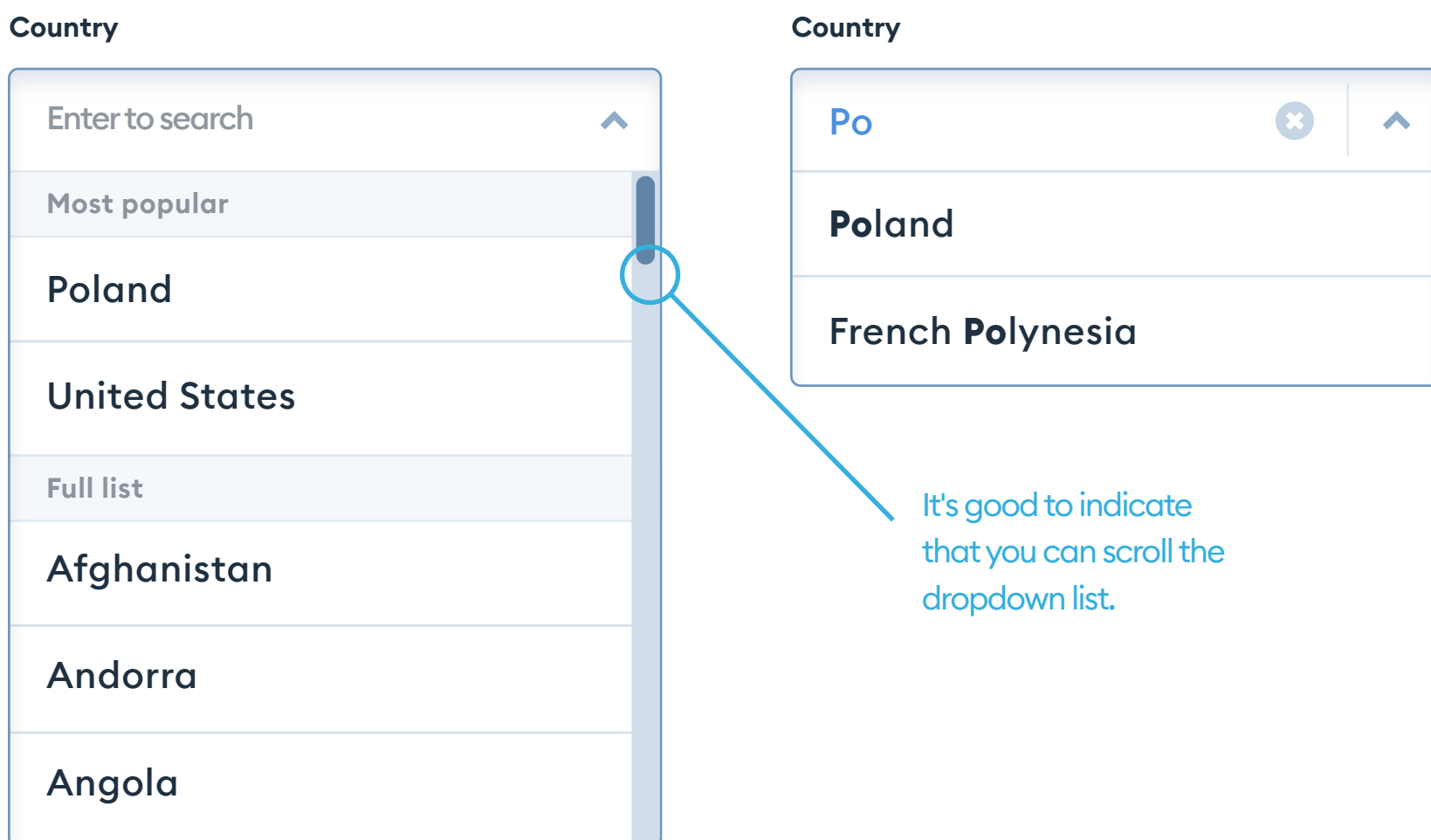
An expanded dropdown menu. The top part is a header with the text "Pick one" and a small upward-pointing chevron. Below the header is a list of options: "Beginner", "One design workshop", "Intermediate", "Advanced", and "Expert". The "Beginner" option is highlighted with a bright cyan background and white text. The other options have a white background and dark gray text.

After our dropdown box expands, we need to show the selected option clearly. Try using a contrasting background as the selection, as just changing the text color can often not be enough. When designing an "open" dropdown, remember to rotate your arrow 180 degrees, which indicated that clicking again will fold it.

LONG DROPDOWN LISTS

Very long dropdown lists are particularly tricky to use, as they require a lot of scrolling to find and pick the right choice. You can help your users by adding a search/filter box at the top of your dropdown and highlighting the most popular options at the top of your list.

For example, if most of your users come from the United States, you can place it first on the list, with all the other countries sorted alphabetically. That way, most of your users won't have to scroll all the way to the U.



SHORT DROPDOWN LISTS

If your dropdown has five options or less, consider changing it into radio buttons (or standard buttons that act as radio-buttons like in the example below).



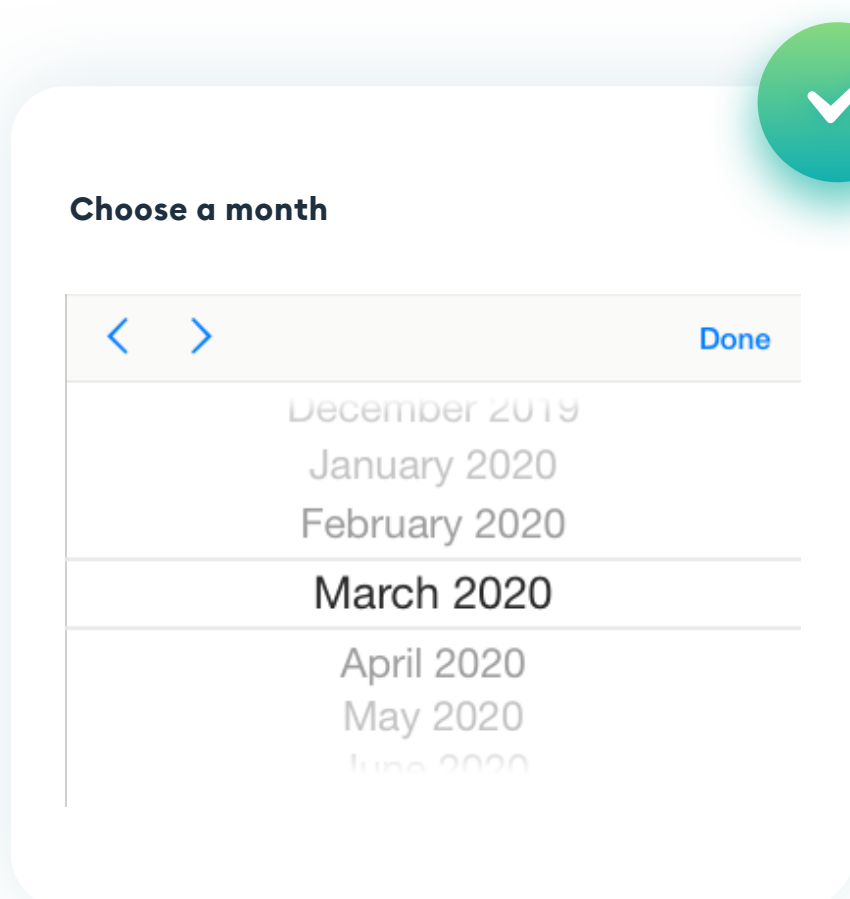
“

**Very short (3-5 item)
dropdowns will work
a lot better as a set of
radio buttons ”**

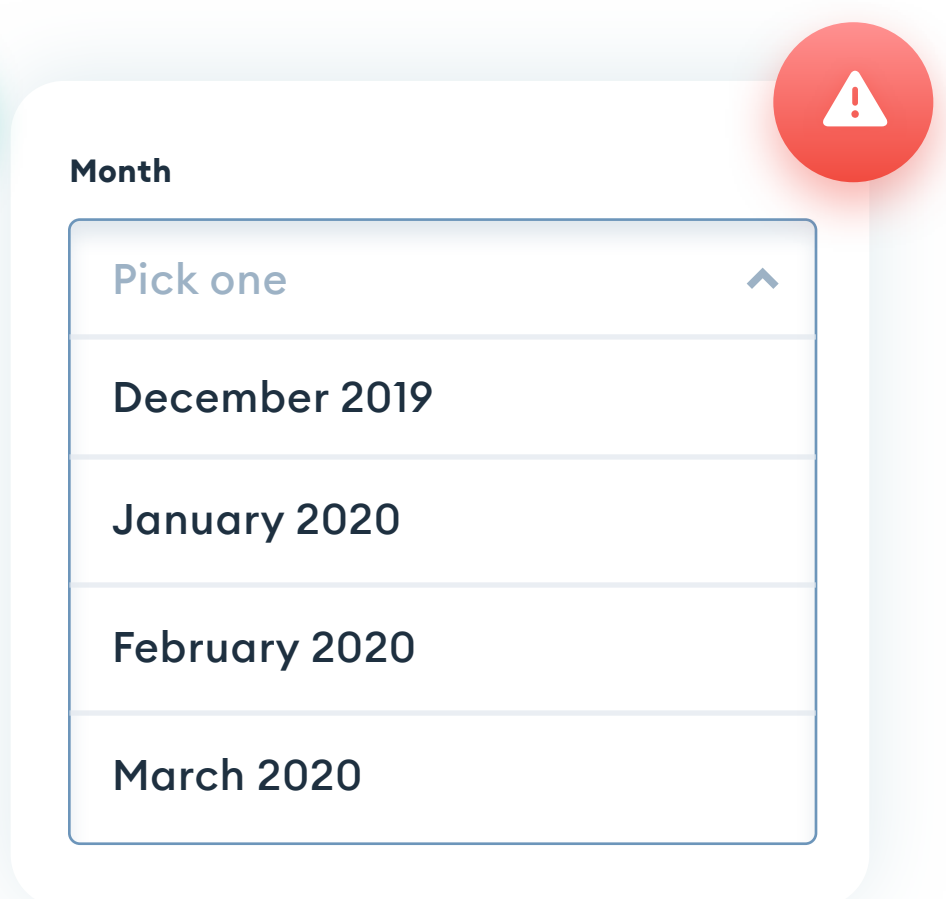
DROPDOWNS ON PHONES

While we can go wild with dropdown designs for desktops, keep in mind that on mobile devices, it's best to use the native picker instead. It's usually some kind of a spinner-wheel that comes out from the bottom of the screen.

It's a more natural and a lot faster way to pick an option from a long list on a touch-enabled device.



Classic iOS picker should always be used on phones instead of a web-imitation dropdown.



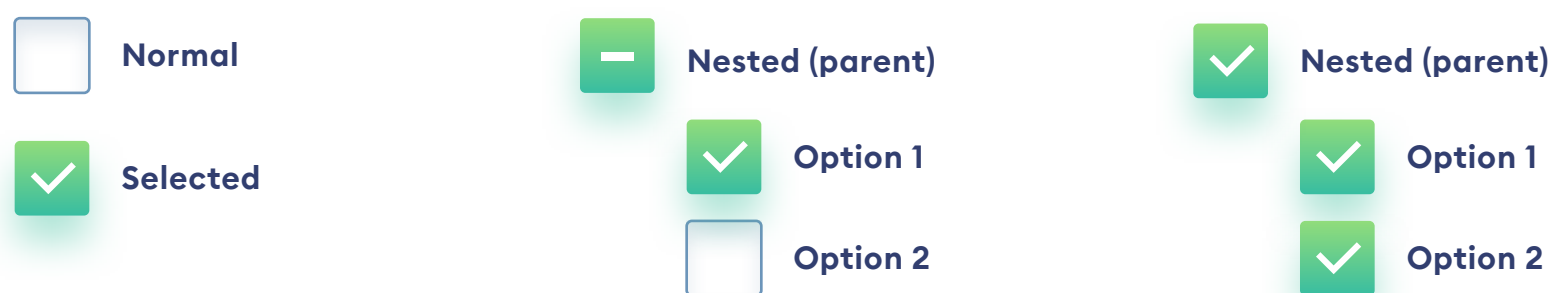
Enforcing a desktop style web dropdown on a mobile device is cumbersome and should be avoided.

Remember that both iOS and Android display the pickers differently.

Try not to modify it, however, as people are already used to this pattern and understand it well. Having a custom version can create unnecessary confusion.

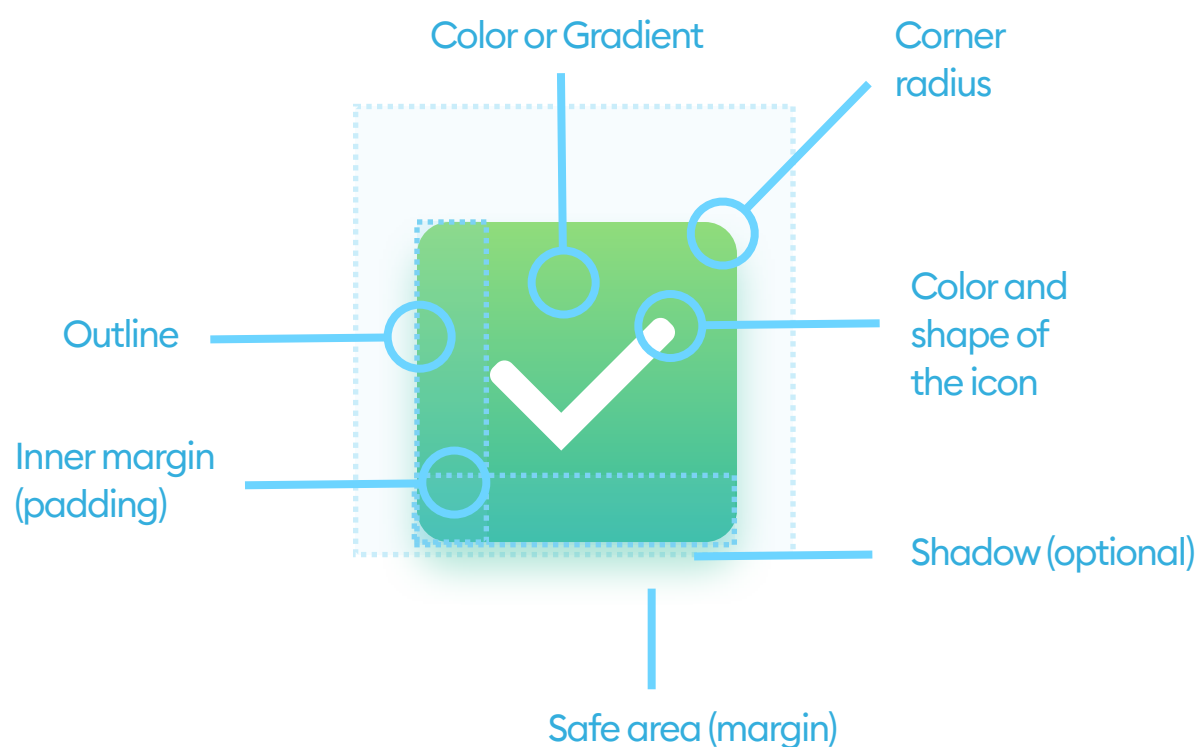
CHECKBOXES AND SWITCHES

Checkbox, or "tick," has three primary states: normal, selected, and nested, which allows for enabling some, or all checkboxes nested under one name. In general, checkboxes are a multiple-choice type field, so activating one, should not affect any other checkboxes on the screen.



DESIGNING A CHECKBOX

Showing the checkbox in an enabled state means two things: the background needs to be a color fill (usually the accent color), and there should be a checkmark (V) inside. Just as with icons on buttons, you need to have enough space inside the square shape for the icon to look good. Keep the check icon consistent with all the other icons in your design (stroke-width, color, shape, roundness, etc.)



CONSISTENCY

Just as with all the other form elements, if we want the checkboxes to be consistent with the rest of our design, we need to apply a custom style to them. If we don't, they will look different on every OS, as the browser will display a generic system version instead. The most common is a simple, grey gradient that may not fit our product.

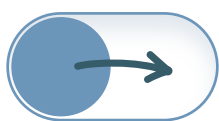


A generic, system checkbox.

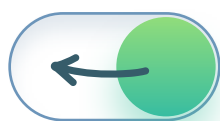


A styled, custom checkbox can have a background fill (or a gradient) and a custom icon.

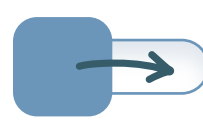
When the first iPhone came out, another version of the checkbox became popular - the Switch. It's still a multiple-choice, separate-entity, so toggling one switch, shouldn't modify the state of others. Try to differentiate the ON state by changing the color of the moving object. The switch can either be fully rounded, slightly rounded, or use sharp corners, as long as it's consistent with all your other form styles.



Off



On



Off



On

One of the main differences between checkboxes and switches is the label placement. Checkboxes are usually on the left side of the label, while the switches are generally on the right. They also differ in how their action happens either instantly or on the confirmation.



I accept the privacy policy



Sign me up for the newsletter

Enable Dark Mode



Display optional fields

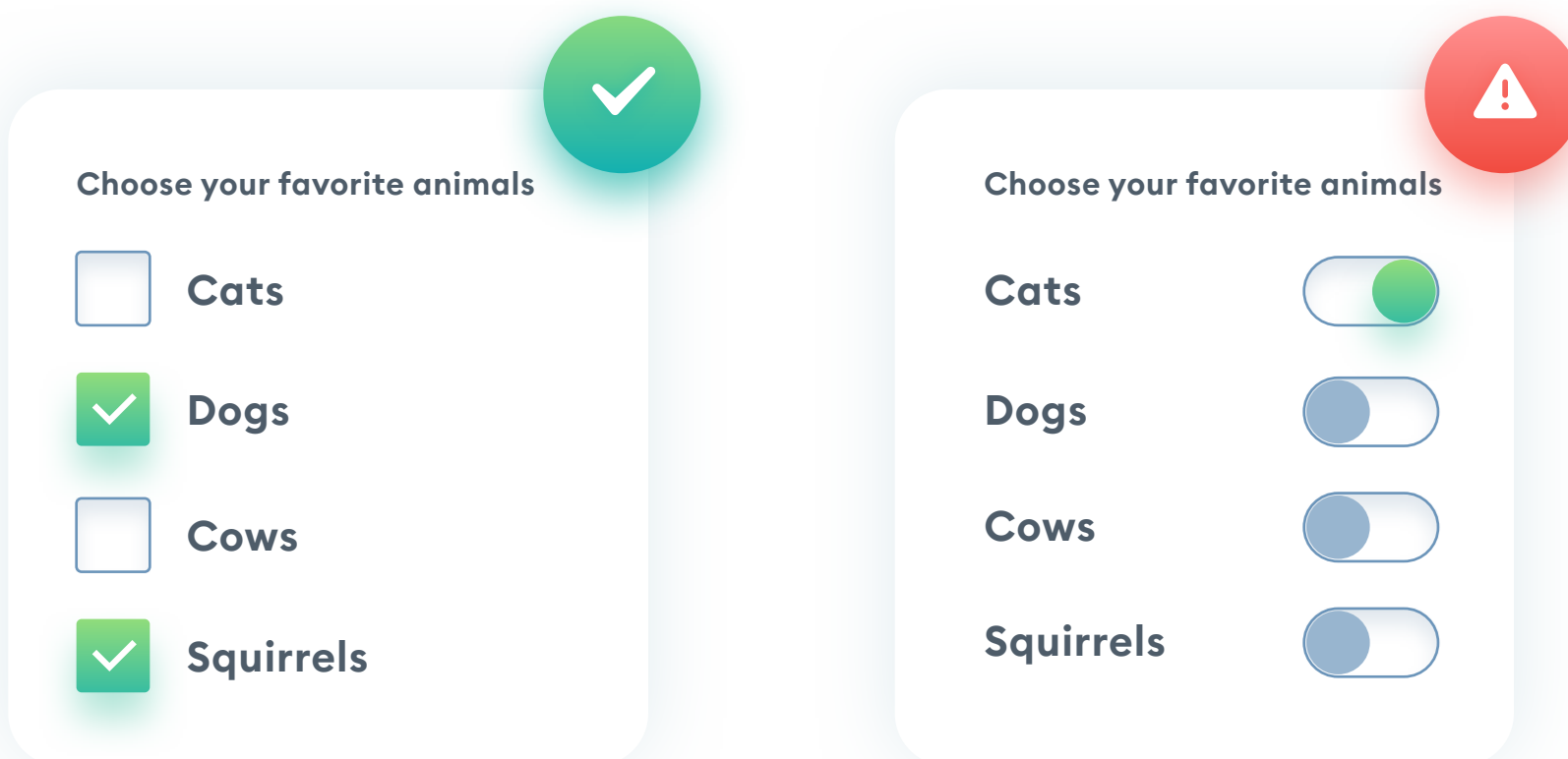


Checkboxes are usually confirmed (and take effect) after you click on the main CTA.

The switch action should happen right after it changes position, without the need to click on a confirmation button.

WHEN TO USE SWITCHES?

If you have more than 2 or 3 options to choose from, it's better to use checkboxes instead of switches. We also should use checkboxes when all the options belong to the same overall group.



Switches are better when they're individual, and they do a specific, instant action each. It means that changing the switch state happens in real-time, and we often don't need to confirm it with a button.



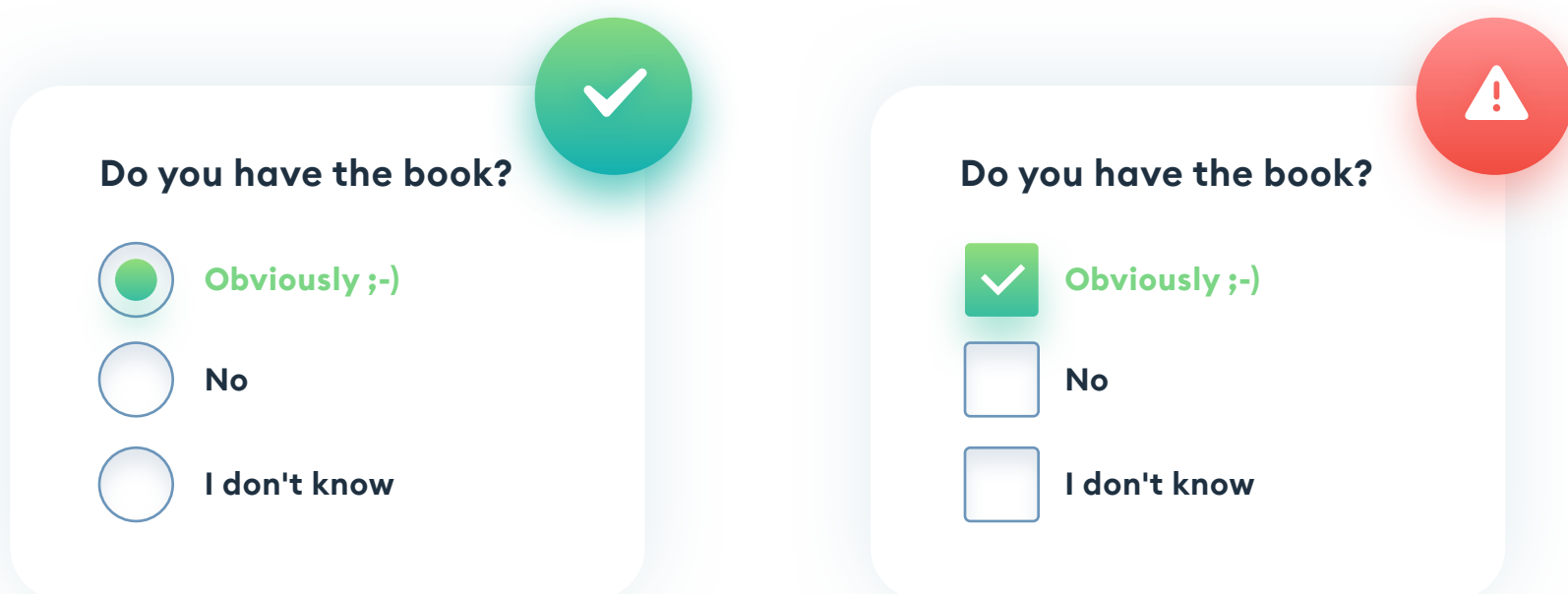
SIZE MATTERS

On mobile devices, both the switches and checkboxes need to have an active zone around them, that's at least 44p in size. The area should also include the label and description when possible. You should do it the same way on desktops, but the area can be a little smaller (around 30p).



RADIO BUTTONS

Radio buttons are a way to choose one (and only one) option from a short (2-4 item) list. If you select one of the options, it automatically deselects the one chosen previously. If you have more than four options, consider using a dropdown.



Radio buttons are used to exclusively select just one answer from many. With those types of questions checkboxes are the wrong choice, and you should always use radio-buttons.

Both checkboxes and radio buttons work best when they're aligned vertically, one after another. The action that they describe should happen after you submit the entire form.

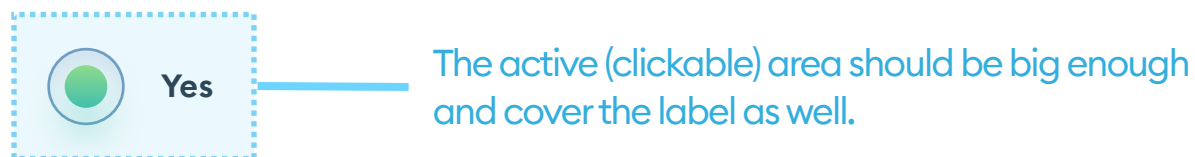
You can auto-select the most popular option, so the user can leave it or change it at will. If you're thinking of using a dropdown, but only have a shortlist of options, consider a radio button instead.

FUN FACT

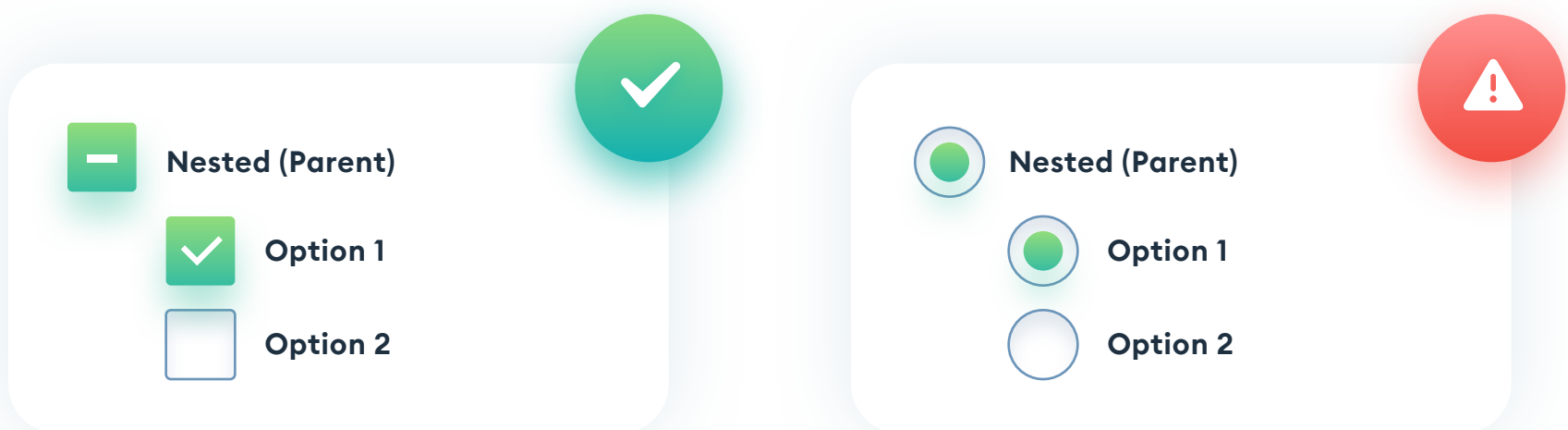
The name "radio button" comes from actual car radios from the past. The physical buttons on these devices worked in a way that pressing one was pushing out all the others.

RADIO BUTTONS

The active area should be large enough, so it's easy to both click and tap to activate the radio button. Just as with checkboxes, the label itself should also be clickable.

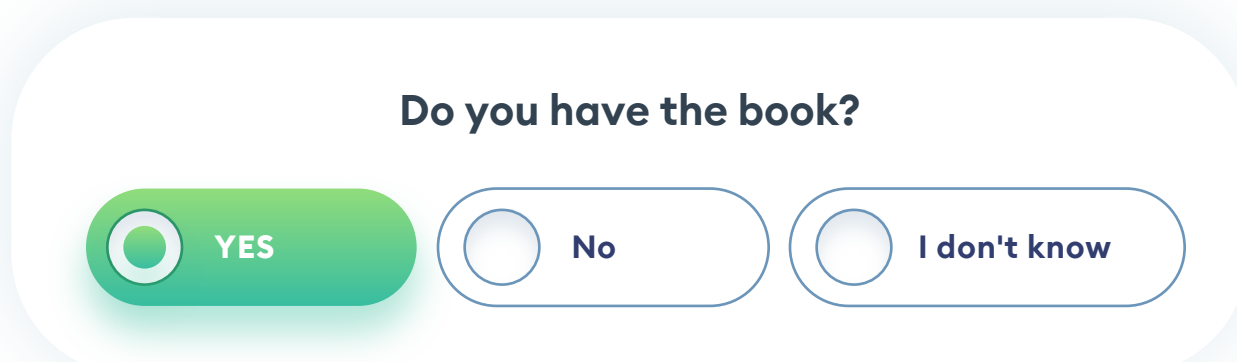


One of the main differences between checkboxes and radio-buttons is also the fact that we should avoid nesting radio-buttons. That concept works a lot better for checkboxes.



Radio buttons can also exist in a more friendly, button form. It's usually best to keep the oval "radio" area on the left side of the button, but even buttons without it can work well.

It's especially fit for mobile devices, where it's a lot easier to select it, as the perceived area is much larger than a standard radio button.



SLIDERS

A slider is a more visual way of choosing a number value. It usually snaps to a couple of most common values, and rarely allows for a full choice from a more extensive set. You can use it instead of a dropdown if there are only a couple of options.



Classic slider



Range slider

Clicking anywhere on the slider should move the indicator to that spot, or the closest approximated value. The draggable indicator should always be big enough to target with either your finger or the cursor comfortably. On mobile devices, you can create a teardrop-shaped tooltip that shows the current value while dragging the slider, so our finger does not cover it.

The main rule for fantastic sliders is clarity and ease of use. Dragging a small, easy to miss element can be a frustrating ordeal for the user.

Avoid using sliders if they have less than four possible values. In cases like that, it's best to replace them with a radio button set.



CONSISTENT STYLES

Choosing the border and fill styles, keep in mind, to use them consistently across the entire form. Starting with the text-field and its border, use the same stroke width and color in all the following fields you design.

Before you start building the actual forms, try to create all the necessary components first, on a separate artboard, to find inconsistencies and fix them quickly.

E-mail

Years of experience

 ▾

Comments

Skill level

 Low Mid High

8

Interests

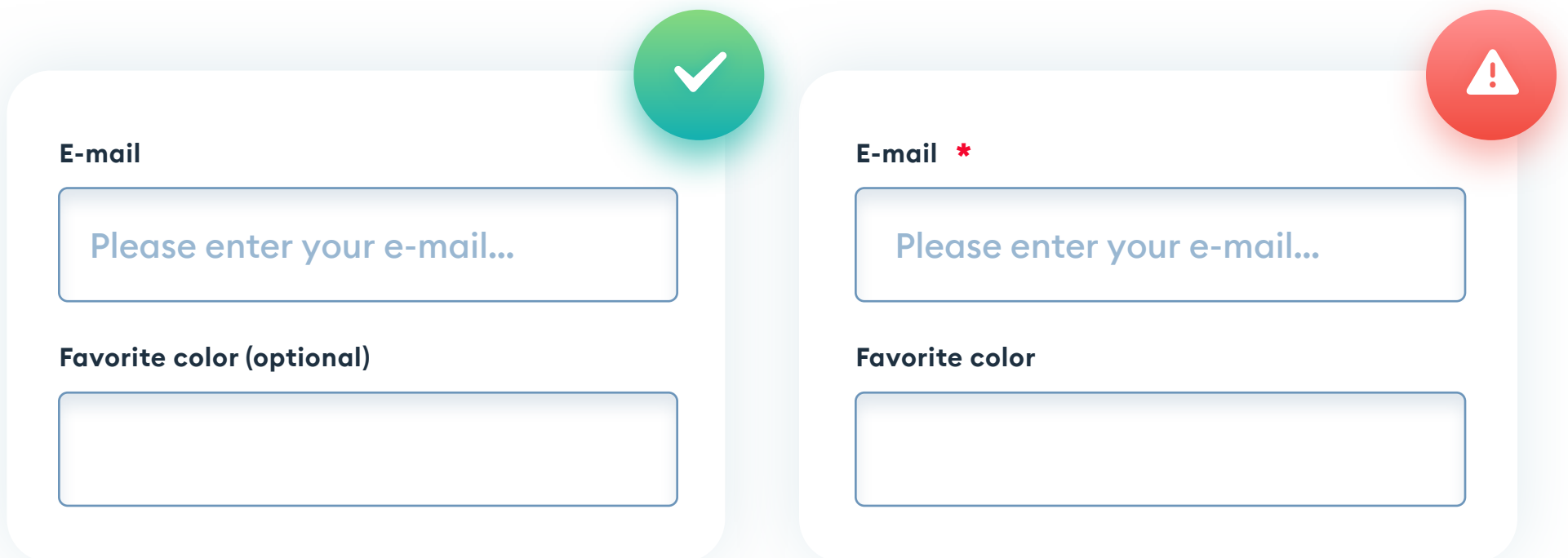
 UI + UX
 Research

Do you have the book?

 Yes
 No

GOOD IDEAS

While creating a form, remember the rule of "less is more." It's even more relevant when it comes to good form design. If a field isn't necessary, don't add it - it will positively affect your conversion rate: the fewer fields, the better.



The image shows two side-by-side form designs. The left form is marked with a green checkmark icon, indicating it is a good design. It has two input fields: "E-mail" and "Favorite color (optional)". The right form is marked with a red warning icon, indicating it is a bad design. It has two input fields: "E-mail *" and "Favorite color".

If you need the non-essential fields (optional) in your form, try to avoid the "red asterisk" to signify the necessary ones. As your form should mostly be about the required fields, it's a lot better only to inform the users when a field is optional. Keep the necessary fields distraction-free.

The best forms should only have one button. It will make it a lot easier and frictionless to take that final submitting action. If, for some reason, you need more than one button, always differentiate them highlighting the most important one. The user needs to know which button to activate right away - without thinking.



The image shows two side-by-side button designs. The left design is marked with a green checkmark icon, indicating it is a good design. It features two buttons: a white "Cancel" button and a green "Log in" button. The right design is marked with a red warning icon, indicating it is a bad design. It features two green buttons: "Cancel" and "Log in".

“

**A good form
should only have
the essential
fields.”**

GOOD IDEAS

Our call to action button should precisely state what is going to happen when the user activates it. Avoid labels like "Next" or "Forward" when possible.

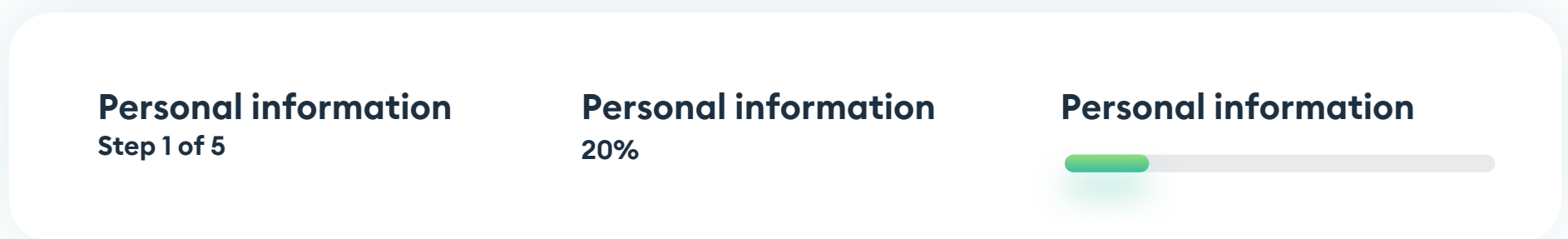


If your form is quite long, and you don't want to break it into separate pages, try to segment them by grouping similar fields. It will make it easier to understand that the form is a combination of a couple of overall concepts.

The image shows two examples of form segmentation. The first example, marked with a green checkmark, shows a form with two sections: 'Login info' and 'Personal info'. The 'Login info' section contains 'E-mail' and 'Password' fields. The 'Personal info' section contains 'Your name' and 'Your date of birth (DD/MM/YYYY)' fields. The second example, marked with a red warning icon, shows a form with four ungrouped fields: 'Email', 'Password', 'Your name', and 'Your date of birth'.

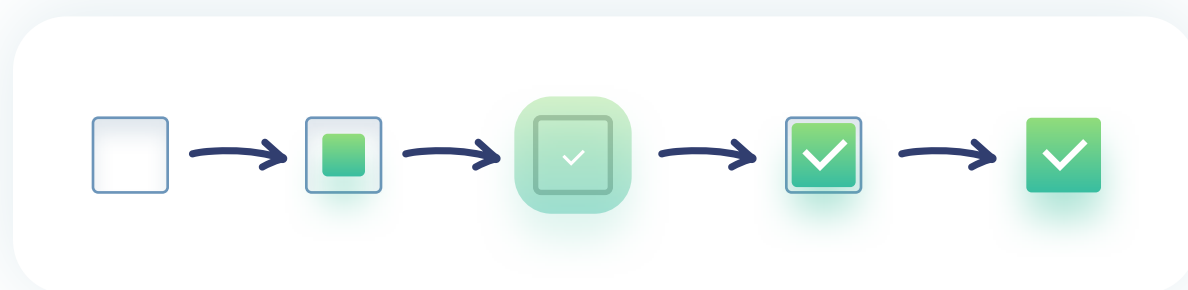
GOOD IDEAS

Split long forms into cohesive, separate steps. Remember to inform your user on their progress with a % value, a number of steps, or a slider. They need to be aware of where they are in the form filling process at all times.



ANIMATION AND MICROINTERACTIONS

Use animation (but subtle!) and microinteractions to give your form a memorable feel and make it more fun to use. Changing a checkbox state doesn't have to be switching an image. It can bounce with a nice, small transition.



BE CREATIVE!

Forms don't have to be boring. You can create an animated character that follows the active field with its gaze and smiles when the user clicks submit. There are no limits for creative ideas, but the form elements themselves need to be predictable at all times.

KEYBOARDS

If you're designing a form for a mobile device, remember to use the right keyboards. When the email fields are active, a keyboard with a quickly accessible @ sign should appear. If, in a particular field, the user can only enter numbers, use a number pad.

Bank transfer

1	2	3
4	5	6
7	8	9
.	0	✕

Keep that fact in mind while prototyping your forms. Show the difference between keyboards and field types to your developers so that they can pick the right one in code. Of course, on desktops, we don't use an onscreen keyboard (in most cases), so these rules only affect mobile designs.



18.

Modals &

Popups

Modals & Popups

THE BASICS

There are a couple of modules that appear above our interface, showing additional actions or information. They can appear on their own (not recommended), or be a result of a specific user action.

The most common types of these modules are popups, overlays, and action sheets. The first two usually have a small X in their top right corner to close them. Clicking or tapping outside the window should yield the same result. Action sheets are a bit different - they usually don't have an X, but rather a cancel action near the bottom. You can also swipe them away (usually downwards).



Popup / Alert

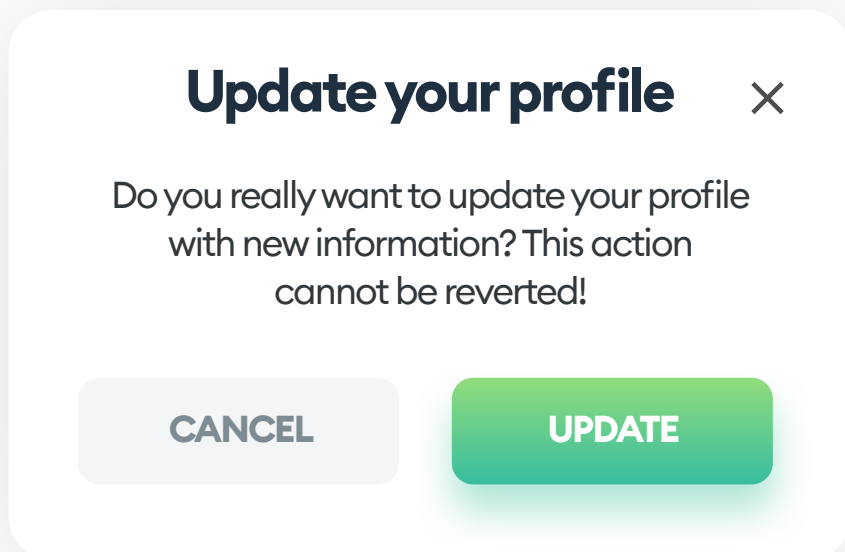
Overlay

Action sheet

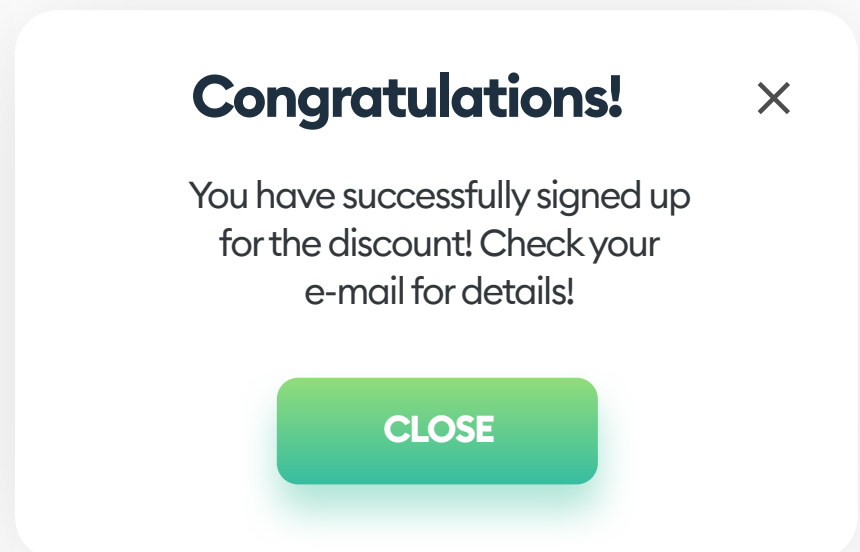
Popups work best on desktops, overlays are pretty universal, while action-sheets are mostly a mobile pattern. In general, however, using any of these modules should be your last resort. They can be extremely frustrating for your users.

POPUPS

Popups are the most popular type of top-level prompts. The main types of popups include a success (or failure), confirmation (a question, that you answer), and signup (for a newsletter).



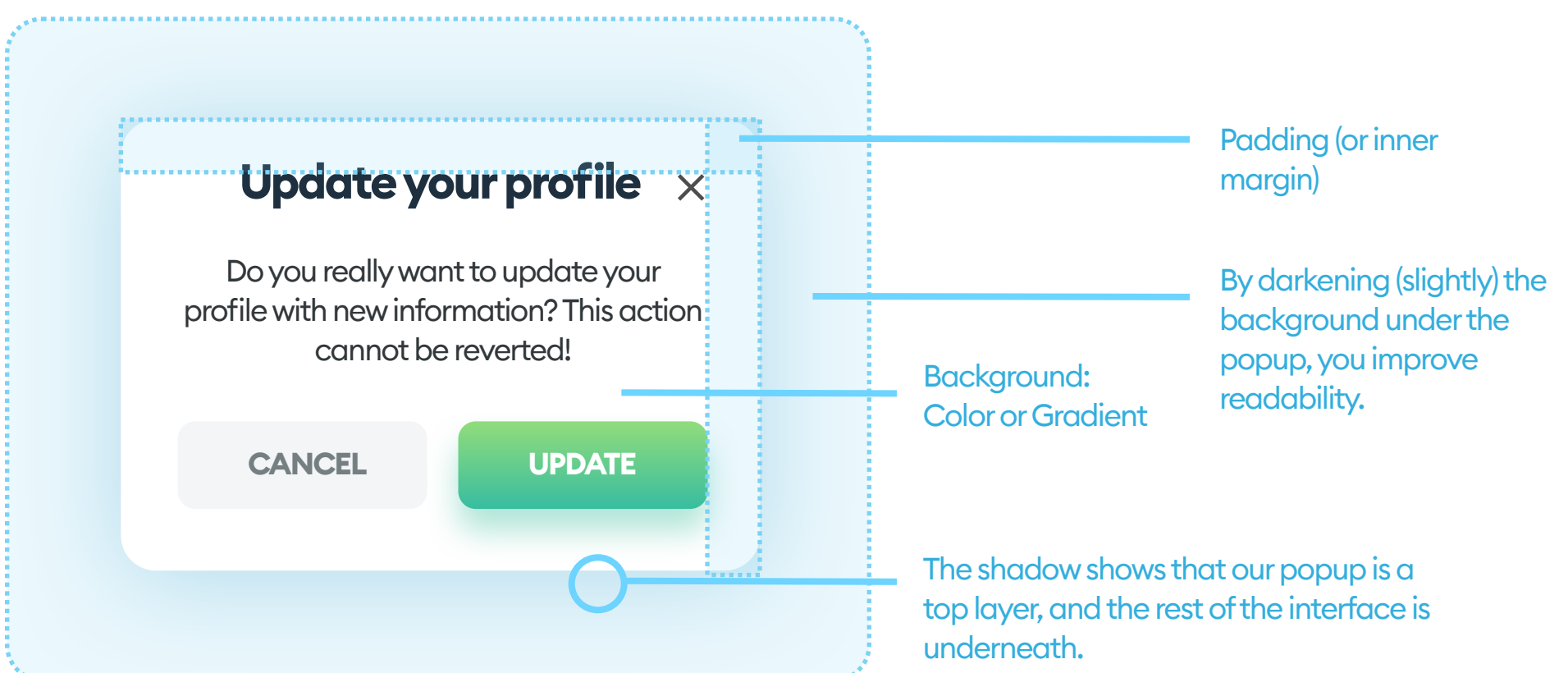
Confirmation popup



Success notification popup

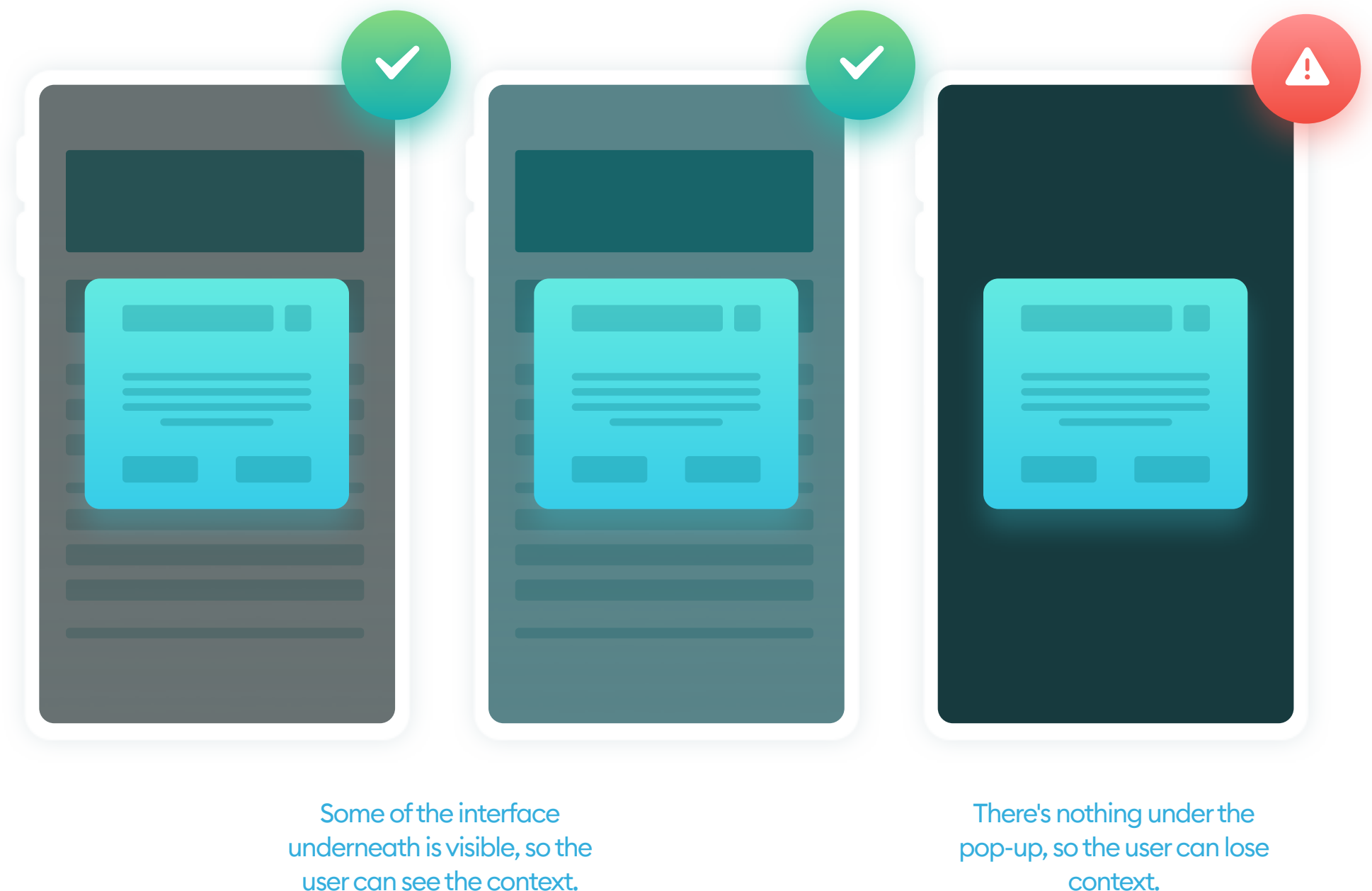
BUILDING A POPUP

A typical popup is a window with a title, description (optional), at least one button, and an X to close it. Clicking outside the area should also result in closing the popup. When the only action available is closing the pop-up, ALWAYS use both the X and the button to do it.



BACKGROUND OVERLAY

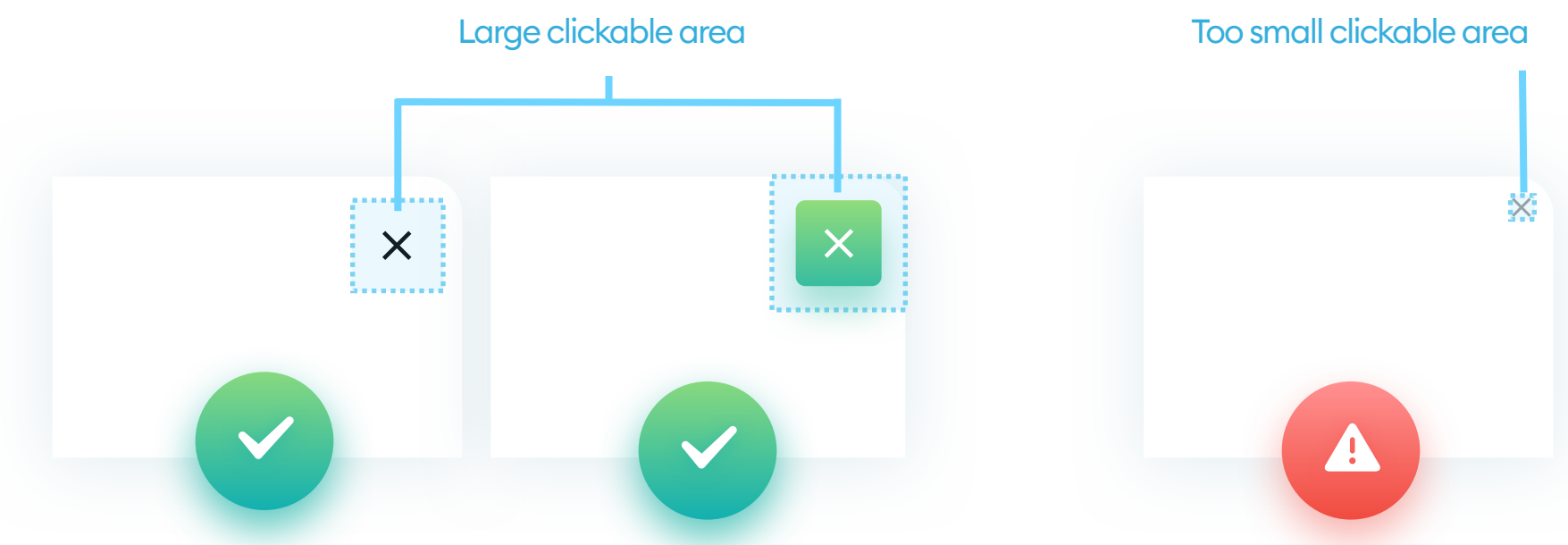
You can make the popup more visible by adding an overlay to the background under it. In most cases, it's a dark hue with about 70% opacity value. Having it slightly transparent allows the user to be still able to discern the context of the entire screen. In other words, they understand and remember where the popup stands in the interface.



You can use full black (#000000) at about 70-80% opacity, or a dark shade of your primary color for a more consistent result. In any case, make sure there's enough contrast between the popup and your background while keeping the screen under at least slightly visible.

GOOD IDEAS

Popups don't need to be utterly frustrating! Remember to make it readable, clear, and have all the active elements easy to click.



Your closing X button should always be the right size, have enough contrast, and a large enough active (clickable) area. A good idea is simply creating the X as a button with a background, so it's easy to understand the clickable area.

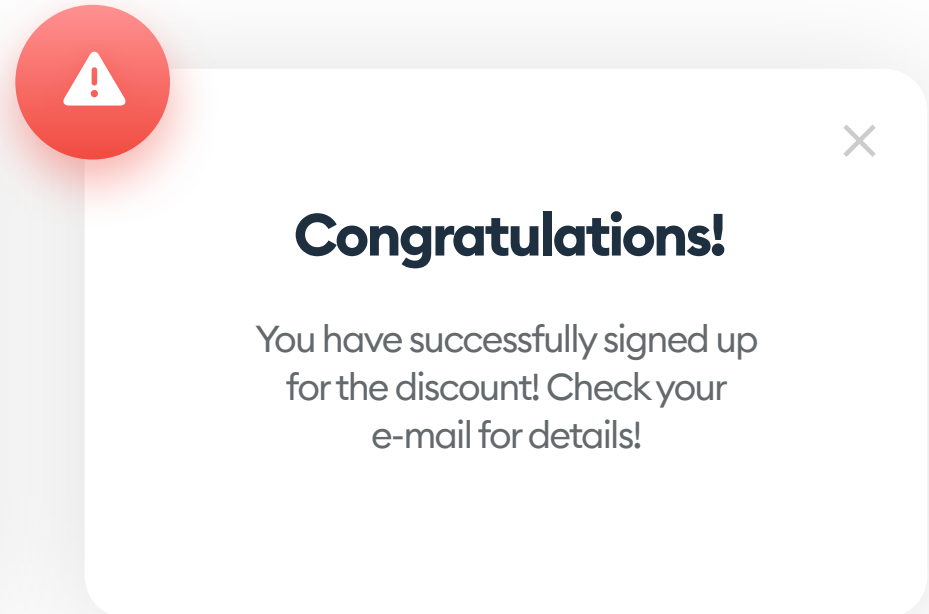
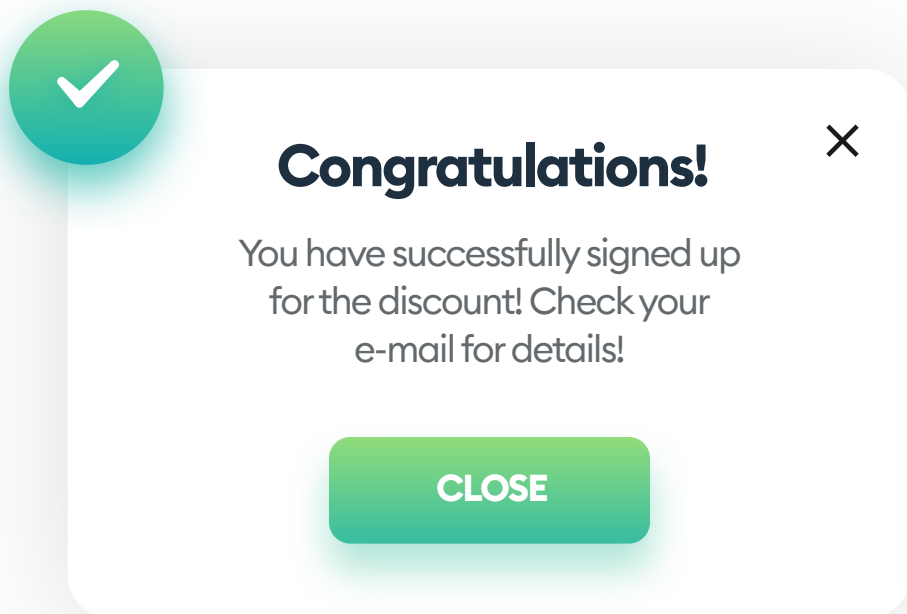
That clickable area (blue square) should always be larger than the X itself. Good values for the active region are at least 44x44 points for mobile and 32x32 for desktop, while the X itself can be smaller than that.

One of the worst things you can do is creating a super-small X button with very low contrast. If the active area is also small, your users need to be very precise while clicking to achieve the desired action. If clicking outside the popup doesn't close it, you end up with one of the most frustrating UI mistakes ever.

Avoid placing your X button right near the edge of the popup window. It needs enough whitespace around it to be comfortably visible.

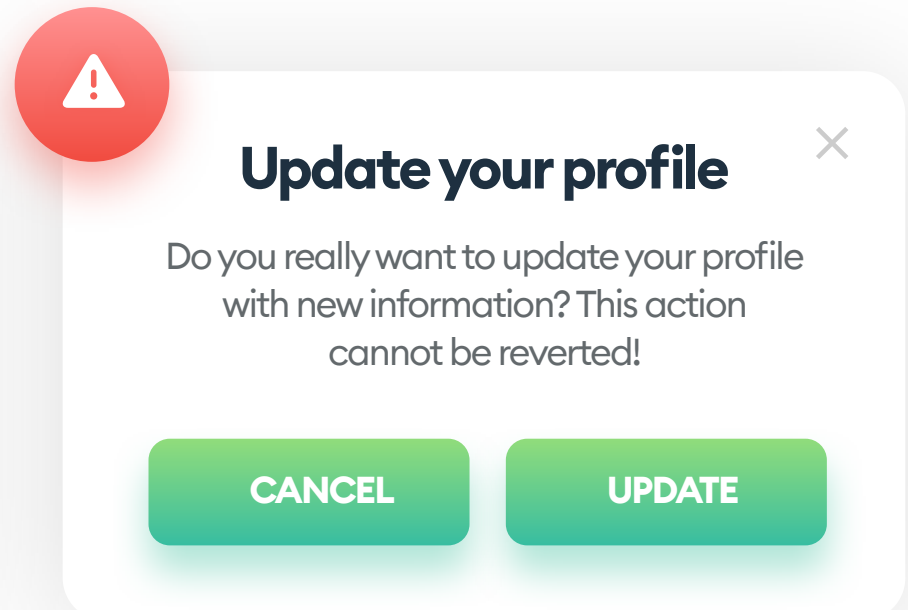
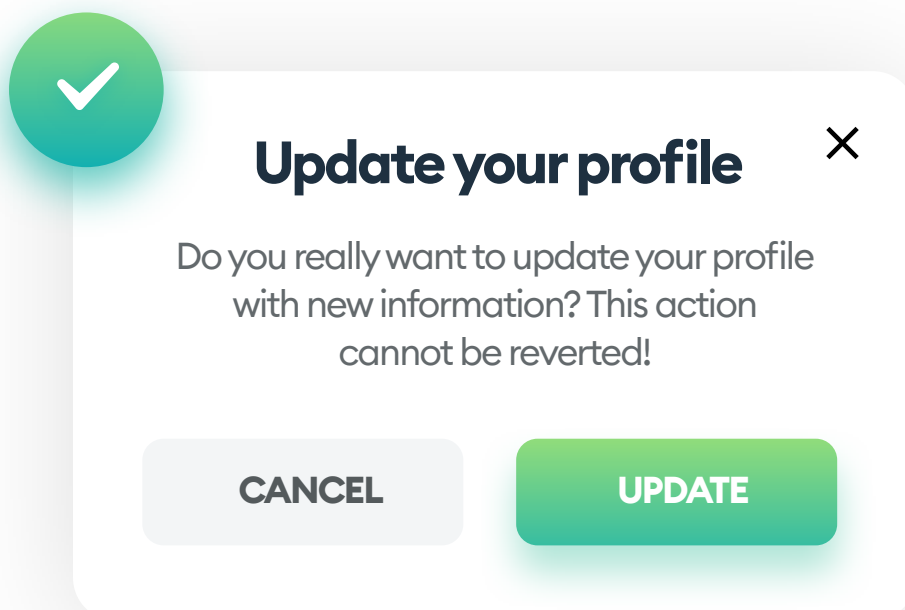
GOOD IDEAS

As said before, even if your X is clear and accessible, you should always use an additional closing button at the bottom of your popup window.



CLEAR ACTION HIERARCHY

If you're using more than one button, make sure to differentiate the main action from the secondary one. It will help your users quickly understand the hierarchy and pick the desired outcome. A widespread mistake is the cancel button having the same "visual weight" as the main Call to Action.



GOOD IDEAS

Design your popups with the "mobile-first" approach, as it helps with picking the right widths and button sizes that work across all devices.



If you design your popups to fit on a mobile devices, they will work just as well on larger screens.

LANGUAGE

Use short, precise messages in your popups. Avoid using more than two sentences of description and keep your button labels explicit. Don't use "Next" or "OK" on the buttons. The labels should always describe the action that happens when you click.

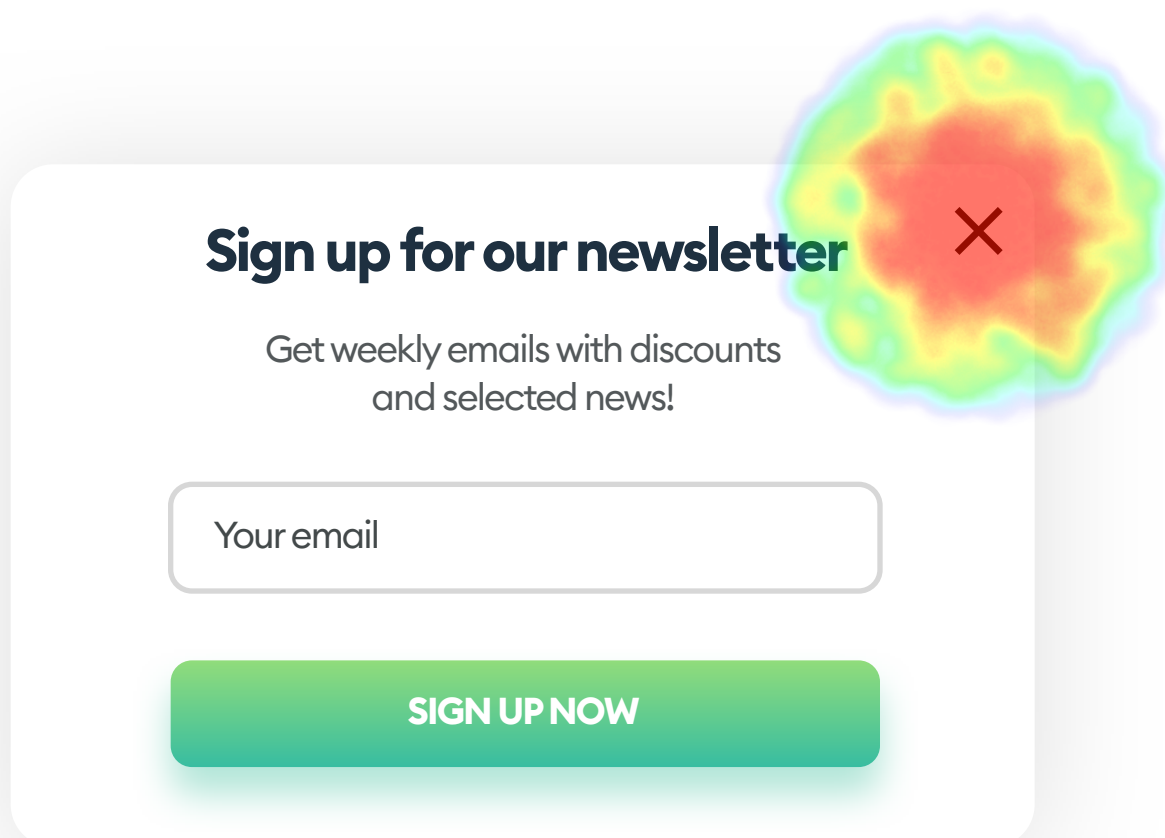
POPUPS SHOULD APPEAR DUE TO USER ACTION

At all costs, avoid automatic popups (either on entry or while trying to exit the website). For a pop-up to appear, there needs to be a conscious action taken by the user.

POPUP FORMS

Popups with sign-up forms are the result of statistics meeting lousy user experience. The users don't expect these popups, so most of them instinctively close the window. While it may yield a small number of subscribers, they often convince more users NOT to trust your brand.

Consider having a sign-up form on the page itself, without resorting to popups. Doing that will help you get genuinely interested users to sign-up.



Even if statistics are showing a single-digit percentage increase in sign-ups, the unmeasurable negative effect can have a much stronger impact on your brand.

Do you know what can also help with getting more subscribers? Quality content! Make things so well that people will be lining up to sign-up to receive it!

“

**The X is the most
popular element
your users
will click.”**

OVERLAYS

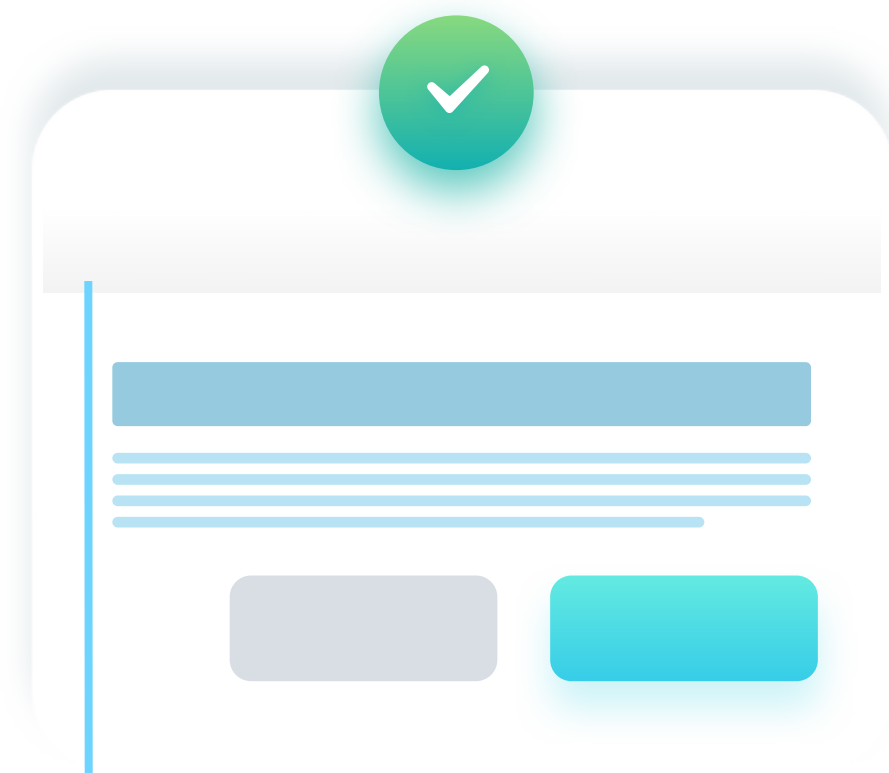
Overlays are small panels attached to the edge of the screen - usually to the bottom. They are a good choice for non-essential information or agreements. The most popular overlay is the cookie information prompt, traditionally positioned at the bottom of the display. Just as with popups, you can close an overlay by clicking the X, or one of the buttons on it.

Most of the good popup practices regarding readability, spacing, and active item sizes apply to overlays as well. The main difference is that clicking outside the overlay area does not close the window.

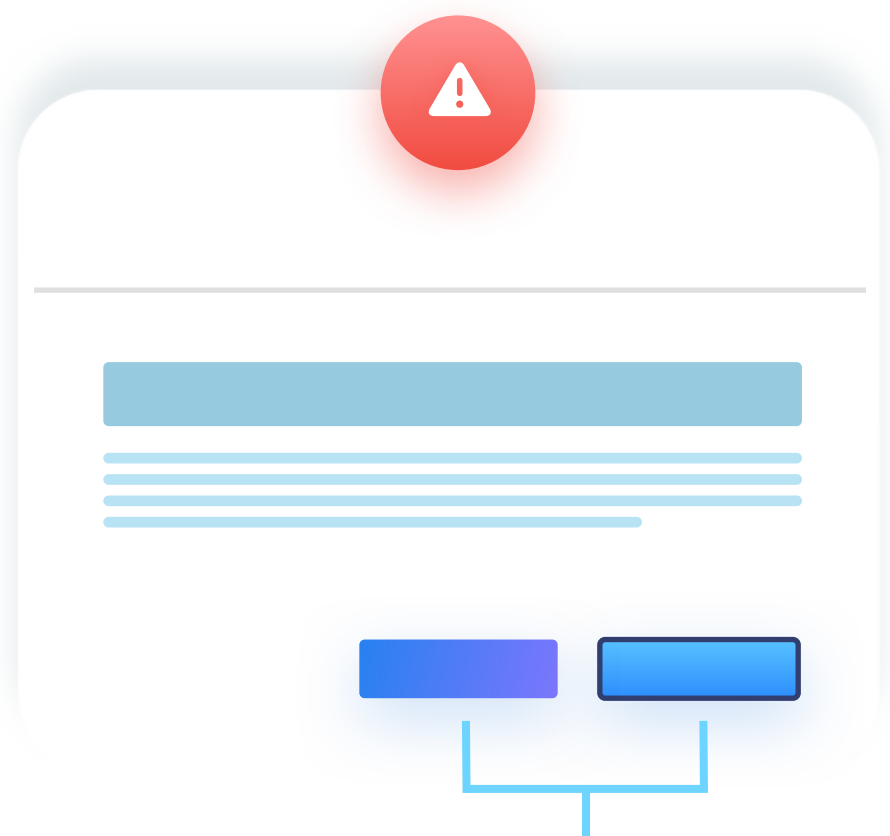


When using overlay prompts, avoid the darkening of the rest of the interface, and have a clear, accessible way of minimizing the window.

OVERLAYS



The shadow helps to show that the overlay is above the rest of the page. Buttons are using the general app style.



Inconsistent buttons and no shadow make this overlay harder to use.

Overlays should have a clear indication that they're above the rest of the UI. You can do it with a subtle shadow or a contrasting border. Make your buttons big enough for both desktop and mobile and keep proper whitespace for readability.

If you're using a generic overlay (like a third-party cookie prompt), the spacings, fonts, and buttons can be very inconsistent with the rest of your UI.

CONTENT WIDTH

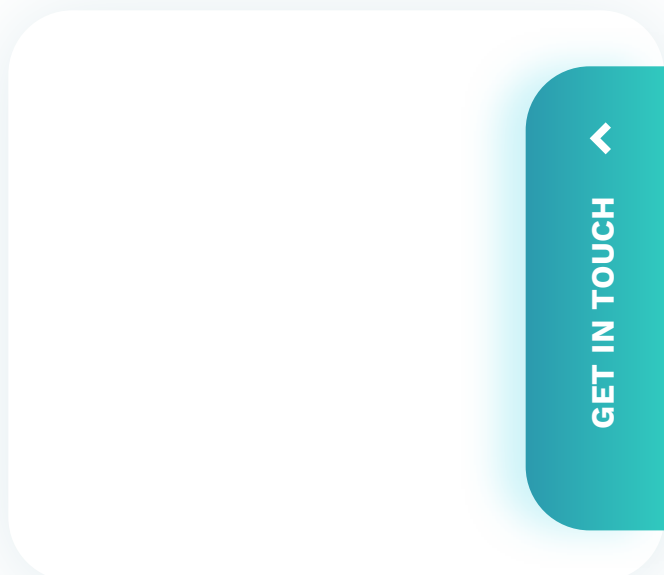
If you're designing an overlay for desktop use, remember to keep your content between 600 and 800 pixels wide. You can leave the sides empty, but don't stretch the content to fit your screen above 800 pixels. Long lines of text are especially challenging to read.



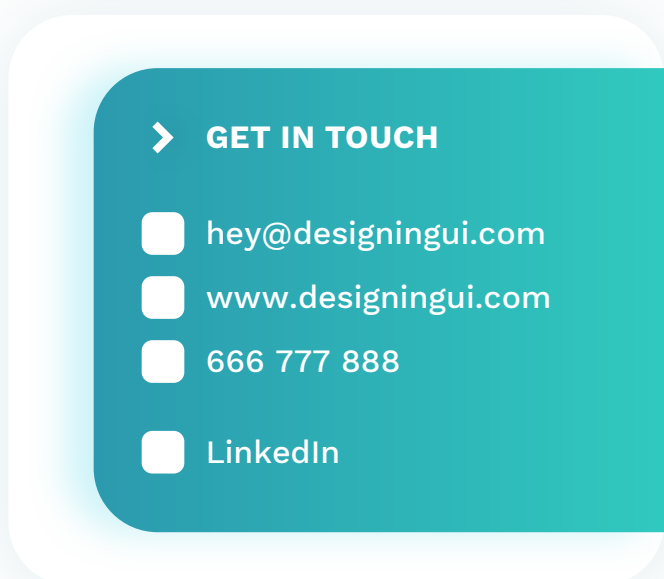
EXPANDING OVERLAYS

An expanding overlay is usually a small stripe attached to the side of the browser window. On activation, it expands into a larger overlay with either information or options. It can be a simple contact form, a chatbot, or a quick share action for social media. While it saves space, it also obscures the actions hidden within, so try not to overuse this pattern. One per page is more than enough.

Hiding content from the user is not the optimal choice, so consider whether you need an expanding overlay in your project. Because of its hidden nature, some users may never expand it and use the options hidden within. Avoid putting essential navigation or info in one of these, and only use them if your page is already complex.



As this element expands, it's good to show it using a direction arrow. On click, the indicator should turn around to show what happens when you click again.



The arrow itself should, of course, be clickable, but it's more of a suggestion for action, so have the entire heading of the overlay clickable.

ACTION SHEETS

Action sheets are a natively mobile component that usually comes out from the bottom of the screen. It can be a set of filters, display options, sharing, or sorting. The most common use is the share-sheet, which allows the user to pick a way to share content online.

Actions inside action-sheets can be a set of vertically placed buttons or switches.



If you're designing your own action-sheet (instead of using a native one), remember to apply high enough contrast, clear labels, and an overall readable, hierarchical structure.

It can also help to highlight the most popular action.

ACTION SHEETS

If your action-sheet is nearly, or completely fullscreen, you can consider adding an X button in the top right corner. If it only takes up to three-quarters of the screen, consider communicating its swipeable nature by adding a "flat-arrow" or a horizontal line on top.



Action-sheets can also have several levels of visibility. They can show the most popular actions first, and expand more by dragging the card up. Remember to design every state of an action-sheet if you're making a completely custom component.

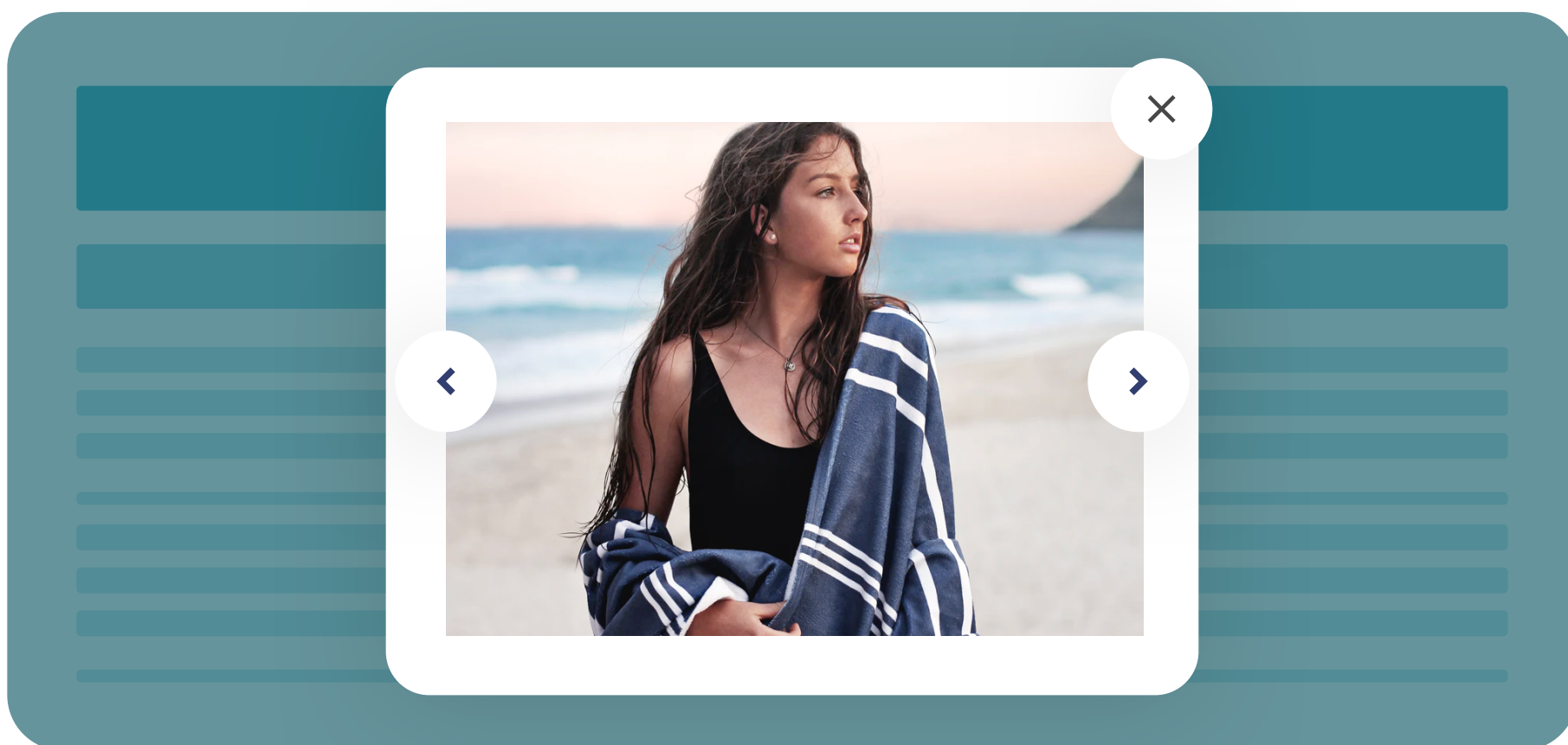
CLEAR INTERACTION

One of the essential features of action sheets is the transition that they use to appear. They should visibly slide-in from the bottom, so the user understands it's possible to slide/swipe them back down.

That animation gives your users the context on how to use the action sheet right away.

LIGHTBOX

Lightboxes are a type of popup that allows for displaying a photo (or video) gallery. It's an excellent approach to showcase larger photos in a distraction-free way. Just as with standard popups, make sure you darken the rest of the interface for a better viewing experience. Make all the navigation buttons clear and visible (like scrolling arrows, thumbnails, or the closing X).

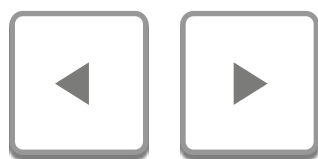


EASY TO USE NAVIGATION

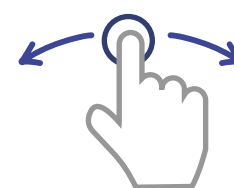
Pay special attention to your lightbox navigation. It should always allow for all of the three possible methods - clicking on arrow buttons, touch gestures, and keyboard arrows. Don't stick to just one navigation method - implement them all to cater to all of your users.



Cursor-based navigation



Keyboard navigation



Touch navigation

TOOLTIPS

A tooltip is a small, informative, user-activated pop-over. In most cases, it appears when clicking on a question mark sign or a little **i** icon. In rare instances, tooltips can also have modal-like qualities, including some simple actions (like add this person to friends).

It can appear on click (or tap) or hover, with the first option being the better one here.



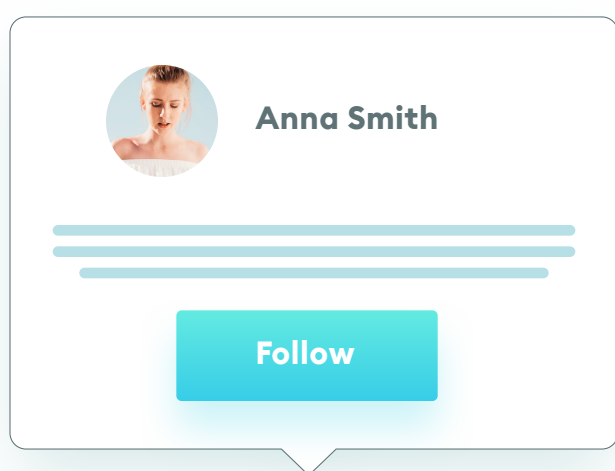
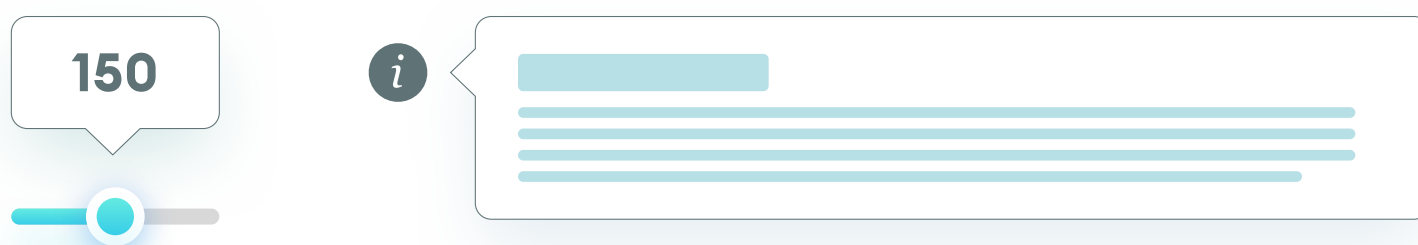
A tooltip is a rectangular shape with a small triangle pointing towards the object that activated it. Keep enough whitespace inside your tooltips for the text to be readable and make sure it has the proper contrast. The arrow/triangle shape can point in any direction or can depend on where the tooltip appears. That way, it can automatically adjust to the amount of space it has to display, and won't end up behind the screen.



A tooltip can appear after clicking on an icon, button, link, slider, or a chart. When possible, try to display the helpful information inline, without the need for a tooltip in the first place.

TOOLTIPS

Most tooltips are informative (helpful hints) and non-interactive. If you plan to add an action to your tooltip, try to keep it as simple as possible - one large enough button will be enough. For more complicated interactions, consider replacing the tooltip with a modal or a popup.



Anna Smith

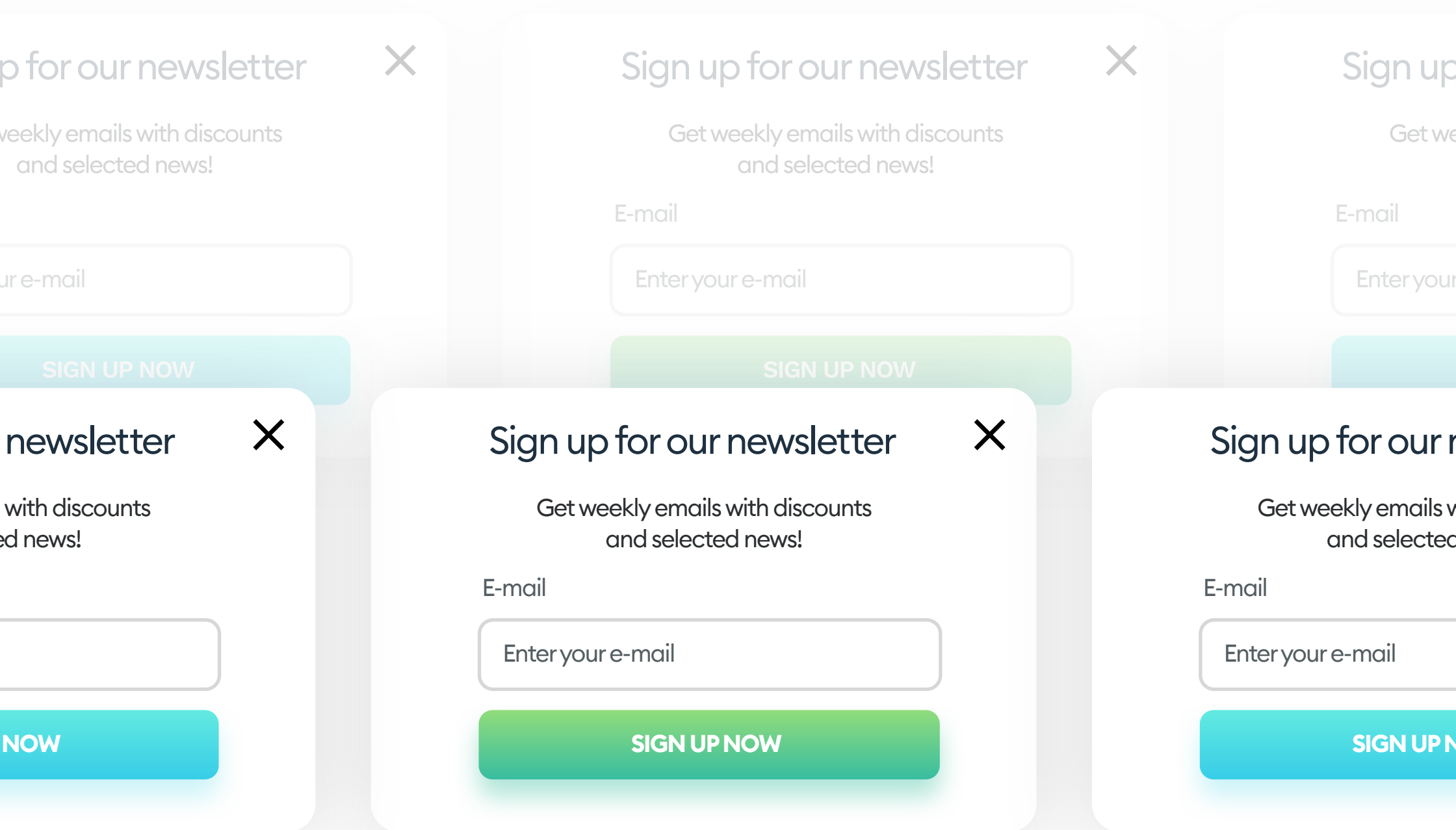
Make your tooltips clear, visible, and readable. Apply the same rule you use to design popups and buttons.

Avoid using long, wordy descriptions inside tooltips. Keep them short and to the point.

TOOLTIPS ON MOBILE DEVICES

Tooltips using a mouse-over function to activate them won't work well on a mobile device. While designing for touch screens, try to avoid tooltips altogether and display the helpful hints inline with your content.

WRAP-UP



Be cautious about using anything that jumps out at the user. Consider popups and tooltips as your last resort. Most of the overlaid-windows are simply unnatural and annoying. If possible, design your interactions without them. The exception here is the mobile action-sheet, which the users understand well and is the natural choice for actions like sharing content.

When designing an overlay or a popup, remember to keep the readability high, proper whitespace, and, most importantly, the size of your active elements. Too small, hard to click X-button can frustrate your user enough even to leave your product altogether.

Keep the design consistent with the rest of your project. If your popup looks completely different than the rest of your UI, the user may think it came from an external source and grow suspicious.

“

**Main rule of popups
is to only use them
when absolutely
unavoidable.”**



19.

Navigation

Navigation

THE BASICS

Navigation is an essential part of the User Interface. Even the smallest problems with how it works can lead to massive user drop-offs. Hard to use, or faulty navigation can make it nearly impossible to use your app or website. That's why it's crucial to design your navigation with a lot of attention to detail. There are three main types of navigation - visible, hidden, and contextual.

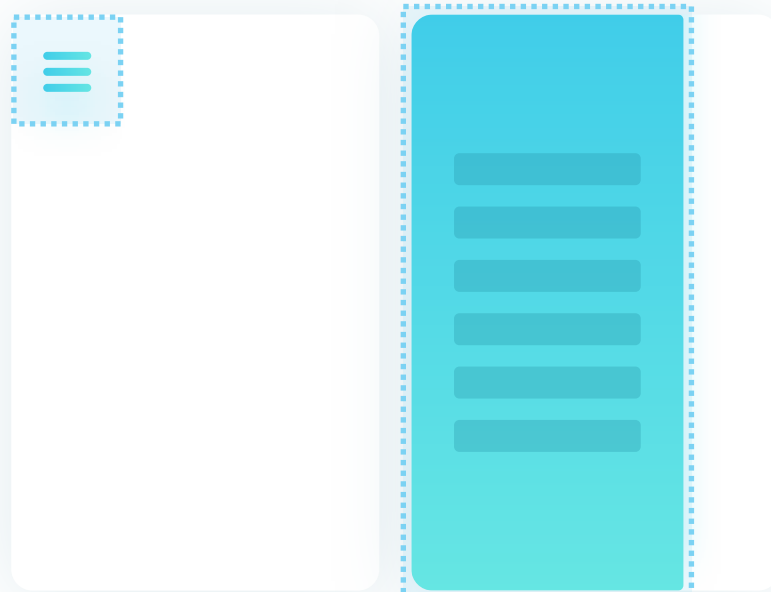
Visible - It's always on the screen and (in most cases) the content scrolls underneath it. In most cases, it's a set of tabs, with the currently active one highlighted.

Hidden - This navigation pattern relies on a hidden menu that is not visible before it's activated. The most common example is the "hamburger menu."

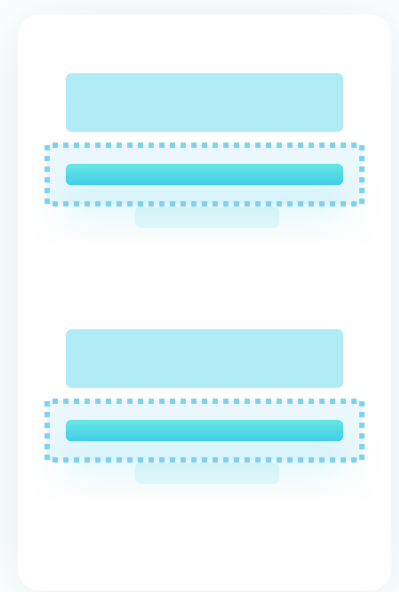
Contextual - This includes active links or buttons within the object itself. The best example of this is clickable categories in the product listing that take you to an additional page from within the product page.



Visible



Hidden



Contextual

“

**All menu items
need to be big enough
to comfortably touch
with your finger.”**

VISIBLE NAVIGATION

Visible navigation should always be your starting point while designing menus. It's considered the best option for most interfaces, both web and mobile. Most often, visible navigation is a set of text-links, buttons, icons, or a mix of these elements (like icons with a text label).

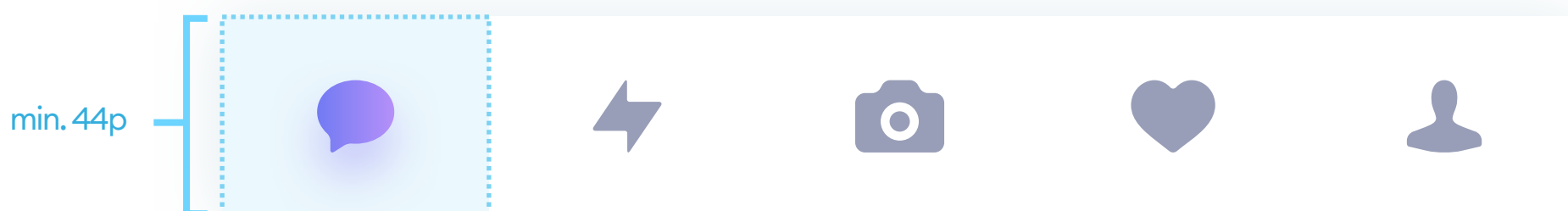
The best spot to place the navigation bar on mobile devices is at the bottom of the screen, so it's easily accessible with one hand.

Most consumer products should try to fit their main navigation on mobile into up to five main categories. You can go as high as seven on desktops. Try not to exceed that number, however, as it's the sweet spot for ease of use. Special, professional products may have to break the rule, but most of the rest should be able to simplify their nav patterns.

TAB BAR

Tab Bar is the official name for iOS bottom navigation, but you can use it as a general description of bottom tabs. The most popular version is a set of horizontally distributed icons with small labels under them.

You can design a Tab Bar without the labels, but in general, it's best to include them. Not all users interpret icons the same way, so adding a label makes it clear what the menu items are. That is essential for all non-standard icons. Your tabs also need to be the right size. If you're using icons alone, the minimum height of the bottom bar on mobile should be 44 points. If you're adding labels, start at around 60 points.



TAB WIDTH

When designing the tab bar, start by dividing your screen width by the number of tabs and create a rectangle guideline for each one. With five tabs, you create five rectangles, each 1/5th the width of the screen. Choose a red fill and a slightly different opacity for each one. That will help with icon and text alignment, and you can hide the guides when all is set and done. You should end up with something like this:

Opacity levels

80%

60%

40%

20%

10%



Each element is 1/5 the screen width.

Do the same with three tabs, setting a different opacity for each rectangle so that you can easily distinguish where they begin and end. If your screen width doesn't perfectly divide into the desired number of tabs (and we don't want half-pixels ;) try adding a 1px line between the tabs. You'll end up with one less divider line, than the number of tabs you need. With three tabs, it amounts to two one-pixel gaps.

80%

60%

40%



Each element is a third of the screen width.

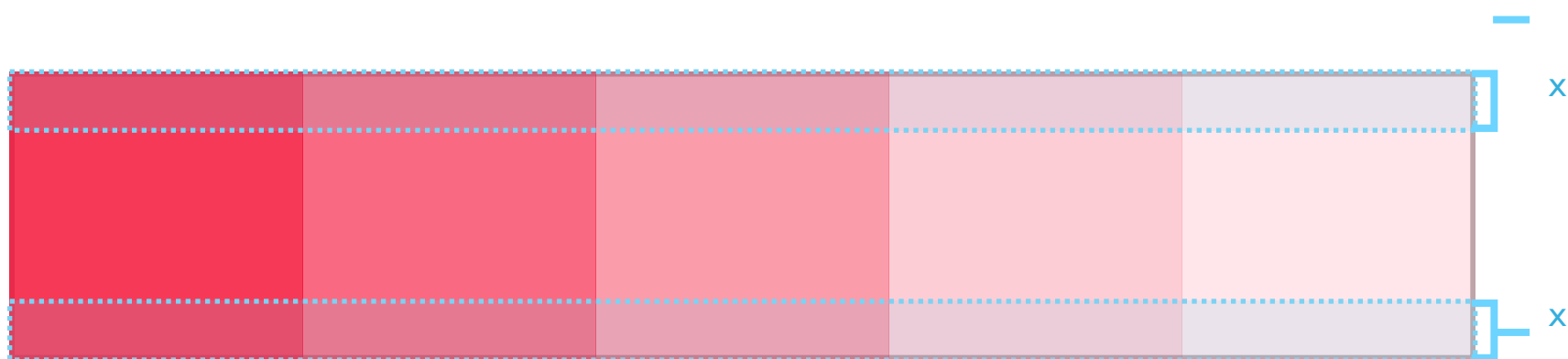


Each element is 1/3 of the screen width with 2 pixels subtracted.

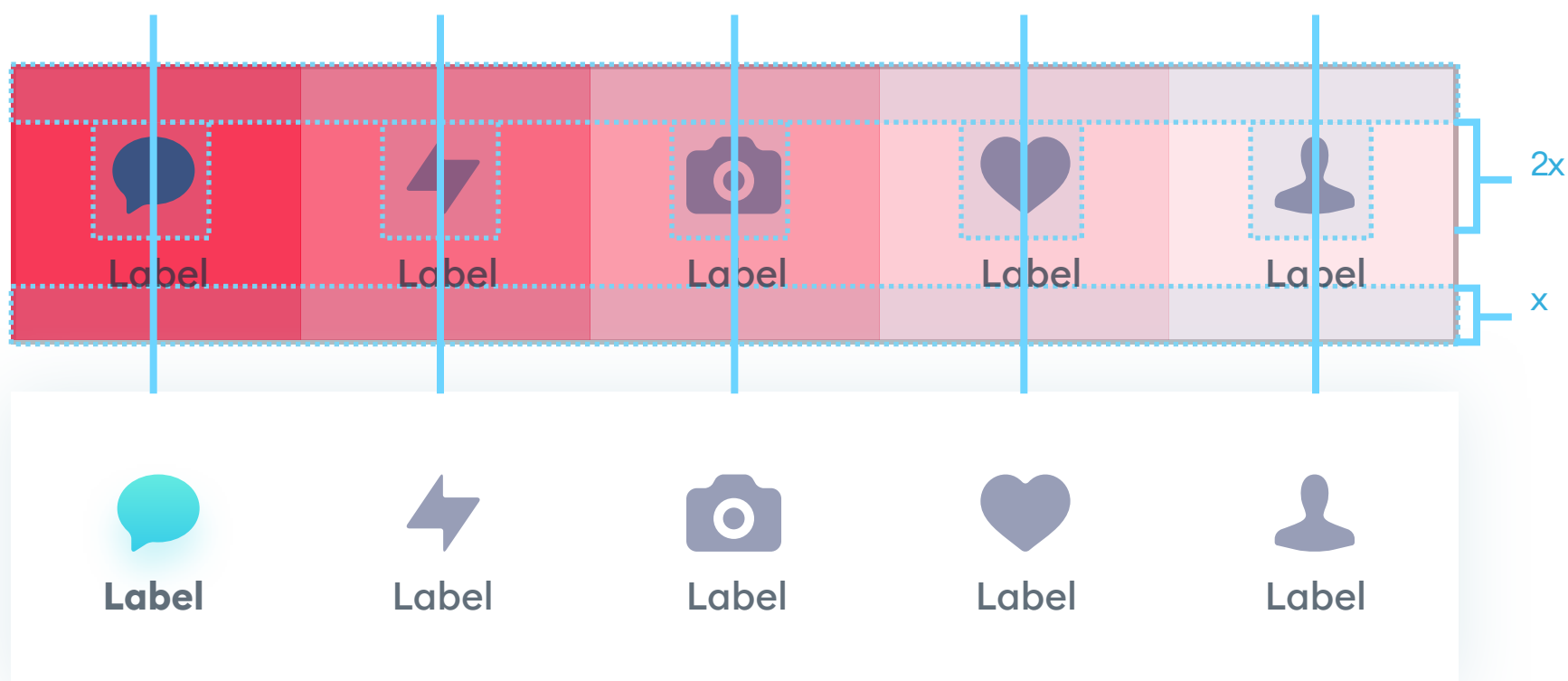
TAB WIDTH

Once you have all your tabs as rectangles, group them and name that group "tab grid." It will be a huge help when aligning your actual tabs. When grouped, it's also easy to turn it on and off when needed.

The next step is setting the safe space from the top and bottom edge of your tabs. You can start with your grid base-value (8 or 10) and use that space as your margin. It means neither text nor the icons can go into those areas and have to fit between them.

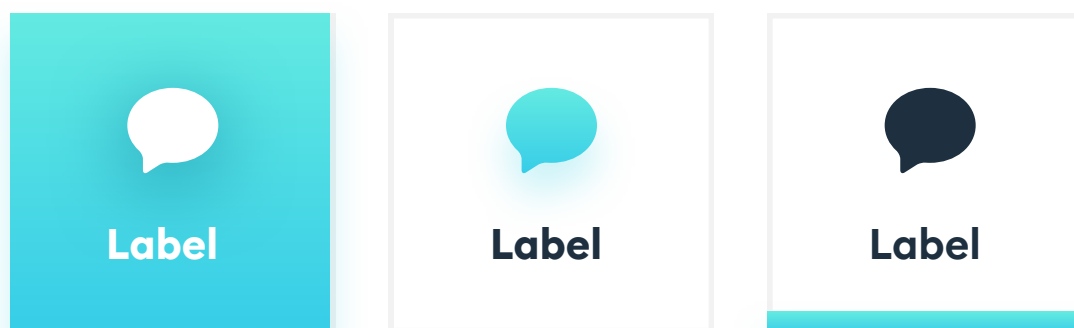


Now it's time to add your icons and labels. Make sure that the icon safe space is around twice the size of your grid base number (2X). You can align the label baseline to the bottom safe space. Test the vertical alignment by adding grid lines through the center of your red rectangles and make sure everything is in the right place.



ACTIVE TAB

The selected (active) tab should be instantly recognizable. You can do a background fill, change the font (weight or color), modify the icon color or size, or underline the entire tab.

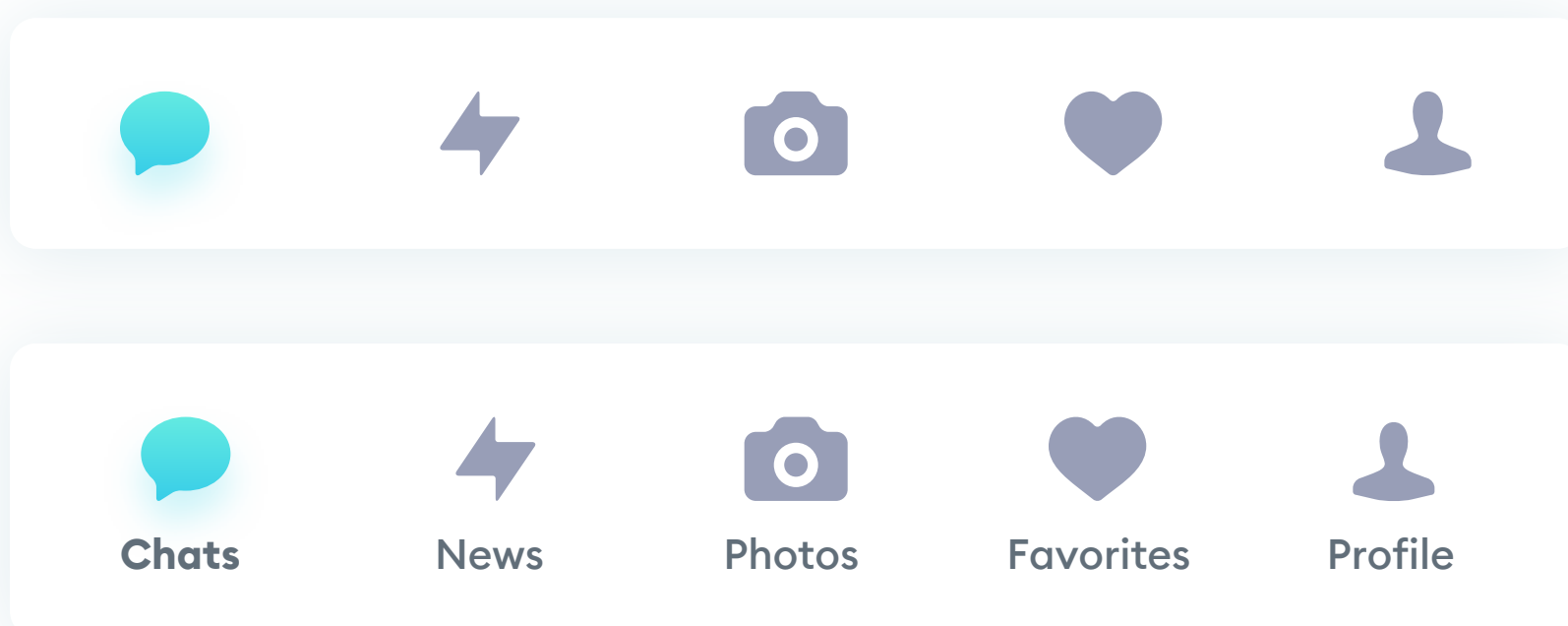


The difference between the regular tabs and the selected one needs to be very pronounced. If they're too similar, it can be very confusing for your users.

ICON ONLY TAB BARS

If you want to keep your app minimal, you can try going with the icons alone. The benefit here is gaining some vertical screen space (which on mobile is quite precious) and having a cleaner looking design.

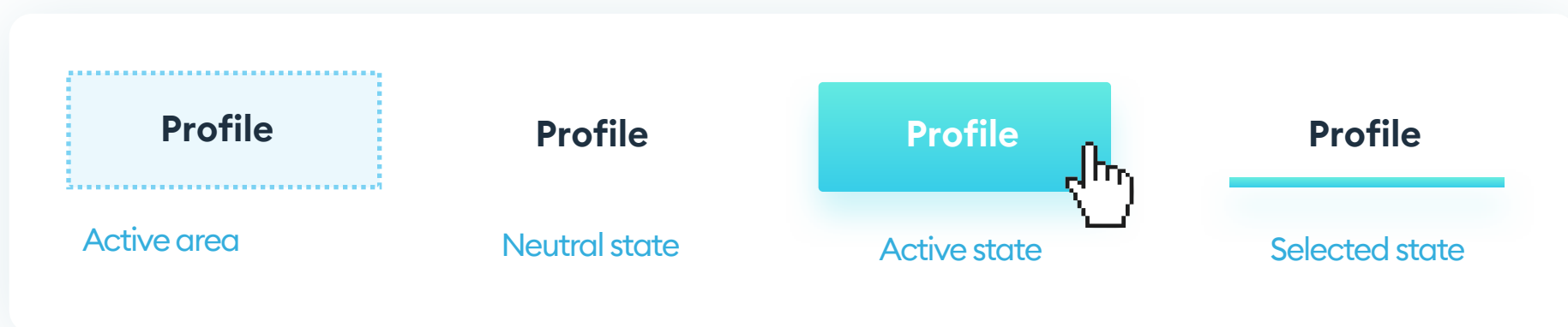
However, most users may feel lost at first, not understanding what the menu items are.



TEXT BASED TABS

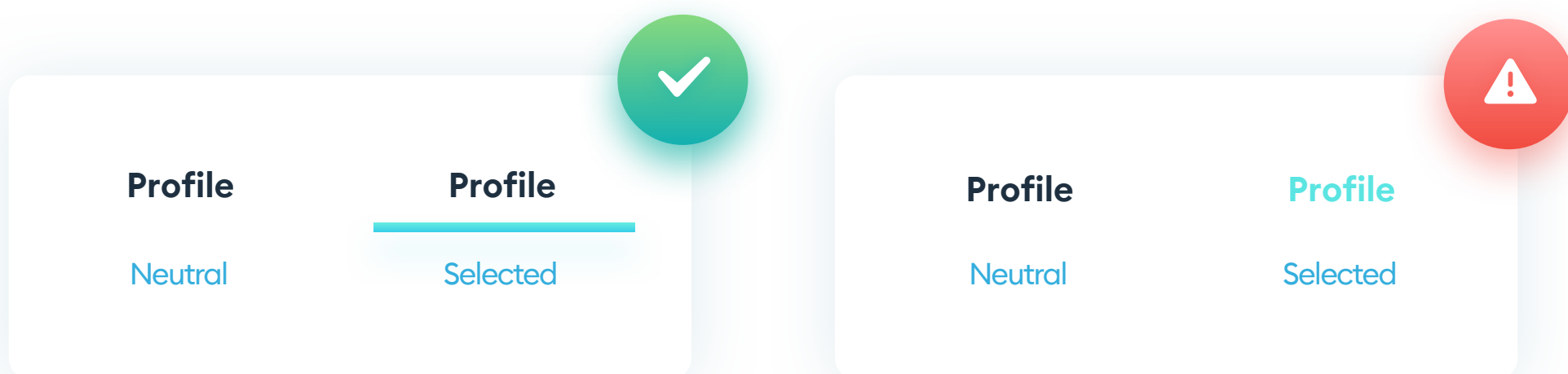
Text tabs are typically placed at the top of the screen. They're an easy way to switch categories, and when they don't fit the screen, you can scroll them left and right (in mobile apps).

Designing this kind of navigation pattern, start by creating a minimum height of 44p for mobile and 24-32p for desktop. Building each tab, treat it like it's an invisible button with a soft-grid applied. That means you need to create a safe (clickable) area, a selected, and an active state.



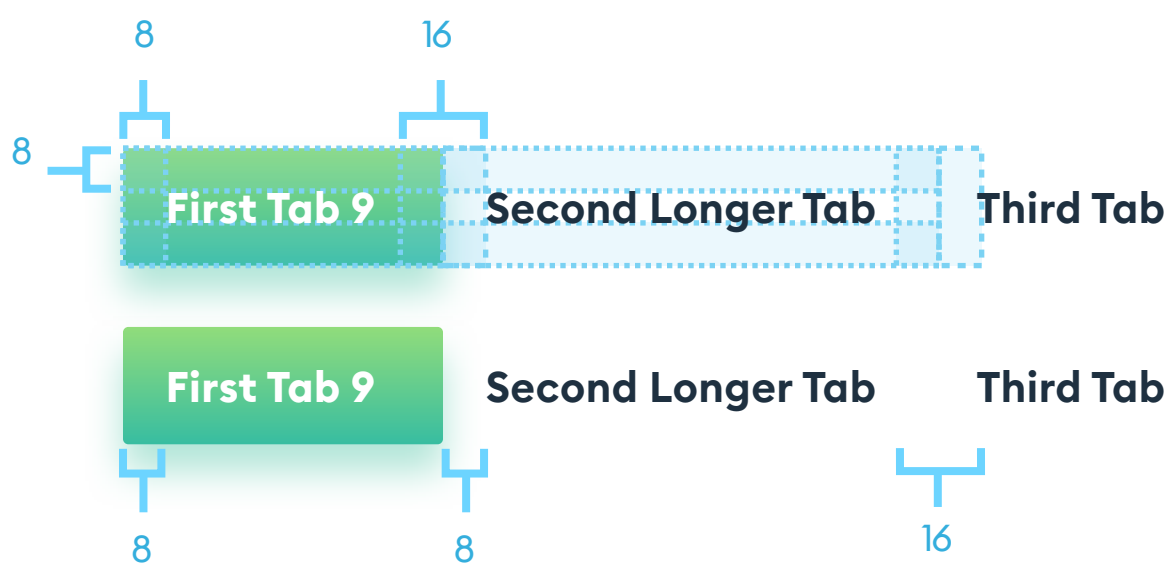
Even though the tab itself is just a line of text, your clickable area should always be bigger and include enough space on each side of your link.

While designing all the possible states, keep in mind to differentiate the active, neutral, and selected states by more than just a different text color. Without relying on color alone, your tabs will be accessible to all of your users. You can use underline or a filled background for best results.



TEXT BASED TABS ON LARGER SCREENS

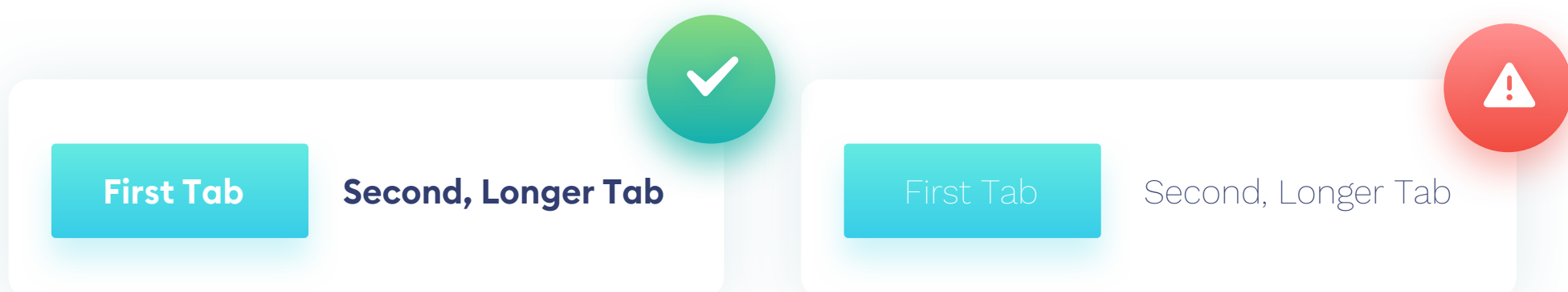
Most desktop-based tab bars can use a soft-grid to align tabs of various lengths. You can start by applying your grid base number (8 or 10) on each side of your text link. It means the space between two links will be double that number. Try to apply the same rule to the areas above and below the text. You can use temporary grid-boxes for easy tab alignment.



With an 8-point grid, the spaces are 8 and 16. With a 10-point grid, they'd be 10 and 20.

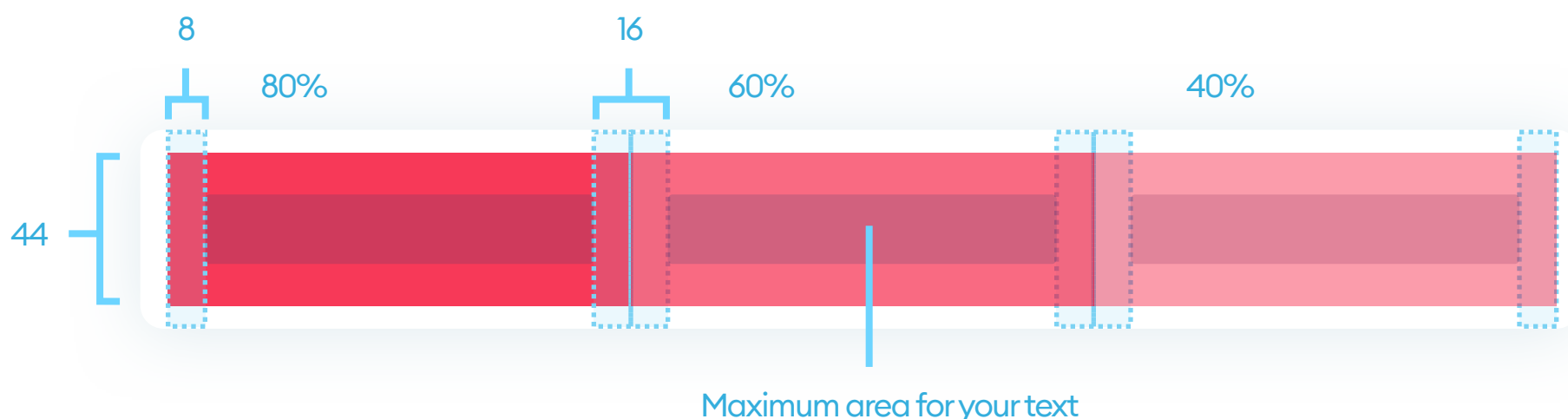
Avoid using very thin (light, ultra-light, or thin) fonts in your text tabs. Many displays still use lower pixel densities, so anti-aliasing these fonts can result in them being almost transparent, and thus barely visible.

Navigation is all about clarity and readability, so start with font styles of regular and thicker.



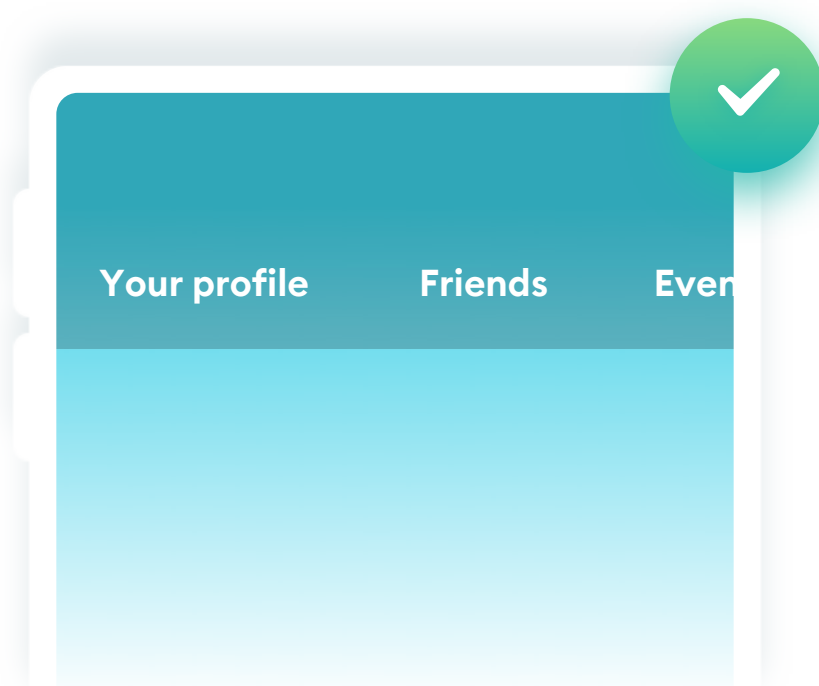
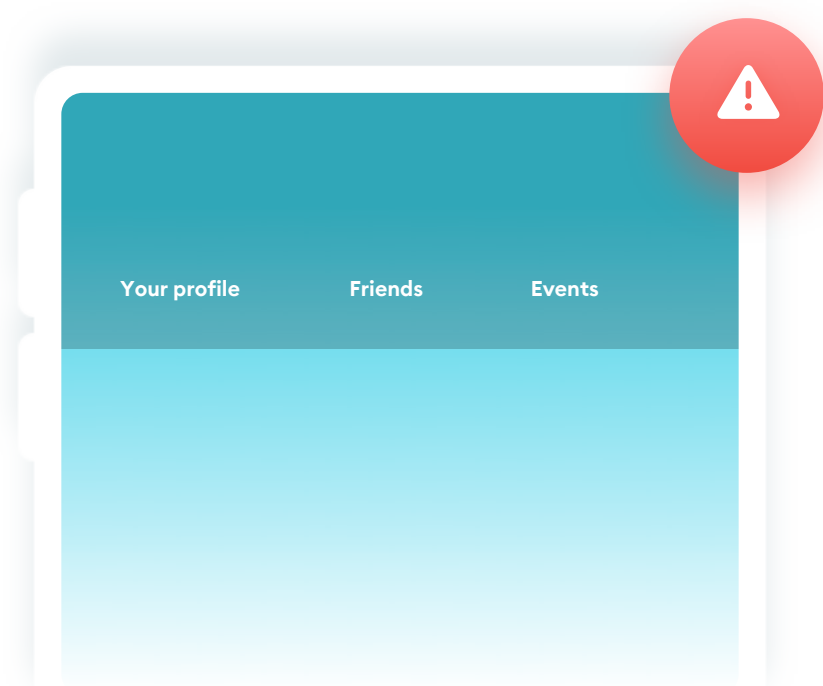
TEXT BASED TABS ON SMALL SCREENS

While designing for mobile, you can use a soft grid, or if you already know your tabs, you can divide the screen proportionally. Avoid having more than three text-tabs on one screen on a mobile device. If you have more, make them scrollable.



The easiest way to design mobile tabs is to create the same helper rectangles, and adding a grid-based internal margin inside them (padding).

If you need more than three tabs, make sure that the last one is slightly cut-off by the edge of the screen. That will serve as a suggestion to your user, that it's possible to scroll the tabs. Keep the font size consistent with the rest of your product and never make individual tabs use a smaller font (for example, ones that are longer).



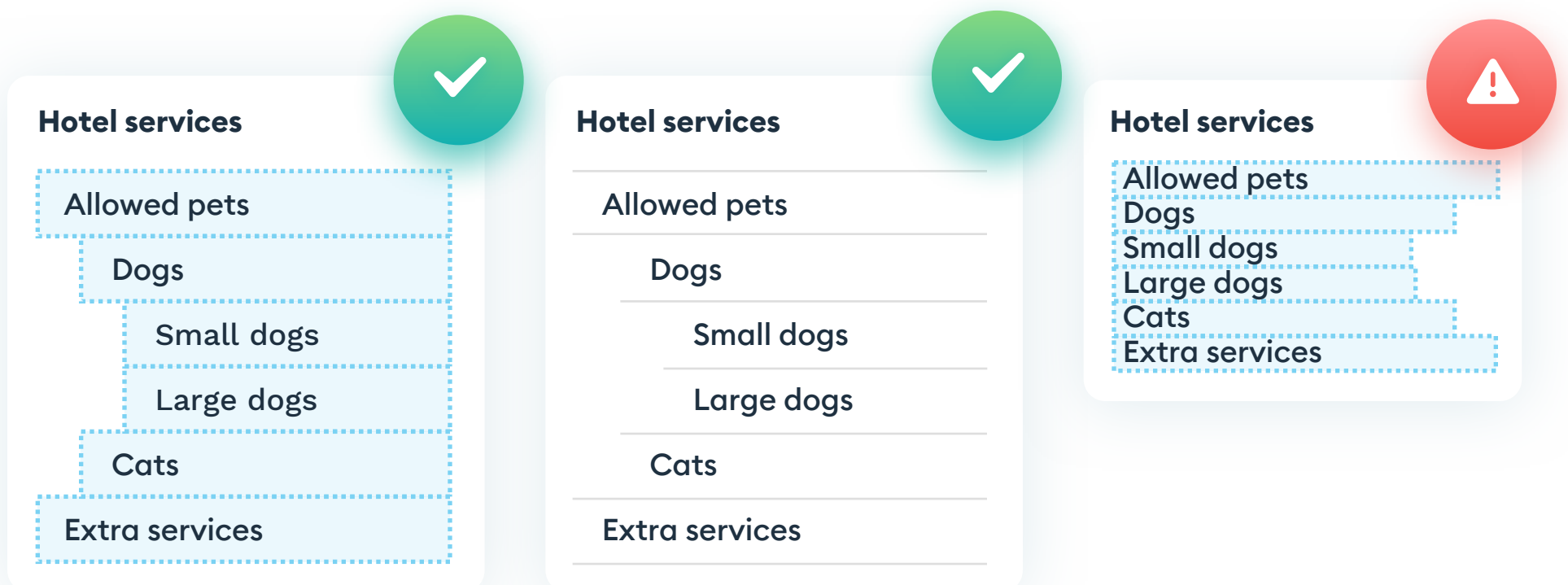
“

Designing a menu that's longer than your screen, always have the last tab partially visible from behind the screen. It suggests scrolling.”

SIDEBARS

You can use side navigation (sidebar) for longer tab lists. This pattern comes in handy if you have a lot of tabs, often with a nested structure. Try to avoid more than three levels of nesting for better readability.

When designing a sidebar, you can easily fall into the trap of trying to fit as many elements as possible on one screen (above the fold.) When your text is so densely stacked, it takes longer to scan and understand. Your users can also accidentally click on the wrong tab because they're so close together.



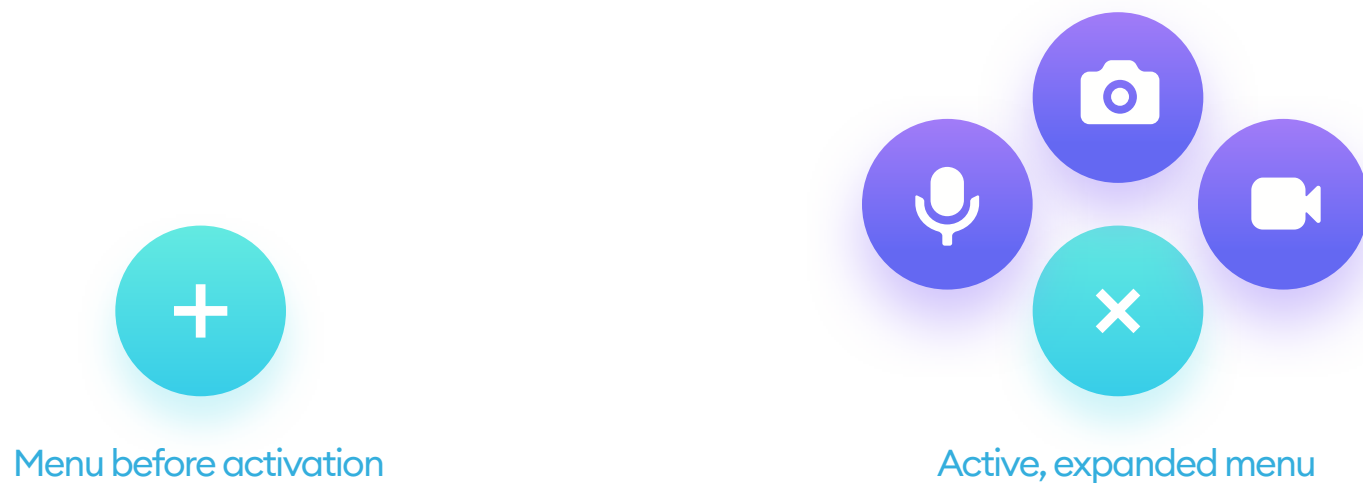
Just as with other tabs, create a helper-rectangle for each line of text. It will help you keep your spacing consistent and a scannable, clickable structure.

You can experiment with light divider lines between your tabs if you want to make the hierarchy more prominent. Otherwise, a big enough line-height will suffice.

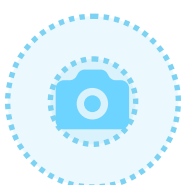
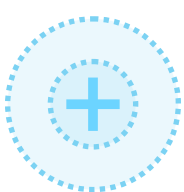
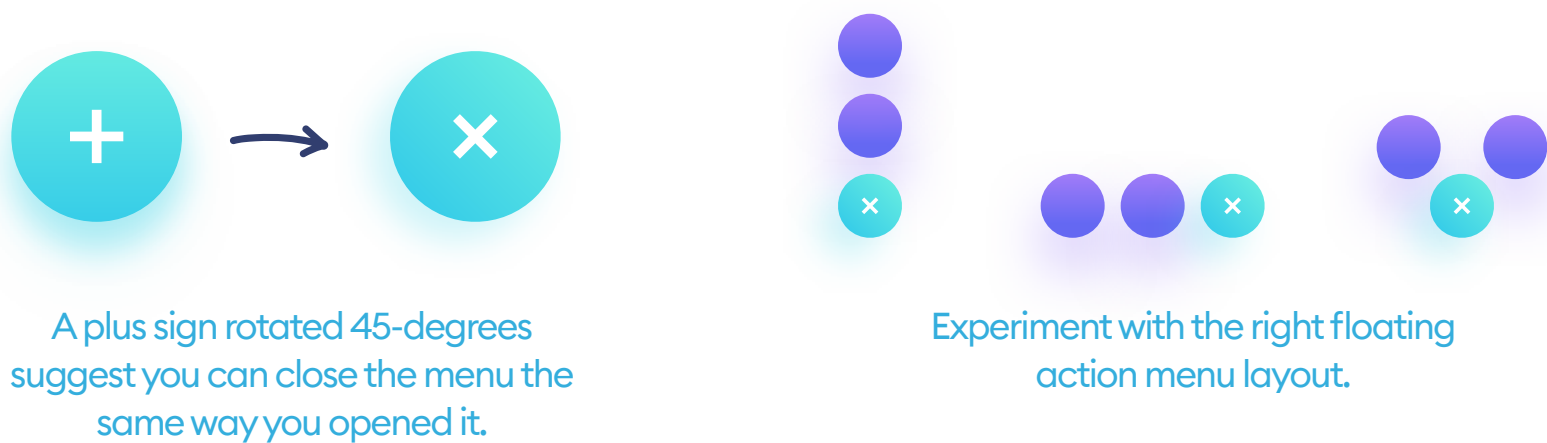
HIDDEN NAVIGATION - FLOATING ACTION

A floating action menu is a way to save space and provide quick actions available from anywhere in an app. Most often, it's a circular button in the bottom right corner of the screen that expands on touch.

This pattern came to life with Android Material Design, but it quickly became popular on iOS as well.



The most popular version of this menu is a circular button with a plus sign inside. When expanding, remember to rotate the plus sign 45 degrees to turn it into an X. That will quickly tell your user that touching it again will close the menu. The resulting fan of options can expand vertically, horizontally, or around the button.



Remember to set the safe space for your icons inside the circular button. The best way is to use the grid-base number (8 or 10) and create a guiding circle with that grid number margin on every side. Then fit the icon inside that smaller oval.

HIDDEN NAVIGATION - HAMBURGER MENU

The hamburger menu is a hidden navigation drawer that expands when the user pushes a three-line icon. It became popular with the first versions of Material Design and has since transitioned onto the web. In native mobile apps, the menu should always be on the left side, while responsive websites align it to the right.

Always design your hamburger menus with ease of use in mind. The shape and length of the lines are not that important. Just keep the general consistency of the stroke and roundness of the corners. You can pick any configuration you like, with either the same length lines or a variation in lengths.



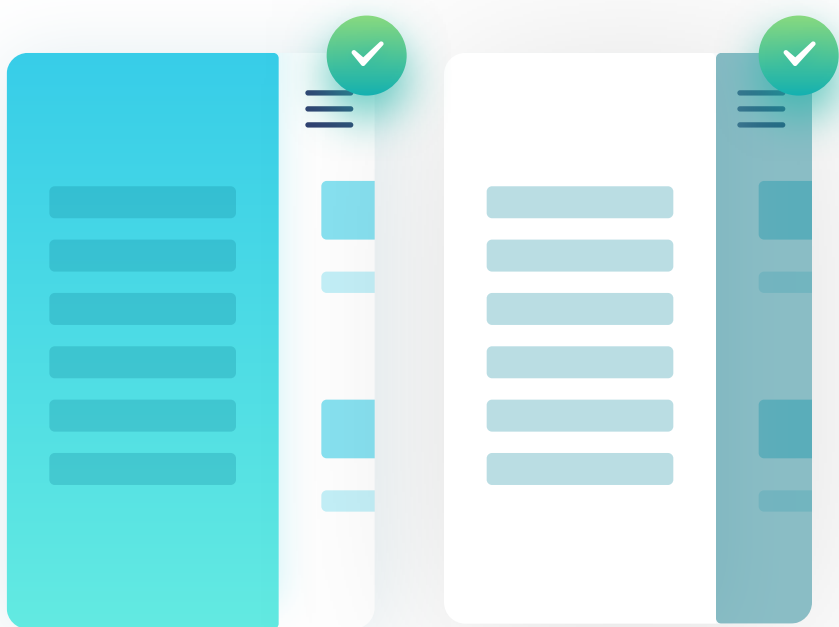
As with any active object, remember to have the clickable area large enough for comfortable touch. Between 40 and 44 points is an excellent place to start, with the actual line shape being half the size, or even smaller.

If your drawer has any tabs that receive notifications, you can add a notification bubble on the hamburger menu itself. That will help your user understand there is new content waiting in the drawer. While designing the notification bubble, remember the numbers can get pretty big and plan accordingly (like a pill shape for three-digit numbers.)

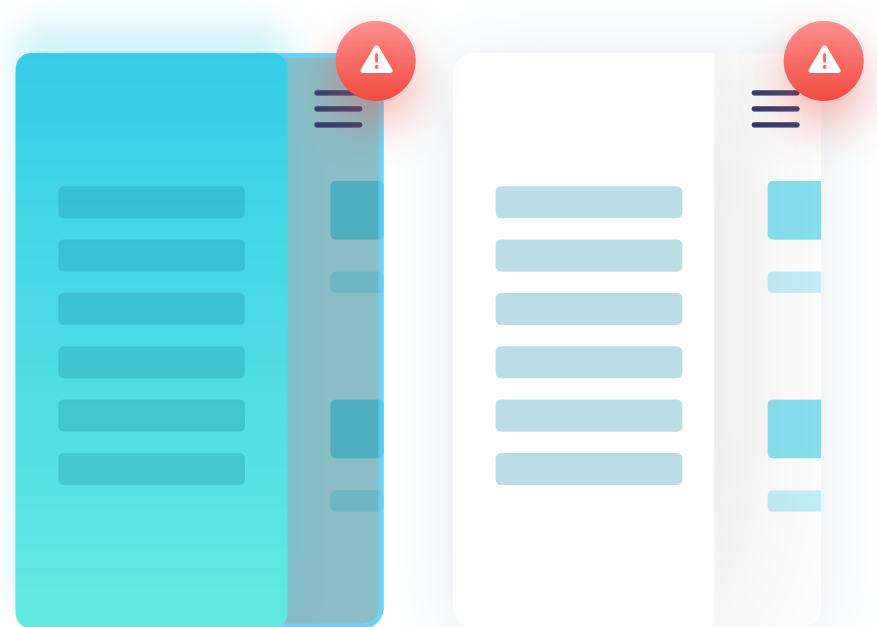


HAMBURGER MENU - THE DRAWER

A hamburger menu drawer is a vertical list of menu items - often with icons next to them. The minimum height of these elements should be the size of the touch target (44p). Try to differentiate it from the rest of the app, by using contrast so the user can instantly recognize where the drawer begins and ends. If you want to have the drawer in the same background color, as the rest of the UI, try darkening the UI underneath when the drawer is open.



Good contrast between the nav drawer and the rest of the interface.



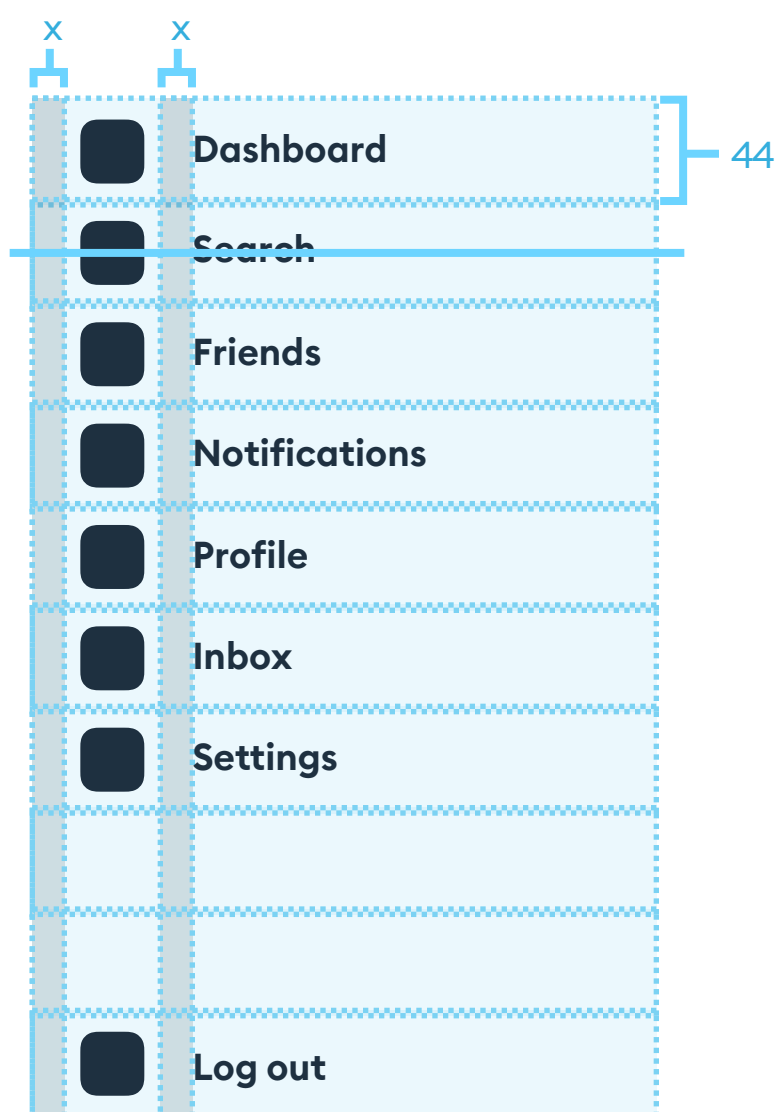
Contrast between the nav drawer and the UI is too low.

The drawer can either open over the content or push the content to the side. Both options here are ok, as long as they have consistent behavior with other overlays you're using in the product.



HAMBURGER MENU - LINK LAYOUT

One of the most common problems of navigation drawers is poor alignment and margins of icons and labels. You can quickly solve it by adding guiding rectangles with the right height (44p) and proper margins. Create a guide for every menu item right away and align your text in the guides vertically, so it's precisely in the center. Test it with some red squares above and below to be sure.



To align your icons, first set the right safe area for each one. Our choice is creating a square of 44 by 44 points and aligning the icons in the center. That square should have the same distance from the left side (screen edge) and the label. In the example shown here, it's the X value. That X comes from the grid base-number again (8 or 10.) Then align the labels to the left side and check back if they're vertically center in their safe areas.

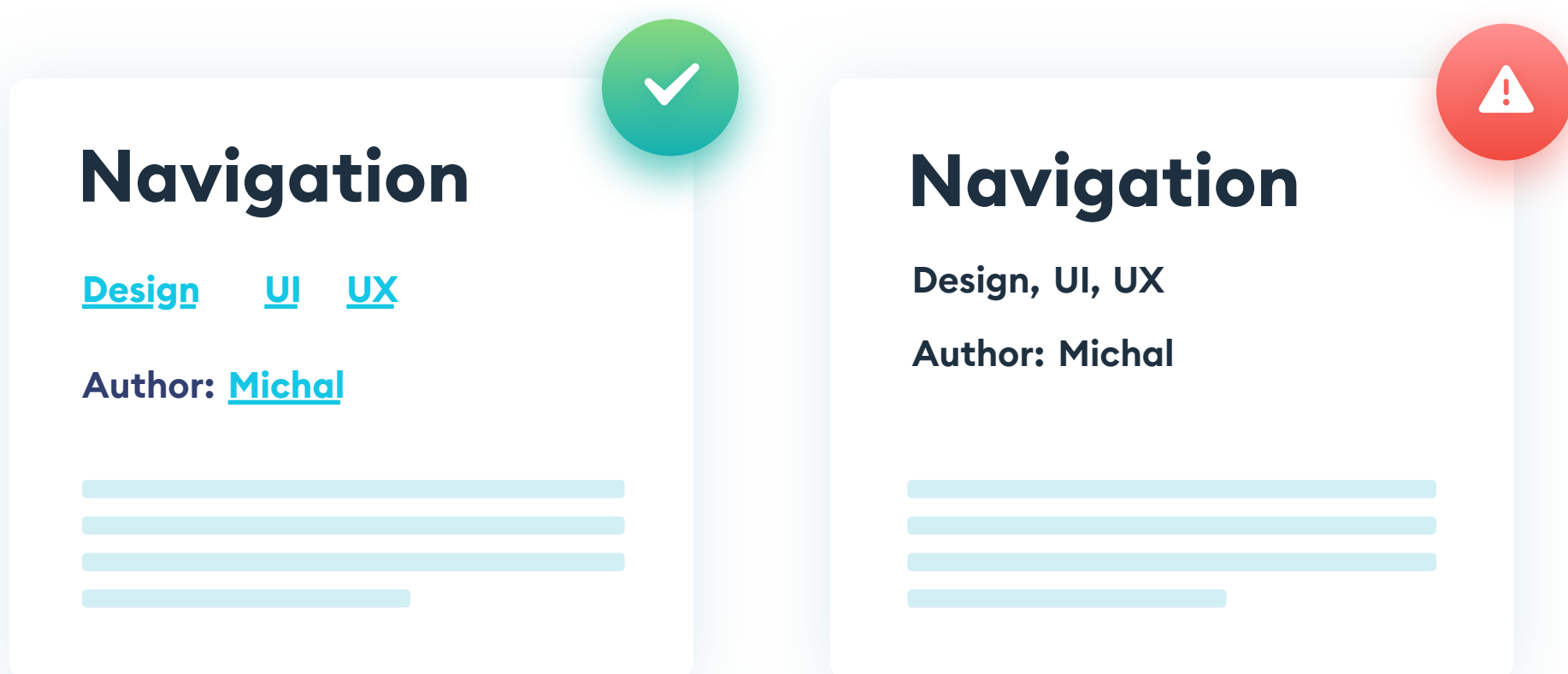
As the entire reason for a nav drawer is to have more than five icons (which we don't usually recommend), you need to pick them from one, consistent set. You can find out how to pick similar style icons inside the icon chapter.

If your nav drawer also has a log-out button, try to separate it from the rest of the menu by at least double-space, or simply bottom-align it. It will make it less stressful for users to try and tap other items, that are closer to the bottom of the navigation. Even with an additional confirmation for log-out, most people have a fear of accidentally triggering that action.

CONTEXTUAL NAVIGATION

Contextual navigation means all the links inside of your content or metadata. It can be a hashtag, category, user name, or a simplified information card that leads to the detail page. It can also appear within longer blocks of text.

Because the links are very close to other text-based content, you need to make sure they're different enough to stand out. Remember to highlight the links with a background, underline, color, or font thickness. The last step is making sure all links have the right active area around them, so there's enough space to use them on mobile devices comfortably.



An underline is a universally understood suggestion that the content we're seeing is a link. If underlining the links doesn't fit the rest of your design, consider a thicker font and a color change.

It's essential to avoid using color as the only link indicator. People with various types of color blindness can simply not see the difference and ignore the link. Experiment with font weights and other forms of highlighting.

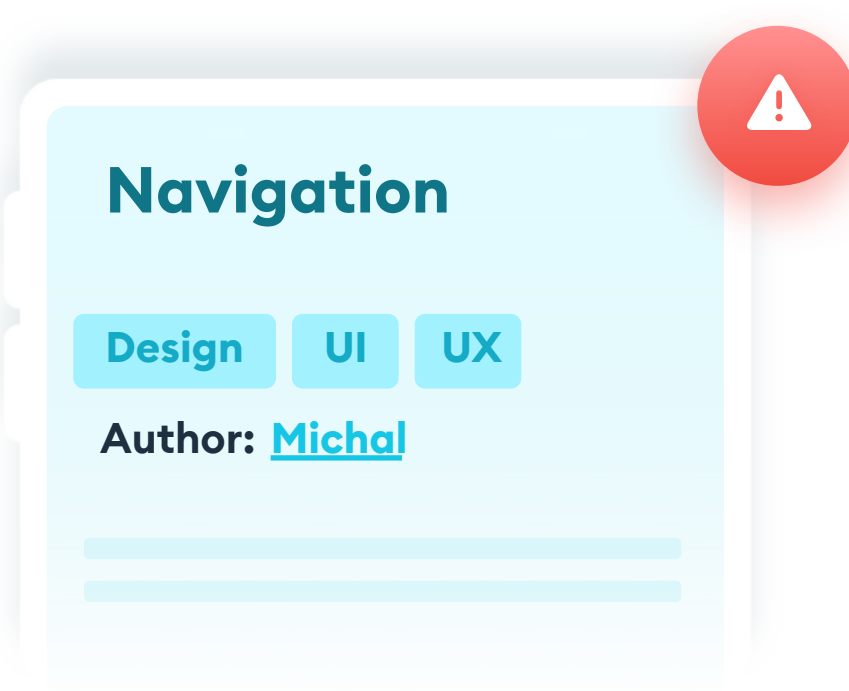
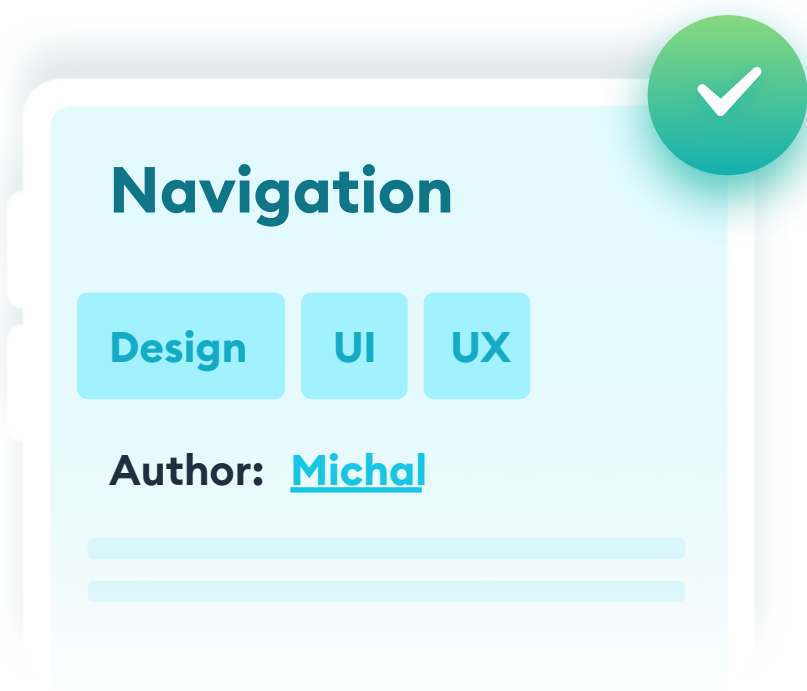
CONTEXTUAL NAVIGATION - GOOD IDEAS

Another way of designing contextual links is simply creating them as buttons. They can be filled or outlined, and often with slightly lower contrast than regular interface buttons. That will prevent them from capturing too much attention and distracting the users from the actual content.



Whether you're using a background or not, remember always to have the active area a bit larger than your button. On desktops, when hovering on them, they should change the state to indicate the fact that the user can click them. On mobile, just keep them large enough and with enough space between them to avoid unintended touches.

Contextual navigation needs to be large enough for comfort. Give enough whitespace to all of the page content, so that the structure and hierarchy give room for smooth navigating.



“

**The best navigation
is the one that's
always visible.”**



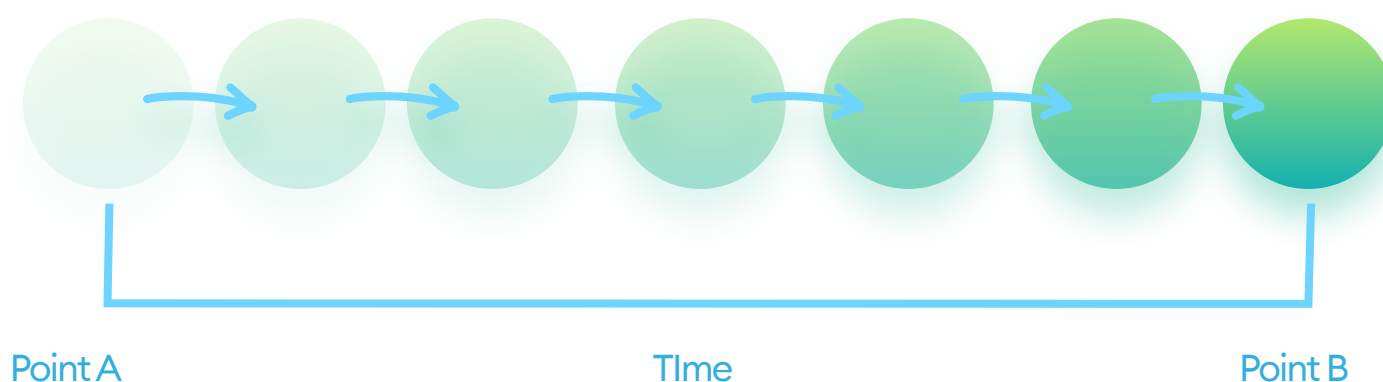
20.

Animation

Animation

THE BASICS

Writing about animation in a static book format can seem like a crazy idea. The subject of animation is an integral part of UI. We decided to cover all the basics in the following pages and let you experiment with motion on your own.



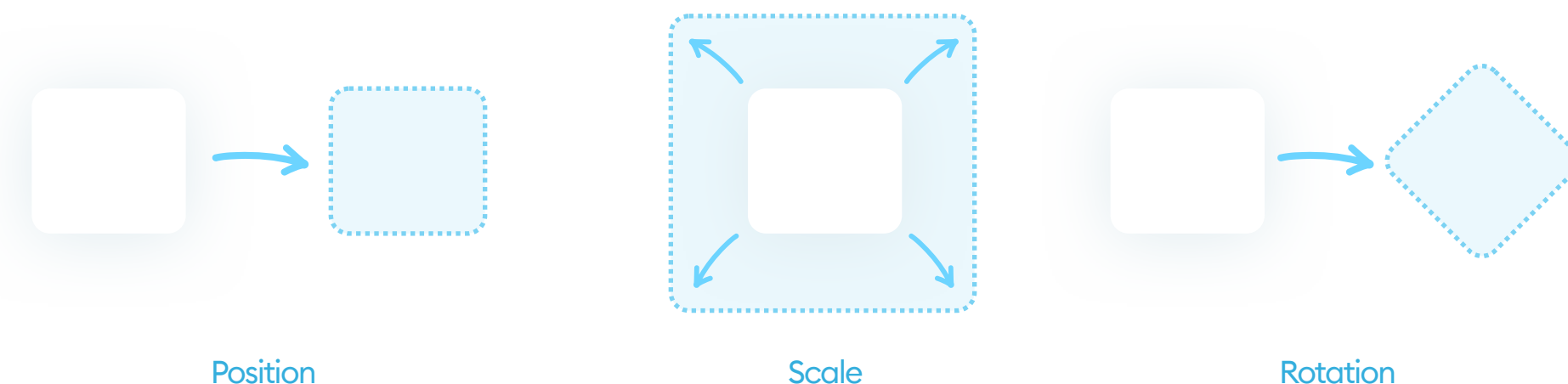
The most straightforward definition of animation is "change of state over time." State changes in animation can be its scale, position, shape, or rotation.

Other, more complex transitions (like 3d movements) usually require additional coding libraries and a lot of extra planning.

Animation in user interfaces can help with navigation, provide information (by giving context or showing the progress of a current task), or be a decoration with the only purpose to delight the user. Try to limit that last kind to a minimum. Having too many objects animated can make your interfaces harder to understand and frustrating.

TRANSITIONS

Transition types are a way to describe the change of our object in a given time. Three most common types of transitions include change of position, scale, or angle (rotation.) Other examples of transformation include change of shape, gradient, shadow, color, or transparency.



Most design tools that can export animation do it through simulating the change between the initial and final states. You don't need to create every single frame of the movement - only the first and the last one.

A movement (position change) is merely moving the UI element to the new position in the "final state." It's often achieved by duplicating the entire artboard and changing the position of individual items. The only thing left is to choose the delay, time, and easing. For example, you can define an object to move from $X=100$ to $X=200$ in 2 seconds.

We can add a delay value, which will delay our animation start with the selected number of seconds. That allows you to structure your animations, so they don't all happen at once.



EASING

Easing is a curve function modifying the speed of our animation.

The default choice (no-easing) is called linear, and it means that your object moves from A to B at the same pace.

Easing makes things look more natural because it speeds up or slows down the start (or end) of your animation within the same timeframe. The realistic feeling comes from the fact that almost no movement in the real world is entirely linear, so this gives your objects a more dynamic, natural feel.



LINEAR

At half-time of your animation, the object is exactly in the middle.



EASE-IN

At half-time of your animation the object is closer to the start and then it speeds up.



EASE-OUT

At half-time of your animation the object is closer to the end, and then slows down.

You can use both an ease-in and an ease-out curve inside one animation. When you choose a particular easing type, remember to use it consistently throughout your project.

If you want to play it safe, you can use linear animation, but for a better effect, try playing around with even a subtle easing curve.

You can modify the easing curve in various ways to achieve some unique, exciting effects, but you must understand the basics first.

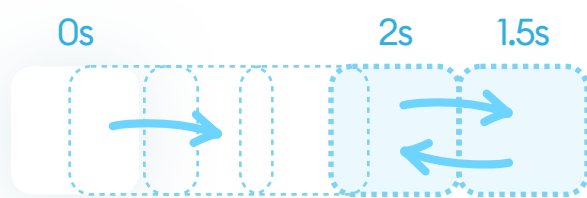
“

**Add Easing to your
animations for a more
natural, realistic
movement.”**

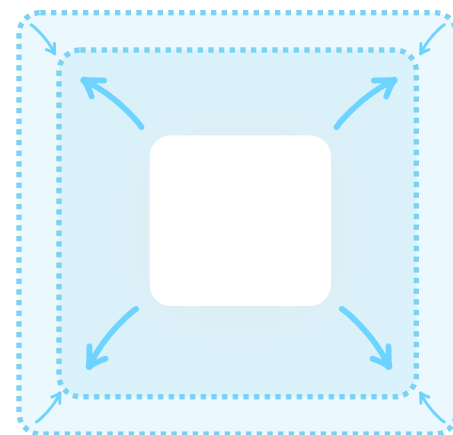
BOUNCE

The bounce effect became very popular with the release of the first iPhone, where it was present across the interface for that familiar "rubber-bending" effect. The most famous example is bouncing a list when you reach the top or the bottom to show there's nothing more to scroll. The effect proves to be very user-friendly, but you need to use it with caution - having an overly bouncy interface (especially with large bounce values) can look a bit grotesque.

The basic rule of the bounce effect is that our object exceeds the desired range at around 80% of the animation time and then comes back during the remaining 20% of the time. For example, if you're moving on the X-axis, from 100 to 200, you end up going beyond 200 (to about 220 or 240) and then going back to 200 in the last stage of your animation.



Change of position



Change of scale

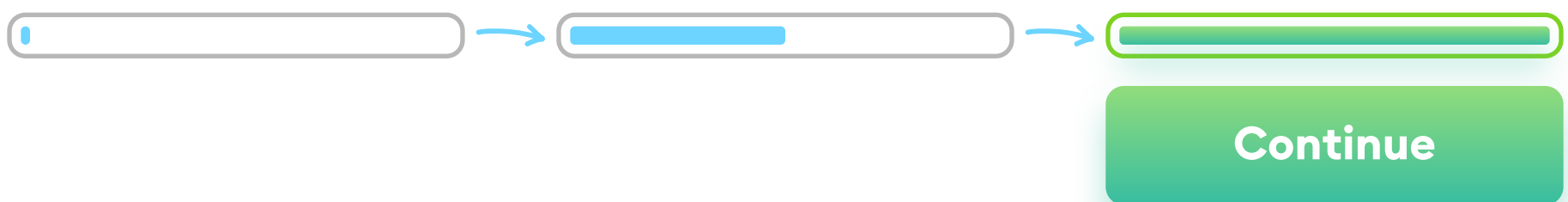
SOFT BOUNCE

Position bounce is the most popular one, but you can also use it for scaling or rotation. Try to remember that your "exceeding" value should never be larger than 20% of the final state. Values between 10 and 20% are optimal. Anything above that range will make it look unnatural and too jumpy.

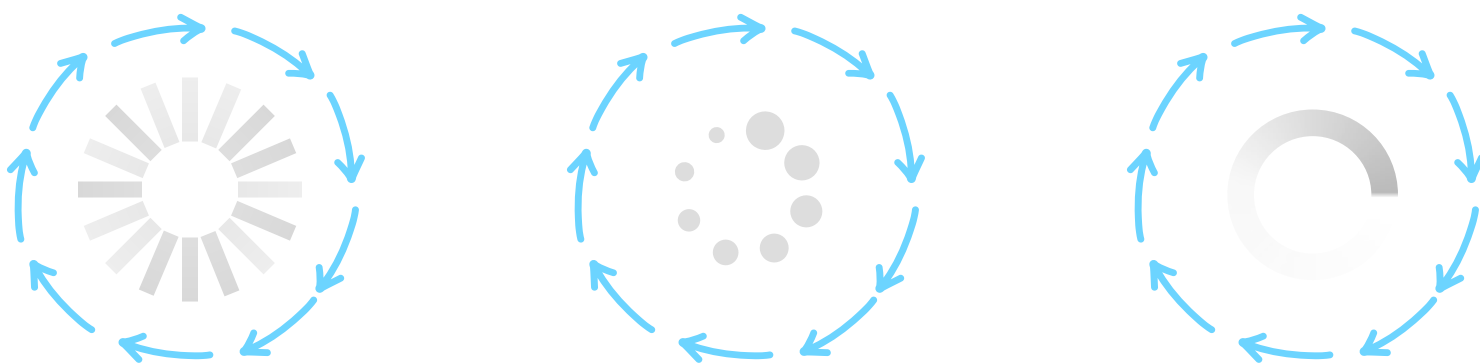
PROGRESS BARS

Progress bars are a popular way to show the progress of a process visually. Even a slow-moving progress bar suggests that something is happening in the background and makes it easier to wait without fear that something went wrong. They're often a rectangular shape that fills with color from left to right. That animation can happen linearly, or it can load the actual progress (so it can jump from 20% to 80% in one step).

While designing the components for an animated progress bar, remember to create it in three states. You'll need an empty progress bar, a half-way filled, and a full one. Sometimes it can be a good idea to create the fourth state signifying success.



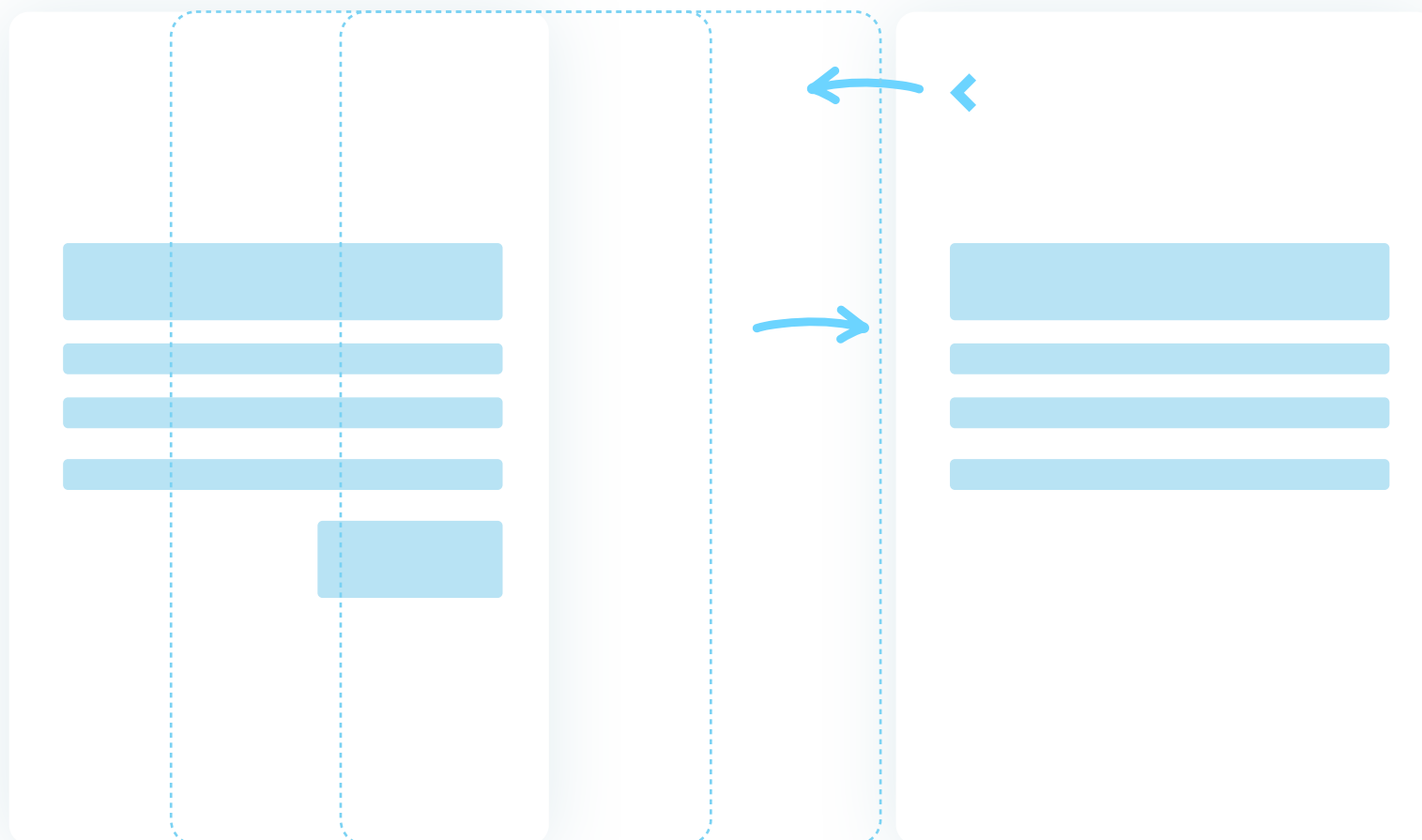
If your progress is interrupted, make sure to create a state for that as well. You can show successful progress by turning its color to green at 100%. If you want to be more creative, you can also change the shape of your full progress bar to become a button, allowing the user to continue that way.



When you can't predict how long a process can take, you can also use a spinner. It's a simple, circular shape that rotates around, showing the user that something is loading, albeit without a precise estimate of the time.

NAVIGATION

In mobile apps, you can use animation for screen transitions. Doing so, will make your interface more realistic, as it can imitate page-turning, and show the context of where each screen exists in space.



For example: if you're coming back to the previous screen and the back arrow points to the left, you can assume the previous screen will move in from the left side, moving the current screen to the right.

If you mess that logic up in any way, it will be much harder for your users to understand the relations between screens—design screen transitions in a way your user is expecting them to appear.

MICROINTERACTIONS

Microinteractions are a vital part of any interface. They help the user get feedback (good or bad) about his actions or the state of the application itself. You can use them to guide the users to achieve the desired outcome.

In simpler terms, it's a way to show (through animation) a change of state of an object. It usually happens through a conscious action on our part. Microinteractions help us understand that we've changed something much better than simply changing a state from 0 to 1.



An excellent example of a microinteraction is turning the toggle switch ON and OFF. Aside from the color change, the knob also moves, which gives us instant feedback that our action (switching the state) was a success.

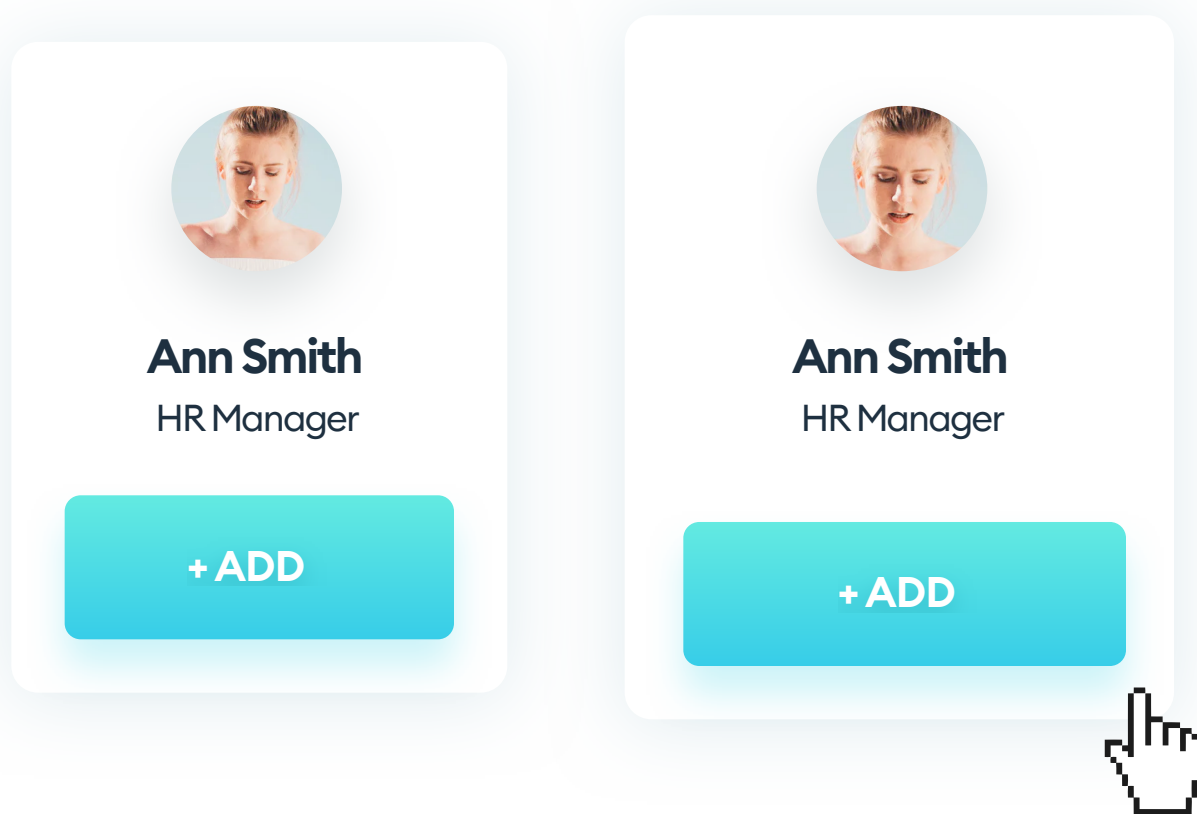
Another example is rotating the plus icon in a floating action button, so it changes into an X, showing the user can close it with another click.

There's plenty of room to experiment with designing your own unique microinteractions and delighting your users.

We believe microinteractions are one of the areas of UI, in which there's still a lot of innovation to be created.

MICROINTERACTIONS

Another popular microinteraction is Material Design based raising of a product card on touch or hover. You can achieve it by enlarging the object itself and changing its shadow to match. This action brings the element closer to our screen, communicating that we're currently interacting with it. You can also enlarge a photo within a mask on hover for a similar effect.



You can also turn a button into a progress bar! It's an excellent way for your user to understand what is going on without having too many elements on one screen. The possibilities here are vast.



The basic rule when designing a microinteraction is that they should only happen within the most important, actionable UI elements (buttons, cards, forms, or photos).

Avoid doing microinteractions on text or inactive objects, as doing so can confuse your users.

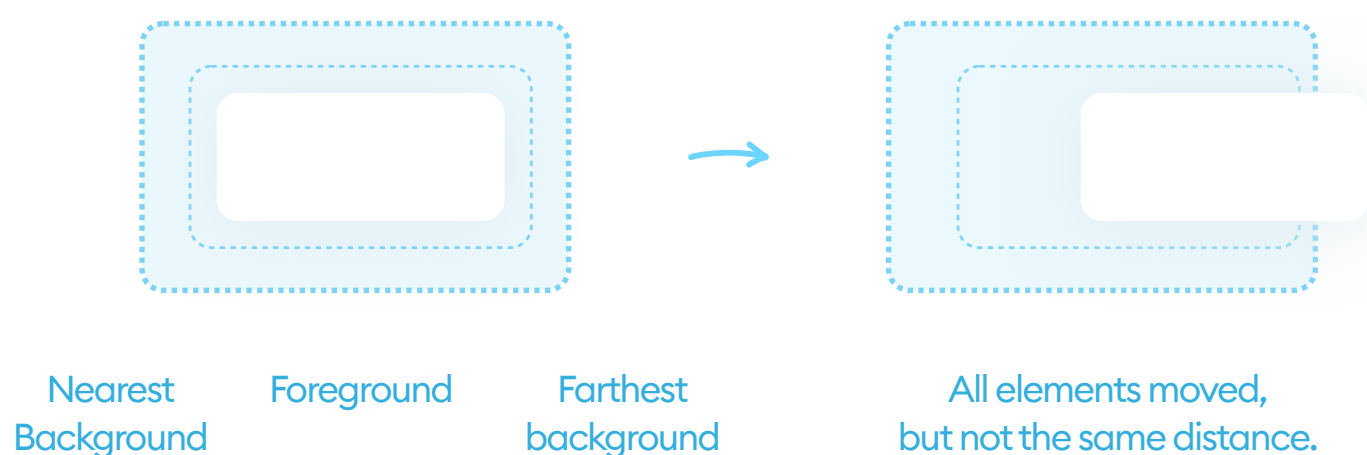
“

**Microinteraction
shows the change
of state of an
object by user
action. ”**

PARALLAX-SCROLLING

Parallax-scrolling is an effect in which the background moves more slowly than the foreground. It creates an illusion of depth in a two-dimensional scene. It initially came from video-games but has since become popular in UI design as well.

The entire concept is based on differences in displacement between layers. The farther away something is, the slower it moves. Use at least three layers to get the full visual benefit of this effect.



The most popular use-case of parallax scrolling is definitely in landing pages and mobile app onboarding processes. It can work with both horizontal and vertical movement on objects, but it's best not to do it on both axes.

As with most animations, you need to make sure it enhances the experience, instead of destroying it. Avoid using parallax scrolling on every element or every section of the website and keep a consistent speed for each layer.

Photos



otos P

21.

Photos

otos P

Photos

THE BASICS



EMILE SÉGUIN / Unsplash

Photos can have many functions in modern product design. They range from purely aesthetic ones (like "nice-looking" backgrounds) to profile pictures, photo galleries, or social media feed posts.

Luckily, the age of fake, obviously posed "stock" photos are over now. Websites like Pexels or Unsplash now offer entirely free, realistic, and high-quality images you can use in your designs. Most of these sites also integrate with popular design tools for seamless importing.

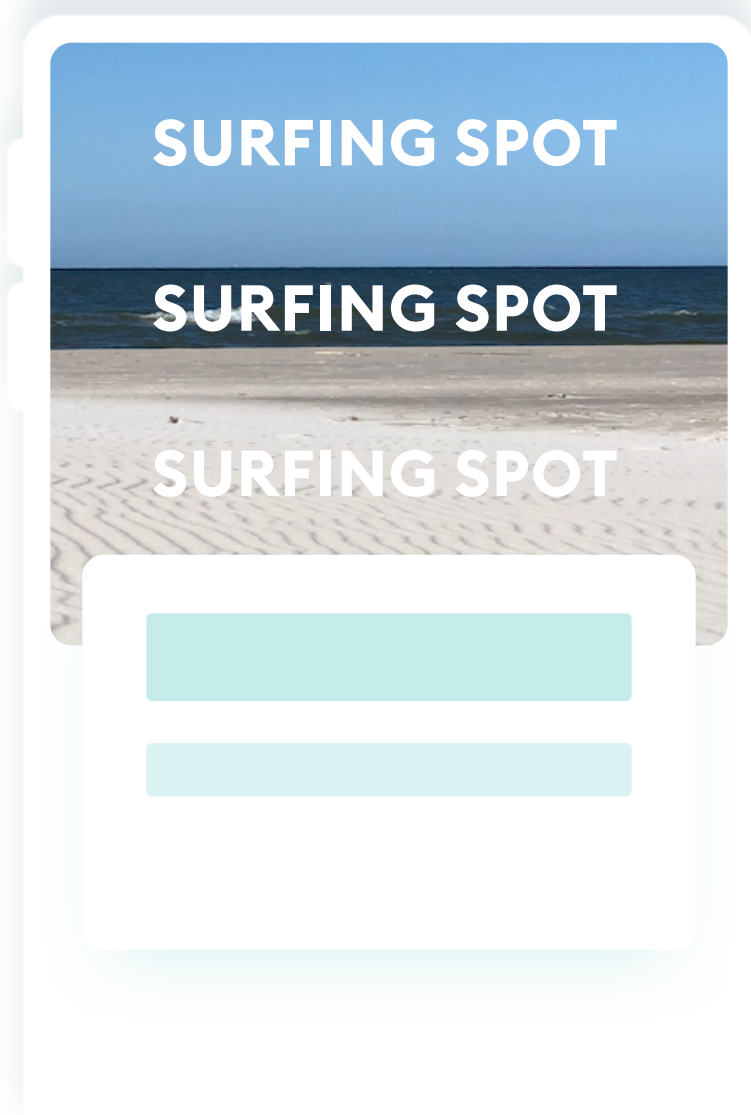
That's good. The overuse of the same stock photos was fairly common in the early 2000s. That same blonde girl that smiled at you in the shampoo commercial, the sausage industry, and on your bank website is now being replaced by less staged photographs.

BACKGROUND PHOTOS

A good photo can make your product screen feel more organic and alive. It can bring the machine-like quality of a minimal interface closer to something human. Background photos are a popular choice in both apps and websites, but many of them take the wrong approach, which makes the product unreadable and cluttered.

The main problem when using photos as backgrounds is a decreased readability of text (or other objects) placed on top of them. They're also often chosen randomly and don't match the overall color palette of the design.

If you plan on using the same static photo, you can adjust its relation to the text and make it readable.



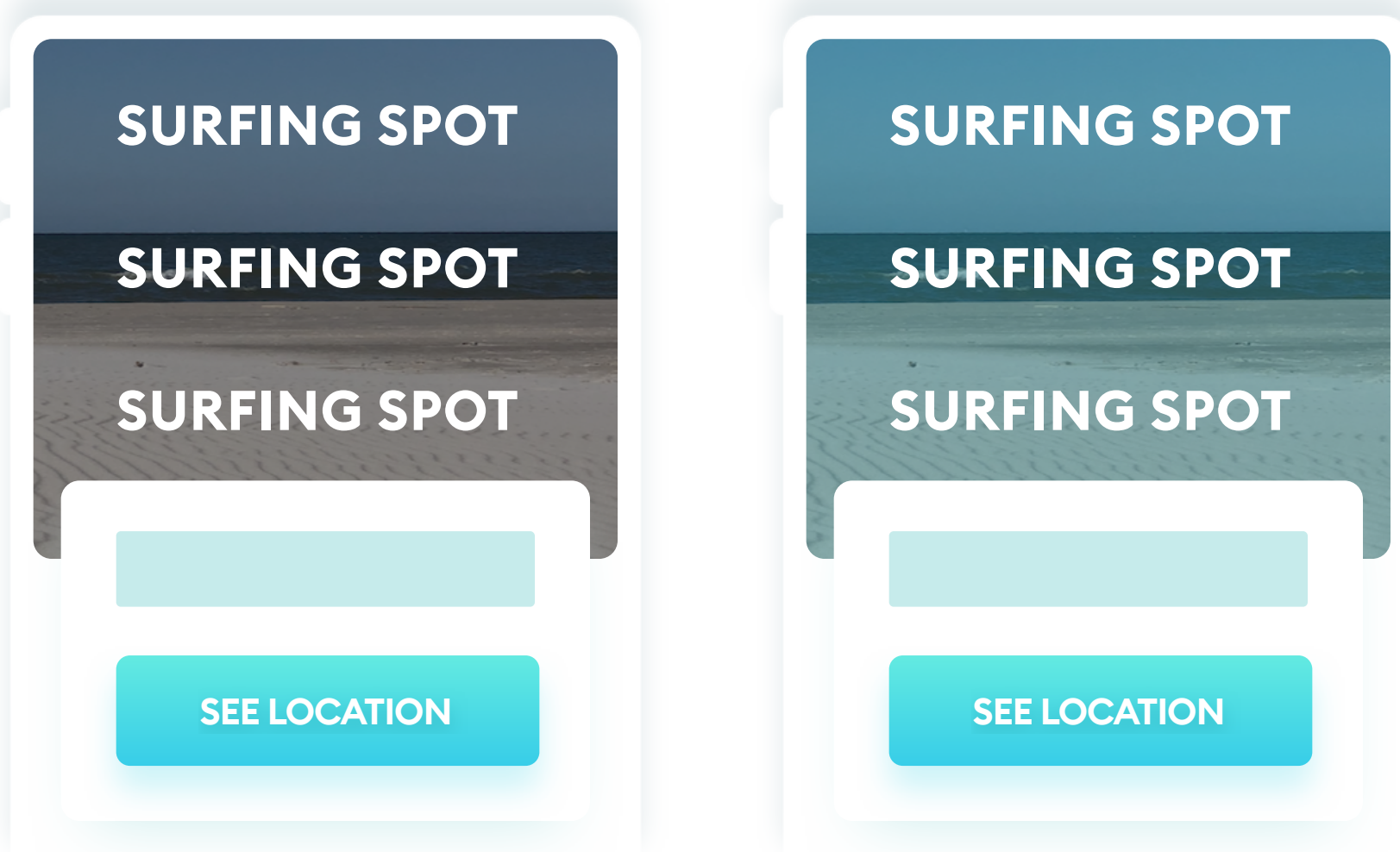
As you can see in the example to the left, the text has different readability based on its position on the photo. If you're using the same photo (on a landing page, for example), then you can modify it to always display your text on the readable portion.

If your photos are dynamic (user-generated), you won't be able to predict the best text placement.

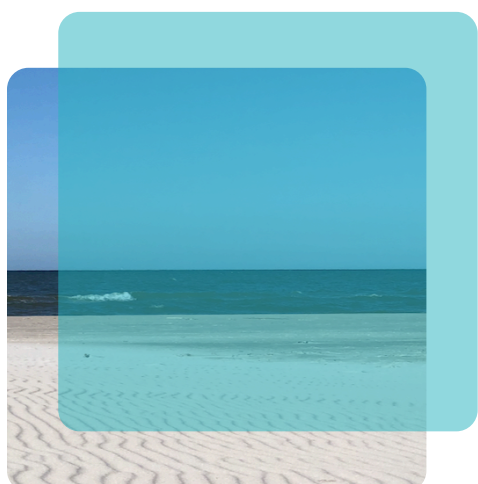
TEXT READABILITY ON BACKGROUND PHOTOS

You can solve the readability problem in several ways. One of the most common solutions is adding a color overlay to the photo to make it darker (or lighter). In most cases, black color at 50 to 80% opacity will make the text readable on nearly any photo.

If you want to make that interface fragment more consistent, consider using your primary color mixed with dark grey.



In the example on the right, the overlay has a hint of the primary color mixed in. That makes it easier to blend the photo with other UI elements like buttons and icons.



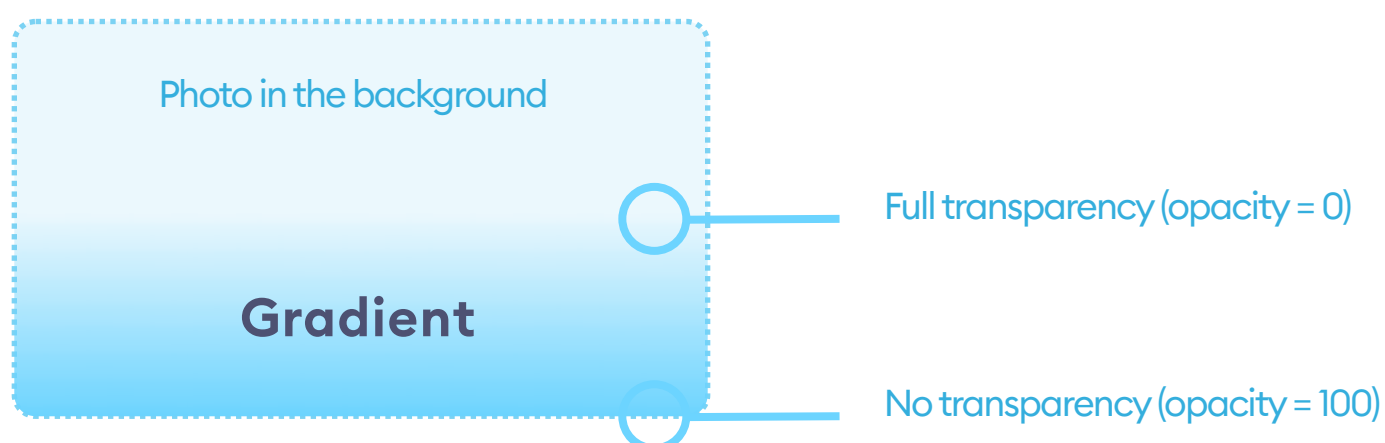
The entire process is merely adding a translucent layer filled with a darker shade of our primary color on top of the photo. Experiment with brightness and saturation to achieve the best effect.

GRADIENT OVERLAYS

If your content likely occupies only a portion of the photo background (top, bottom, or the sides), you can use a transparent gradient as an overlay. It's a trendy way to use gradients to bump up readability, but make sure your text actually IS readable.



If you're confident, your text will only be at the bottom of the photo, start by creating a gradient from 0 to 100 opacity as the top layer. Choose both colors from your primary colors, with a possible -30 brightness shift.

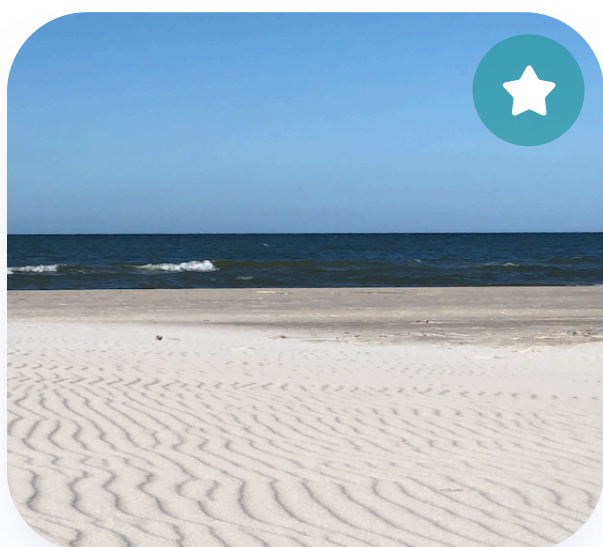


HARD OVERLAY

You can also add a more opaque rectangle under your text to be sure it's readable. It's challenging to pull it off in a way that looks good but is probably one of the best ways to achieve readability while placing text on photos. You can play with the transparency to make this work on any photo with excellent accessibility.



Those hard overlays can either be floating or fixed to the edge of your photo (any side). The fixed ones are often full width (or height if they're vertical) of the picture to avoid unnecessary lines.



If you plan to add an icon on your photo, start by adding it as a button, as it will be a lot safer to keep it readable at all times.

HOW TO PICK THE RIGHT PHOTOS?



Zaj. Alvaro Reyes / Unsplash - One of the most popular "UX photos"

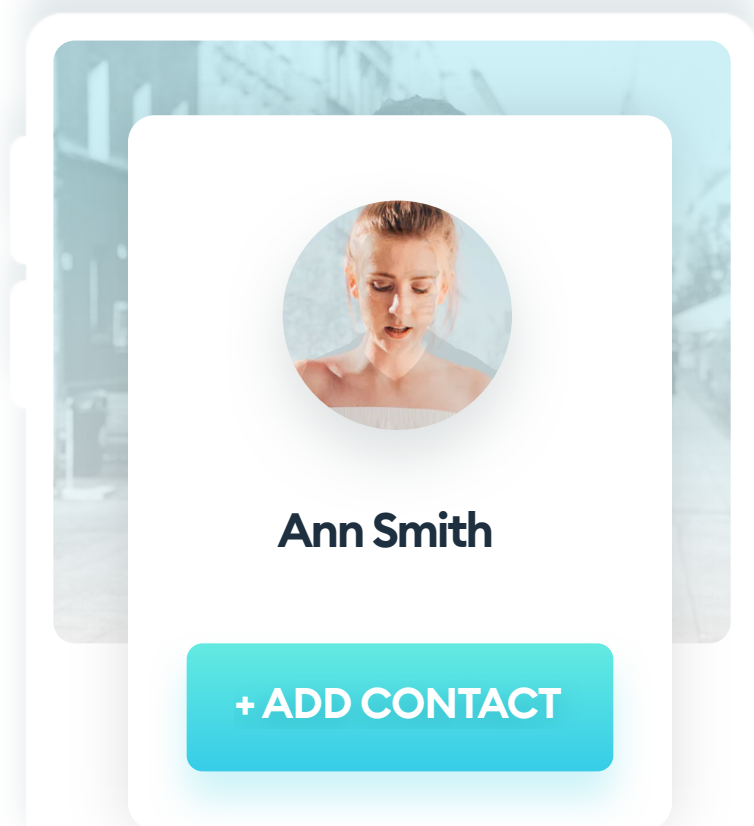
The general rule for photos is DIY. If you want to be unique and professional, take your own photos. With stock catalogs, you always risk using an image that is so popular that your project starts to look like everyone else's.

Try searching for UX on the leading free stock sites. Those same shots of a table with wireframes and devices have been in circulation for years. They've been used in articles (both good and bad ones), blog posts, portfolios, and more.

If you don't have an option to do the photos yourself, remember to make your stock selection fit the context of your product. It's not supposed just to be a "pretty picture" - it needs to add value and make sense with all the rest of the design.

FACES

When you're building a product that shows user lists and profiles, always use your own photo as the profile picture. All of the other users can be your friends and family. By going down that path, instead of choosing "free faces" from online portals, you make the product more emotionally connected to you. That allows you to better engage with what you design.



Are you designing a profile screen? Make sure it's your profile. Are you adding people to friend lists? Add your real friends in there! Don't use random, unknown people - make the project your own.

Many design tools now have plugins that allow us to randomly pick a face and place it within a selected shape. That's the easy way, and if you're in a hurry, you can use it. But if you want to build something special, we encourage you to use photos of yourself.

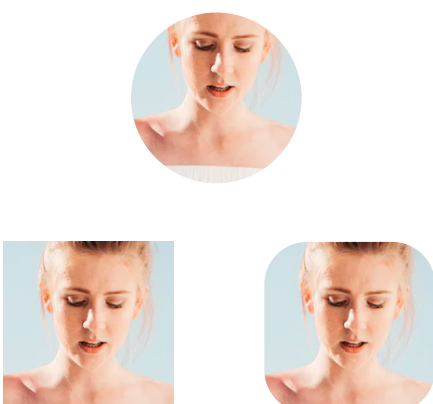
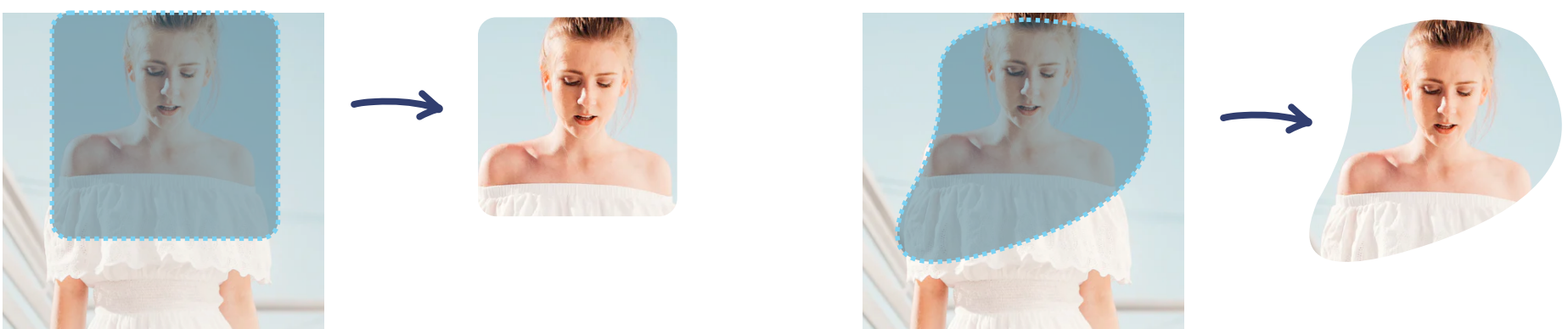
MASKING

Masking is a process of putting one object (in our case, a photo) in the confines of another object (the masking shape). Most pictures are (by nature) rectangular, so masking is the only way to have a photo inside a circle.

Recently, organic, uneven blobs are gaining popularity as well. These shapes can either be a mask for a photo, a color, or a gradient. They mostly work on landing pages, however, and are very rare in mobile apps (aside for maybe onboarding processes).

HOW TO CREATE A MASK

A mask consists of a masking shape and anything you want to fill its confines with. In most cases, it means the masked shape is a photo, but you can mask multiple shapes (even with the same mask) to achieve some interesting results.



Most common mask shapes are rectangles (sharp and rounded), squares (sharp and rounded) and ovals.

BASIC COLOR CORRECTION

Even if you're not planning to have any text on your photos, it may be a good idea to add a barely visible gradient on top of it. If your product is blue, you can add a very subtle blue overlay to the image. Even 2-3% opacity of the overlay will be enough to make it consistent without modifying the original photo too much. You can also do the same trick with maps to make them better fit your interface.

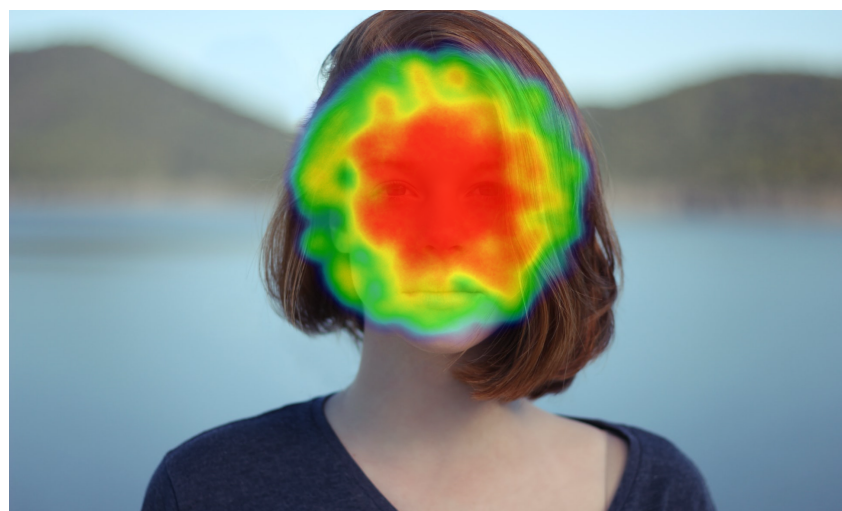


Do it only with images that you're adding to the product yourself.

Avoid modifying what the users upload! But if you're producing your content, having a consistent "filter" on all your cover images will make it feel a tiny bit more professional.

One great example is having a photo in the background, with the person in the picture looking to the side, at your Call to Action button. We naturally follow the gaze of other people we see, because we want to be sure there's no danger coming at us from that direction.

Eye-tracking studies show that if a person in a photo is looking directly at you, you'll focus on the face. If that same person is looking to the side, you will focus on what that person is looking AT.



Avoid using photos of people looking directly at the camera in your marketing materials (landing pages, onboarding processes, etc.)

Use pictures where people look to the side and place your buttons accordingly.

“

**Photos can add
an emotional value
to your designs,
but can also distract
your users.”**



22.

Illustration

Illustration

THE BASICS

Using illustrations is another way of making the interface unique. You can use them in place of photos as a visual representation of an idea. After all, they say a picture is worth a thousand words. Just as with photography, the best method is doing the illustrations yourself, instead of getting them from stock websites.



An illustration adds emotion to your content and makes the entire interface more friendly. It helps with remembering content and building a relationship between your brand and the user.

You can use illustration to create a mascot character that shows new users around your product. An excellent example of that approach is the popular language app - Duolingo. Another, less popular, but even more ubiquitous example is the paperclip icon from the Office suite of apps. If done right, an animated mascot character can significantly enhance the experience of your users and make them remember you better.

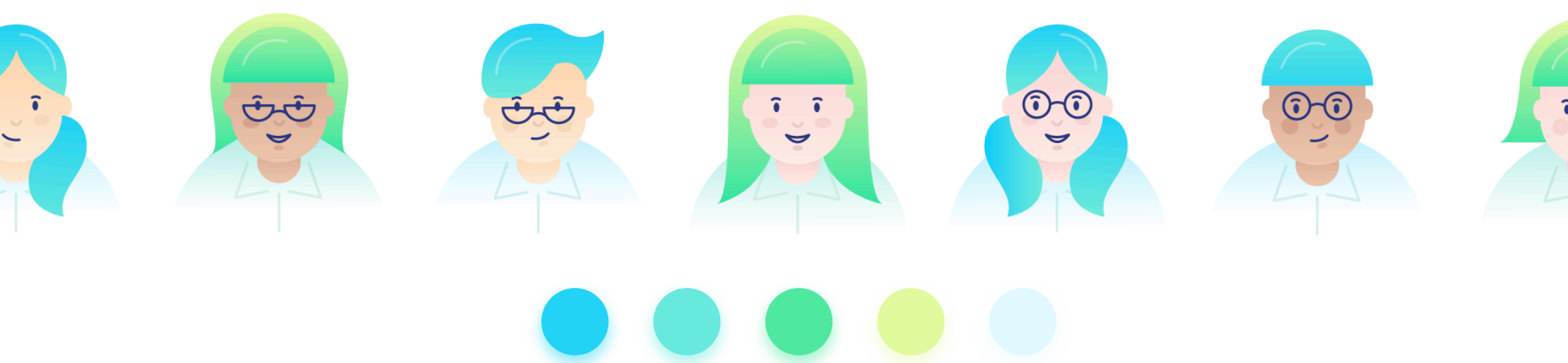
“

Illustration is a good way for the users to remember your brand.”

STYLE, COLOR, CONSISTENCY

Remember that the style of your illustrations must match the overall style of the brand. For example, a set of cute, vibrant colored characters won't be a good match for a fintech or a banking product. You can use them in an e-learning app to make the process more fun and engaging.

Just as with choosing and adjusting photos - make sure your illustrations all have a consistent style and color palette. It's great if the palette comes directly from your brand, as it will make the association a lot stronger.



When creating user avatars for an app, remember to keep a consistent color palette. It's great when it's a mix of your brand colors.

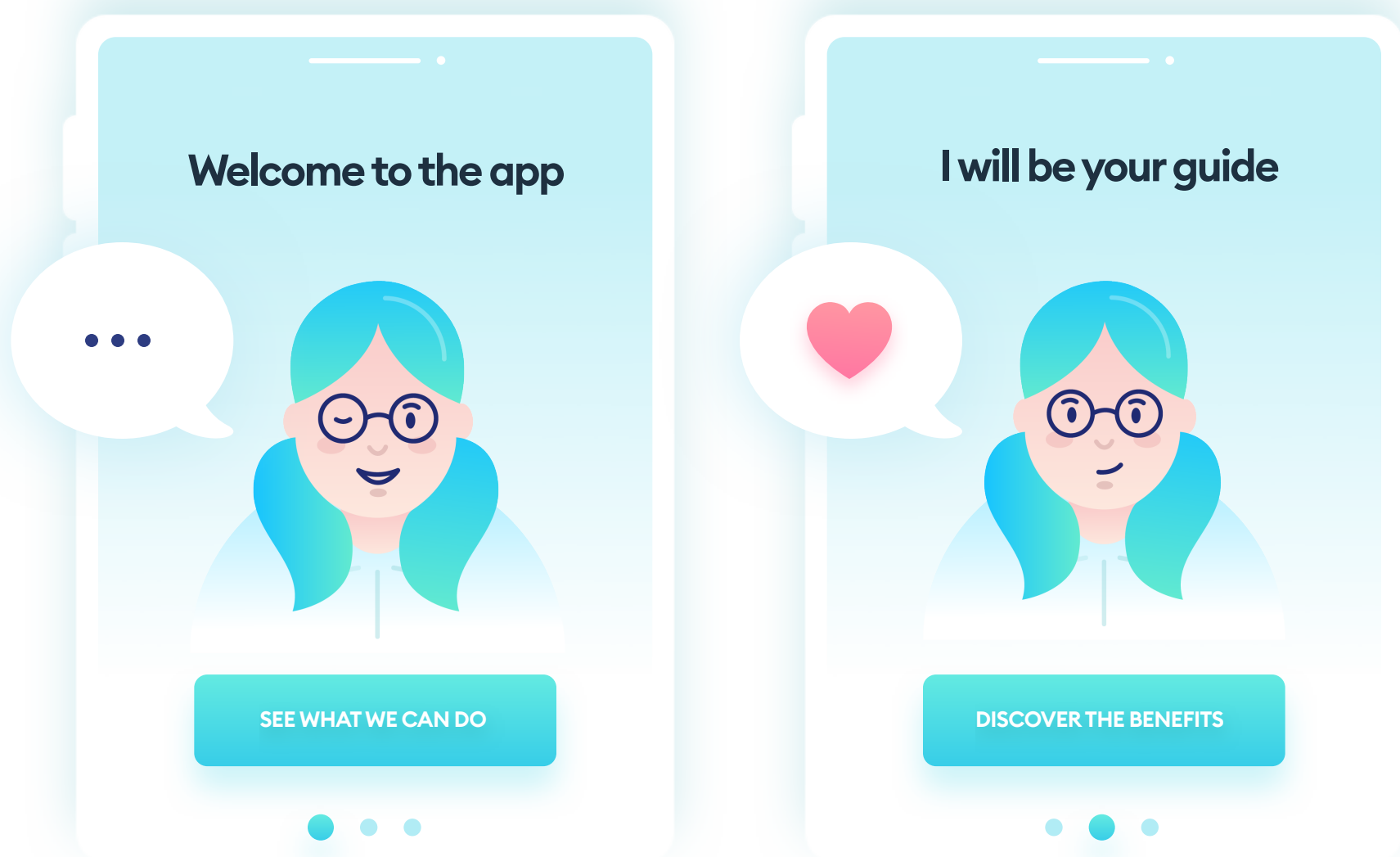
Style consistency means that you need to create each illustration the same way. Use the same stroke type, width, same fill type, and coloring. If you're using outlined images only, don't add one that's in full color as it will only confuse your users.

Keep the roundness of shapes and textures (if you plan to use them) consistent as well. It all has to fit perfectly together as a family of images.

ONBOARDING

A good place for illustration is in product onboarding processes. If you are planning to onboard your users using a couple of introductory steps, illustrating them will make the experience more fun and easier to remember. You can animate them (or use a parallax effect on a few layers) to make it look even better.

First impressions count, and being unique, funny, and engaging is a good place to start.

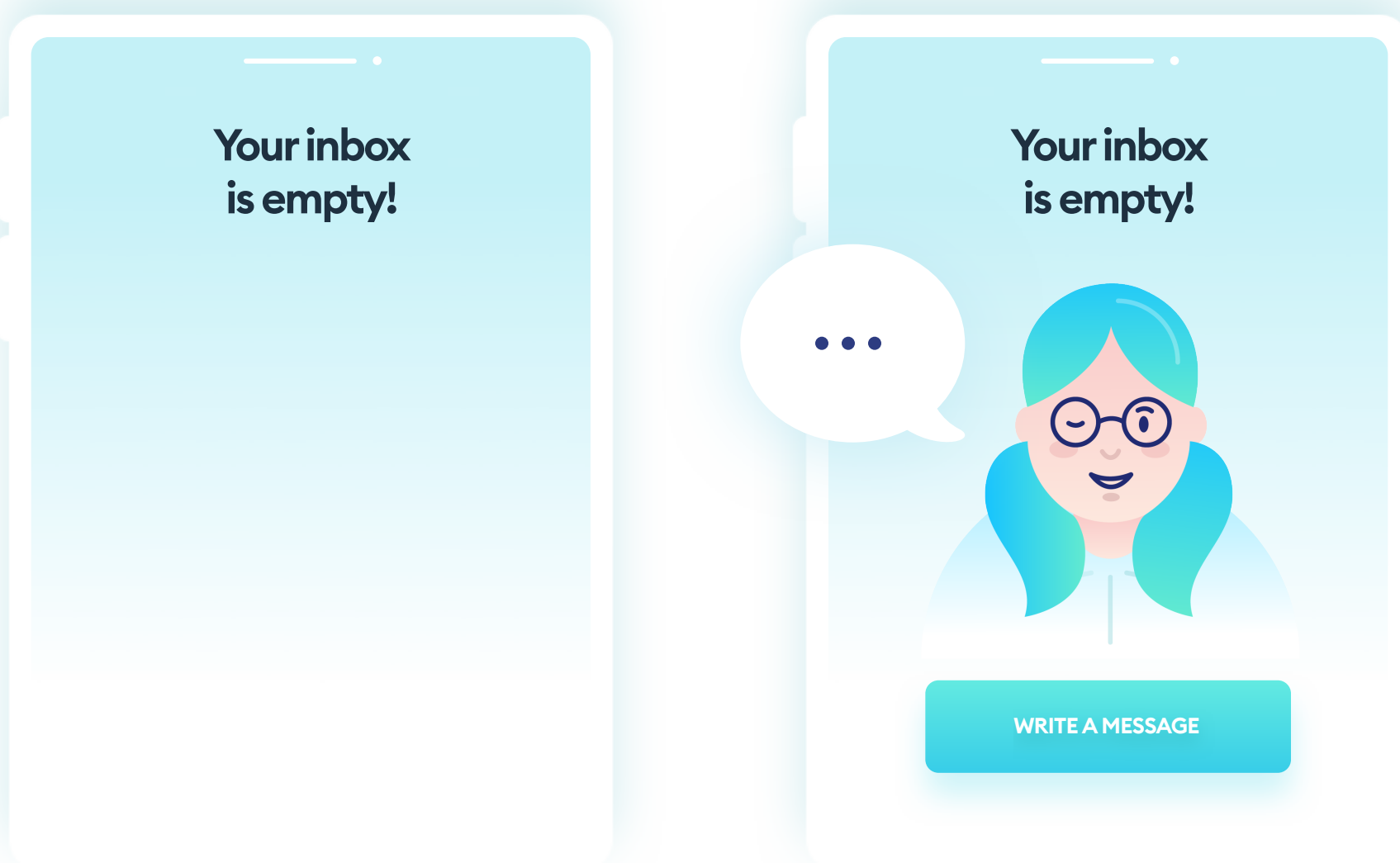


When designing an onboarding process with illustration, don't go too crazy with animating everything. It's best to keep the animations as highlights (like ringing a bell or the character winking). Subtle animations work a lot better!

EMPTY STATES

Empty states, also known as "Inbox zero," are versions of a screen that doesn't have any data yet. Imagine just setting up your email account (or being very strict while using it) and having no messages in the inbox. Instead of displaying a blank page, you can make the screen more alive by using an icon or illustration to explain why the inbox is empty.

You can also go a step further and add a call to action - in our example, for the user to write an email. As this state is a pretty rare sight for most users, you can make it a little more fun than the rest of the product.



Which one would you prefer to see when you open the app for the first time? Having an engaging empty-state design will delight your users and make them remember the product.



23.

Language

Language

THE BASICS

The language of our product, right after the right design, is defining how users will react to using it. We can be very serious, business-like, relaxed and casual or create our own, unique way.

A UI designer doesn't have to be a copywriter. Don't get us wrong here. But there's plenty of micro-copy to be written in user interfaces that often never get any copy treatment. Bad communication can kill a product, even if it has perfect UI.

All micro-copy should be simple, concise, and clear. Getting the messages short is always good, but keep in mind that the copy has to state its meaning precisely. Try to avoid broad labels like "OK" or "Next" on buttons. You can replace them with more action-specific ones like "Save changes" or "Add to favorites."



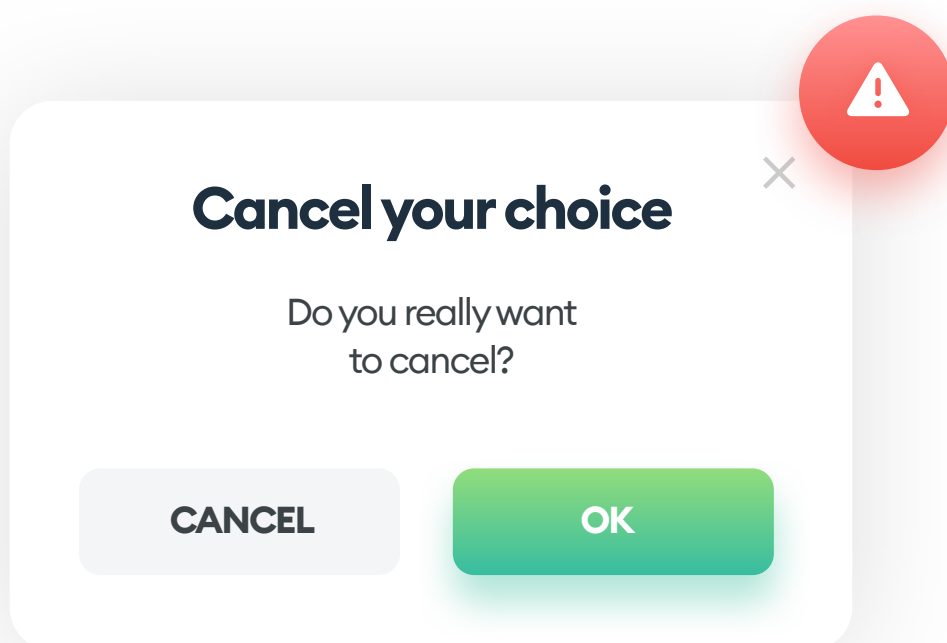
Save Workout!

I am AWESOME!

A popular fitness app replaced their "save workout" button with one saying, "I am awesome!". The friendly, personal touch made their users happy with a small dopamine boost every time they pushed that button. These non-standard ways of handling user communications can become a selling point of the product. Be creative, but without being too vague or losing clarity.

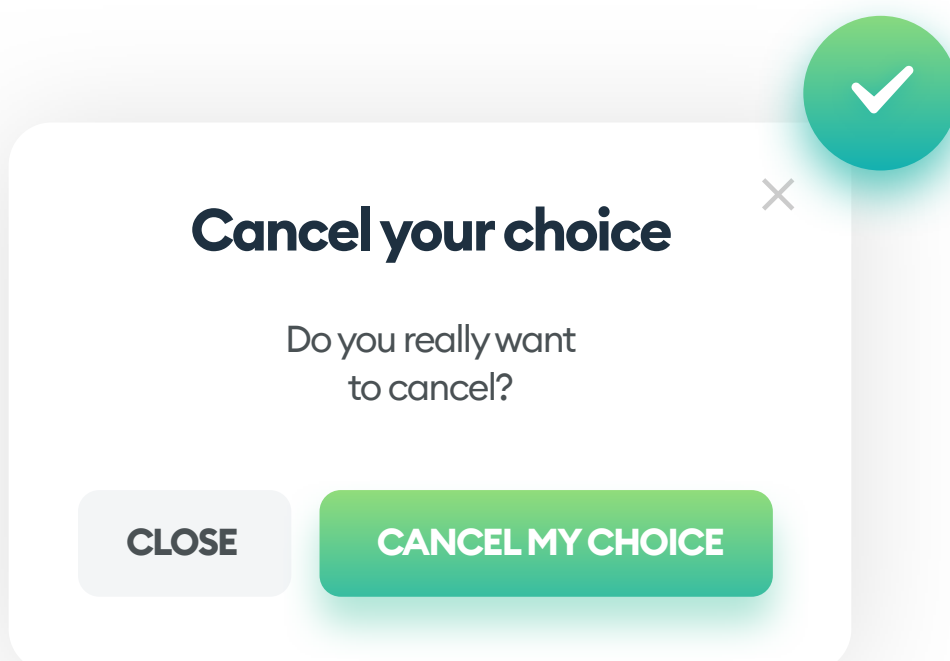
NEGATIVE ACTIONS

One of the most common mistakes is using double negatives or very complex word structures for negative actions.



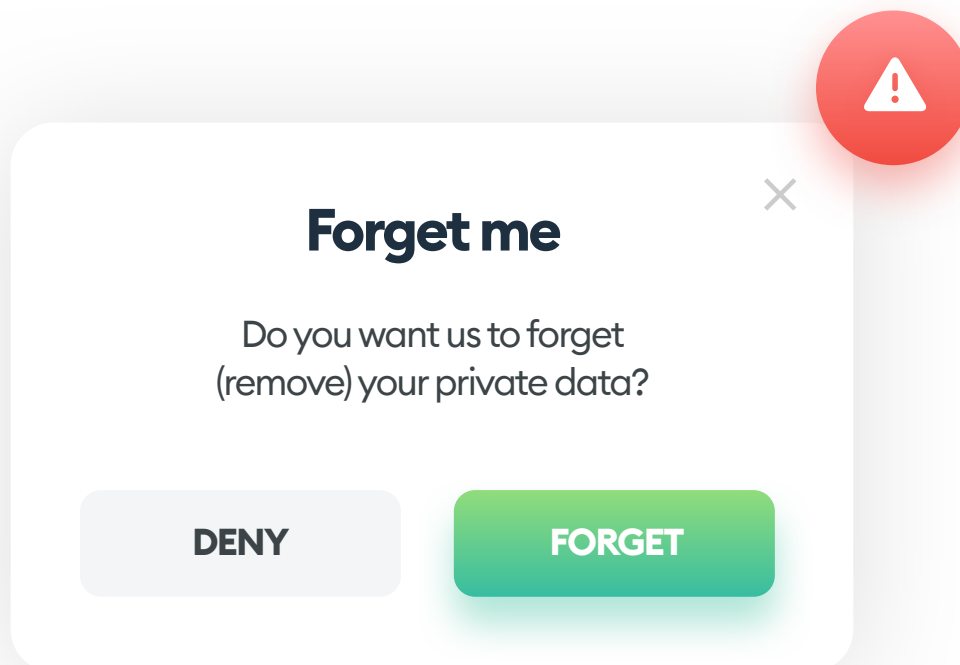
An excellent example of this issue is canceling a cancellation. After a while of looking at the popup, we get that we confirm the cancellation by pressing OK. But these kinds of messages can confuse and be quite annoying. Some users may not understand it at all and close the app or website without taking action.

When a user has to confirm a negative action (like removal or cancellation), it's best to be as precise as possible. Have your button precisely state what happens when you activate it.

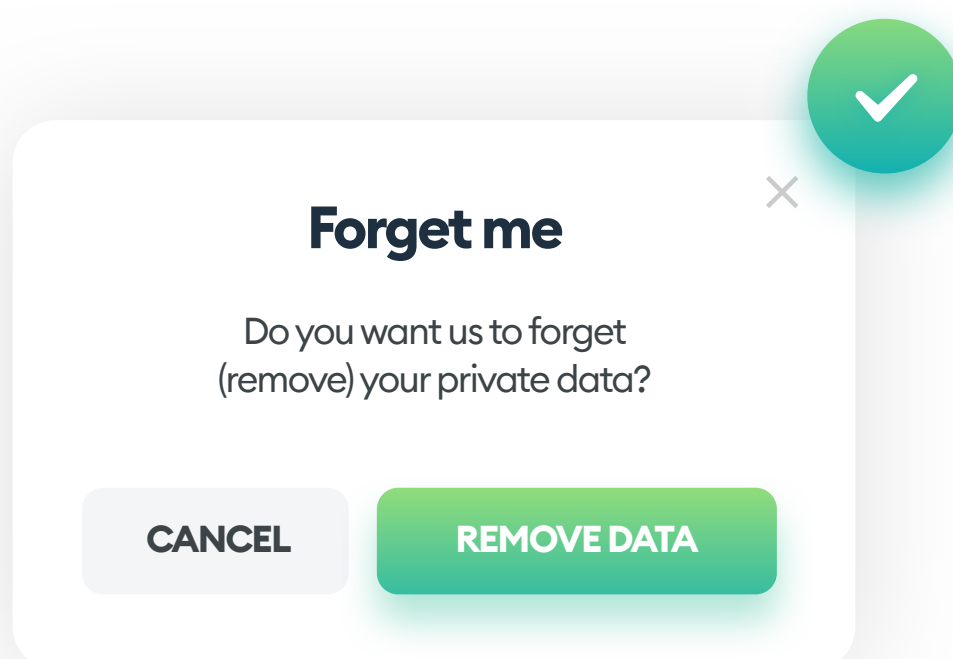


NEGATIVE ACTIONS

Negative actions should only have one button label that confirms the action. With two labels, both sounding similar, it's not apparent which one is the desired one.

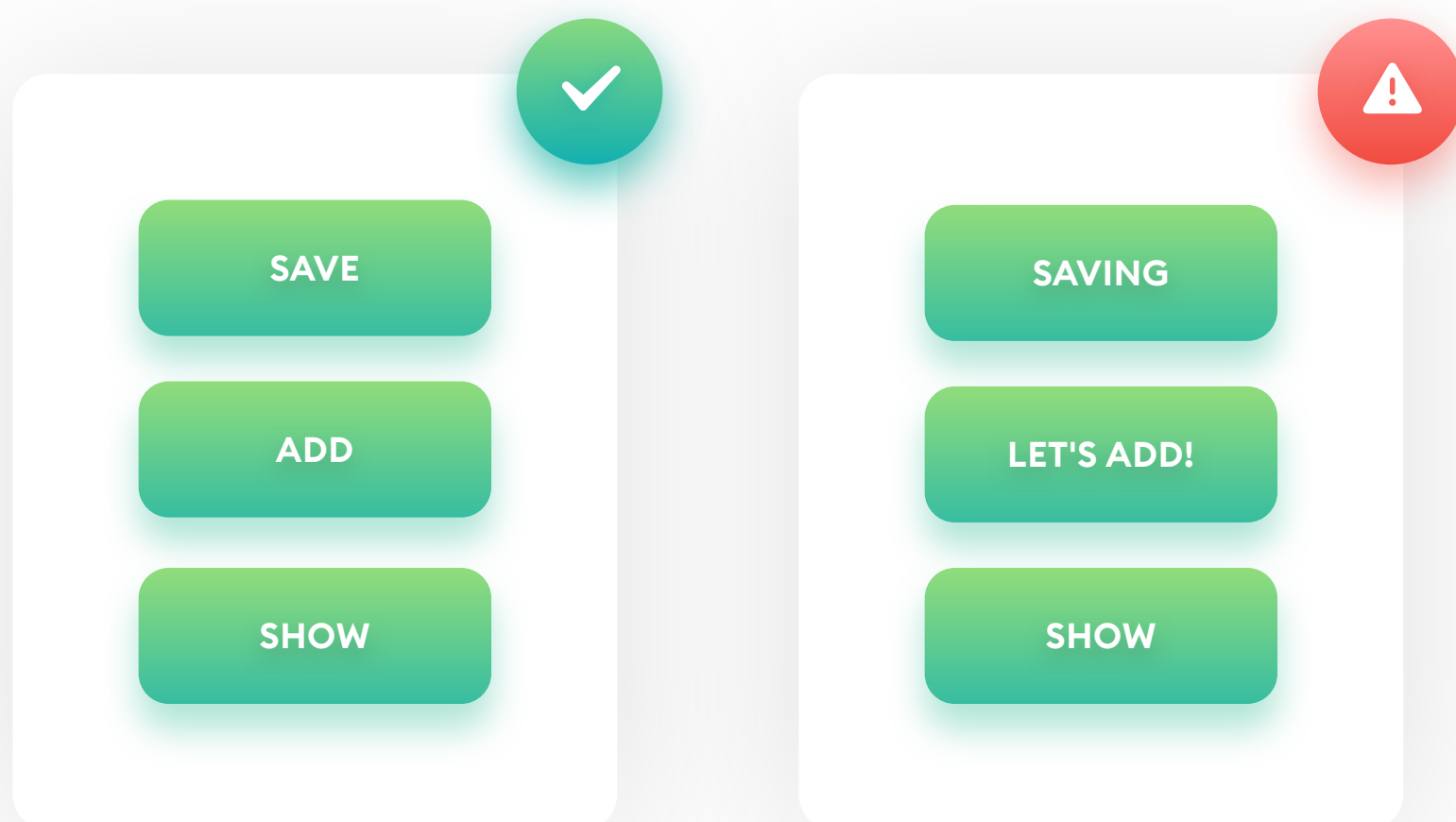


In this scenario, the main action should be "Remove data," while the less important one would be "Cancel." The main action (primary) we can emphasize visually, to stand out from the secondary button.



CONSISTENCY

All of our micro-copy should be consistent. If we decide on an approach, it's best to stick to it with all of our UI elements. Don't mix and match different linguistic styles, especially between same-type elements like buttons.

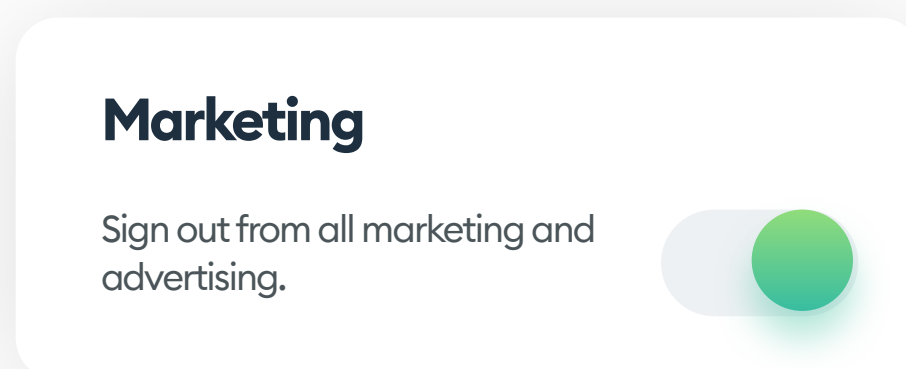


Consistency in language will make the actionable objects easier to understand and much faster to process.

This rule also works for longer descriptions and notifications in the product. Try to use a consistent way of speaking to your user for the best result.

DARK PATTERNS

A Dark Pattern is a UX term that means deliberately lying to the user to achieve the desired outcome. In most cases, dark patterns exist to convince the users to agree to something unknowingly (like giving out private data, location information, or making it harder to opt-out of something).



The example above comes from a real, top-rated product. The label states that you want to sign out from all marketing messages.

The switch next to it would suggest that having it in the ON position would mean you don't want to receive marketing messages. But the truth is that having the switch to OFF is the actual way to opt-out. This difference between the label and the outcome is a way to keep you signed up.

Similar, dishonest ideas are often around the form-agreement checkboxes. In many cases, the labels are written in a way, suggesting that we need to select them all to continue. Only by carefully inspecting the UI we realize that we only need one or two, while the rest is merely a permission to spam us.

These UX-language traps shouldn't exist.

Respect your user!

“

**Don't use Lorem
Ipsum. Ever! ”**

LOREM IPSUM

Lorem Ipsum is an easy path to being lazy and breaking immersion.

Try not to use the "Lorem Ipsum" text in your designs. Not even with initial wireframes and high-level prototypes. If you don't have the right copy yet, you can find something relevant to the screens you're working on and paste it. If you're creating a fitness app, replace lorem ipsum with some real training advice or exercise breakdowns.



Dolor sit amet

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.



Fitness goals

If your goal is to have a strong, healthy posture and higher metabolic rate, you should definitely try this workout category.

If your example copy matches the style and subject of the product, it will make it a lot easier to come up with new ideas and have a "feel" for the design. Faux-Latin breaks that immersion and makes the project look random. Having in-context copy in the design will also help you present it better to stakeholders or other designers to get their feedback.



24.

Design

Styles

Design Styles

INTRODUCTION

Over the years, we have gone through various design styles, coming back and revisiting some of them after a few years. Those styles take cues from general (and well known) art and design styles but have their unique names in UI design.

In 2020 the most popular style is a mix of flat-design and pseudo-3d layering with shadows. It's an evolution of Material Design, that's currently called "modern design."

This chapter will focus on mobile app design styles, as they often come back and influence web design as well.

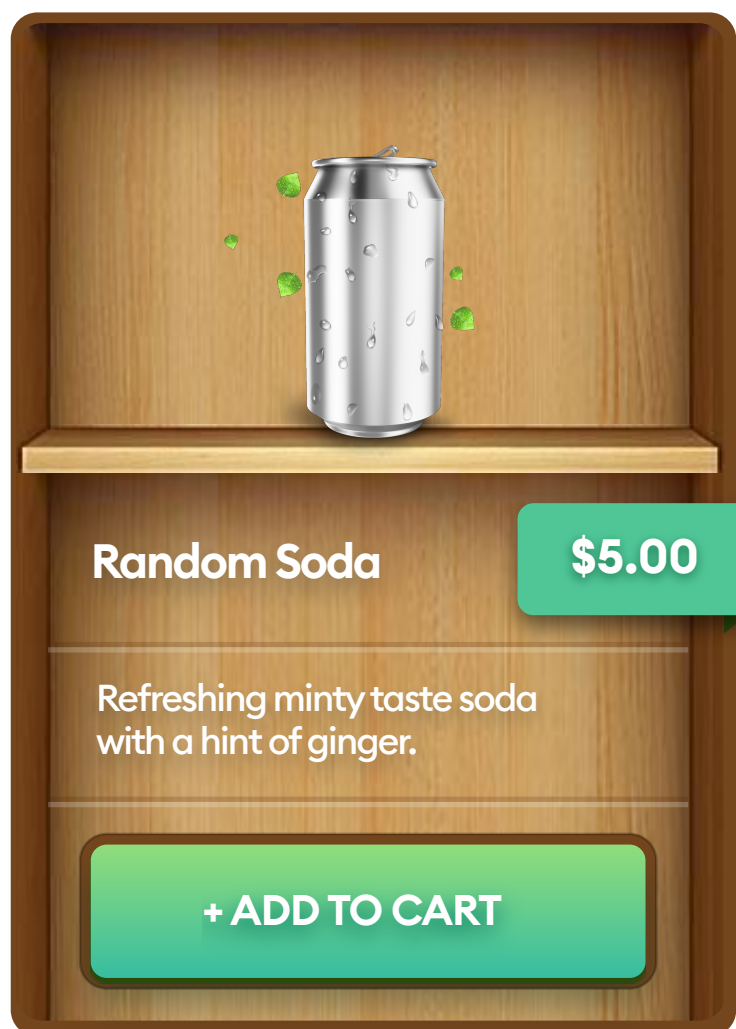


EXERCISE

An excellent exercise to try is choosing a simple object (like a product card) and designing it in all of the styles in one go.

That approach will help you better understand the differences, and how to find ways to combine them in your projects.

SKEUOMORPHISM



Skeuomorphism has been around for a very long time. Even the computer Desktop metaphor (or the trash can) is in fact Skeuomorphic in nature. But in UI, it became especially important with the release of the first iPhone.

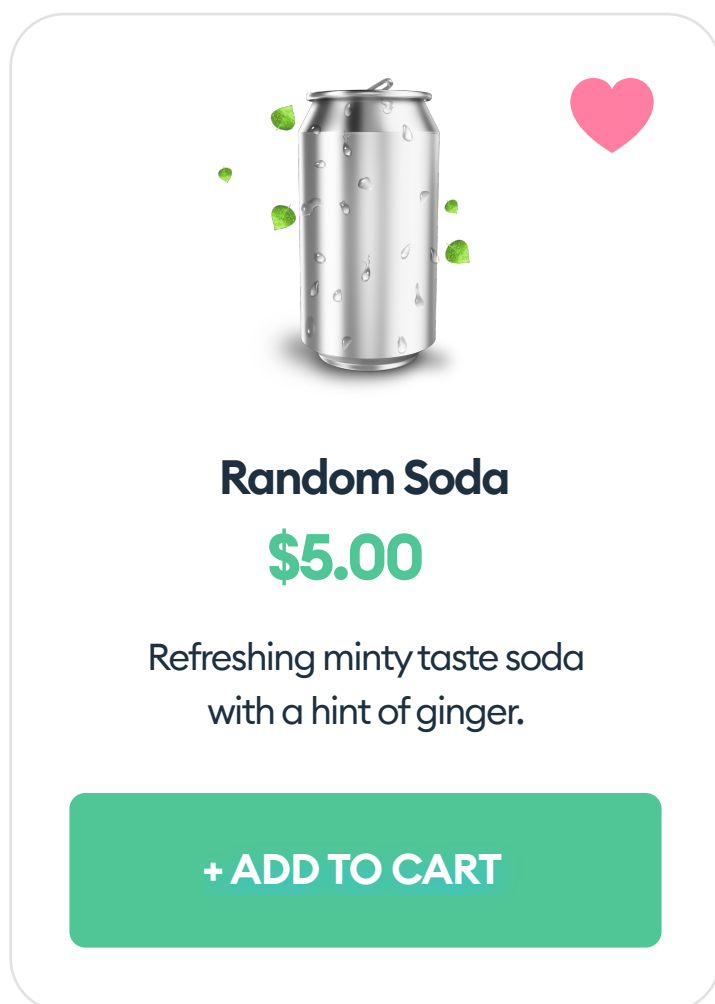
This trend is all about using real objects metaphors to make the UI easier to understand. A library app would have wooden shelves under the books, and everything would try to imitate realistic shadows and highlights.

That digitization of real-world objects was all about helping mental associations for non-tech-savvy users. With new technology, people mustn't be afraid to use it (and the iPhone was entirely novel at the time). In general, Skeuomorphism had a goal of lowering the barrier of entry for new users.

MAIN PROBLEMS WITH SKEUOMORPHISM

Currently, the real "stitched-leather-and-wooden-shelves" Skeuomorphism is as popular in design circles, as Comic Sans is with font enthusiasts. Designers laugh at those rare Skeuomorphic attempts, so the trend is practically non-existent in real products.

It's also a bit difficult to code, as every little element has to be a separate bitmap, while in other design trends, you can generate a lot in code alone. Users understand interfaces a lot better now. They've been using smartphones for years now and know all the possible ways of interacting with their devices.



Flat Design is the complete opposite of Skeuomorphism. The central premise of this style is minimalism. No ornamentation, no effects. That includes the lack of shadows on objects, hence the name "flat."

All of the interface layers exist in the same plane on the Z-axis. When 3d and Skeuomorphism got boring, flat design became popular mostly because of being so different.

An example of flat design at its best is Microsoft's Metro UI made for the now obsolete Windows Phone platform.

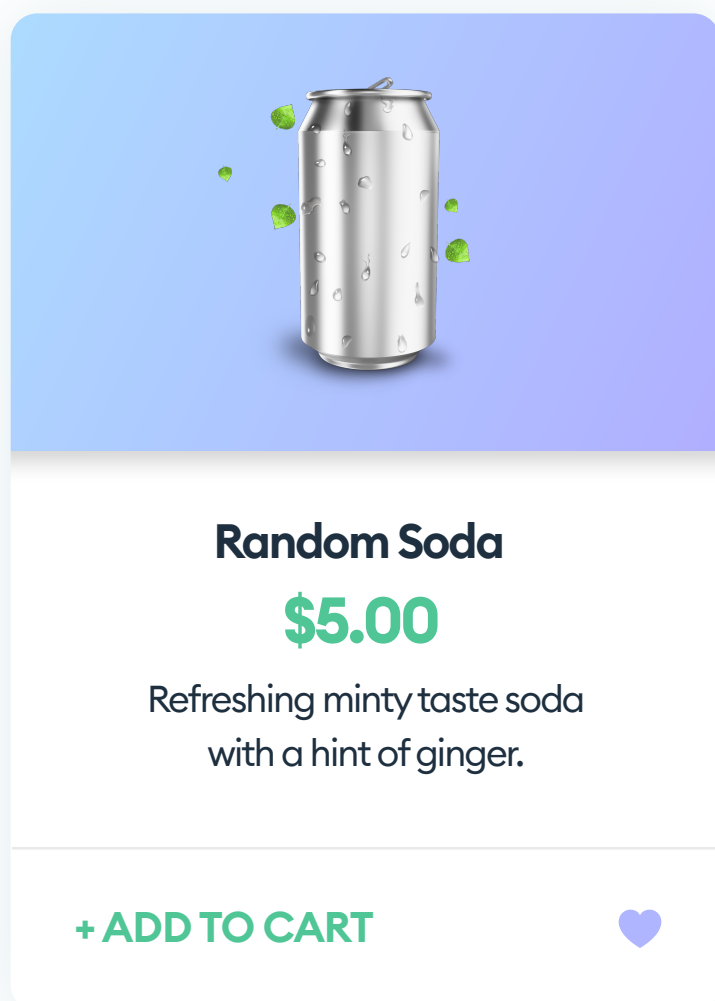
This style is still present, but not it's not highly popular. The main reason for that low adoption is the fact that it's harder to design well without relying on shadows and faux-3d for hierarchy guidance. To create a great flat design, you need to nail typography, spacing, and color at 100%. Otherwise, it won't make an impact or be memorable.

On the user side, there are considerable difficulties with understanding some of the object's purpose. For example, a label can be mistaken for a button. Various studies show that using completely-flat designs is always around 20-22% slower than they're layered counterparts.

All that extra time is necessary to discern active objects from content.

“

**Using products made
with Flat Design,
can be up to 22%
slower. ”**



Material Design is a Design System created by Google in 2014 to help developers create Android apps. Its impact had been powerful, and most of the modern designs take some cues from Material.

The main idea of this design system is to simulate "digital-paper." It means the layers themselves are still flat, but their position on the Z-axis is a way to set hierarchy and order.

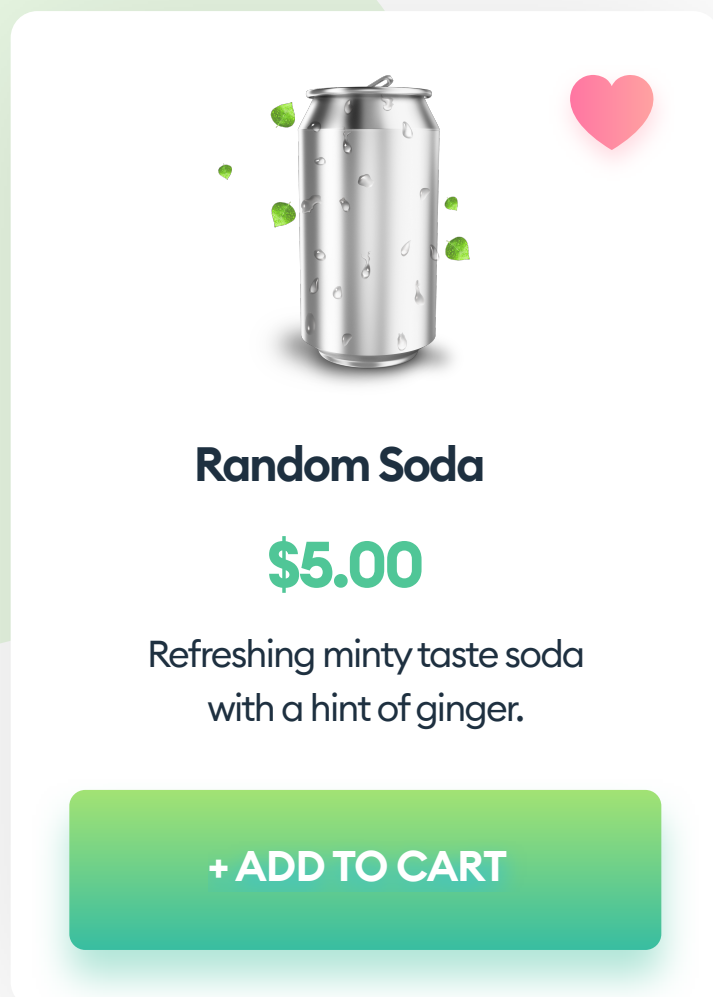
The general rule of Material Design is to have all the important UI elements "closer" to the user. That appearance of depth comes from the use of shadows under otherwise flat card-like objects.

The system uses its own, pre-defined color palettes, with base colors using the 500 number. Darker hues are using numbers below 500, while lighter use the numbers above. The suggested approach to designing in this style is to choose one primary color, one secondary, and use them consistently throughout the product.

It also came with its own font - Roboto, which was redesigned from the very first version and gained a lot of styles and weights.

The main problem with this Design System is that using the same colors, font, and shadows, in essence, make all the Material Design products look alike. Newer versions tried to fix that, but so far rather unsuccessfully.

MODERN DESIGN



Modern design is a mix of Flat and Material, but with a slightly more significant emphasis on breaking the rules, which in Material Design are pretty strict.

Most objects now have subtle shadows (not only the cards but buttons and icons as well), and the shadows are colorful, using the darker primary color as a base. These soft, colored shadows are the characteristic part of this style.

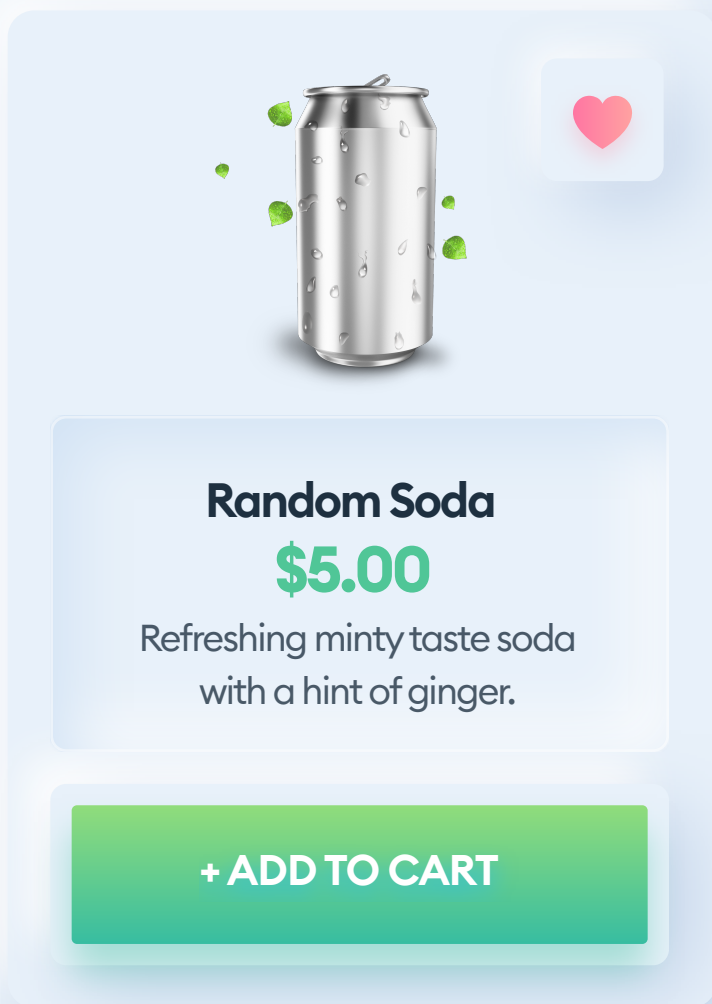
Modern Design was the answer to the problem of Material apps, all looking like re-colored clones of each other. Colorful shadows and angled gradients give the designer a lot more to play with than the constrained rules of Material ever have.

This style is also bringing round, organic shapes back. Of course, you can do sharp corners as well, but most projects done with "Modern" are using rounded corners and rounded buttons. The entire experience is more user-friendly and less generic. Round shapes are also perceived as non-threatening by the users.

The main rule is: bend the rules.

Make your products unique. Place your gradients at an angle, and add curved, organic shapes to the backgrounds. The openness of this approach made it the most popular style for both apps and websites nowadays. While it won't work for very serious business apps, it's a perfect fit for consumer products.

NEUMORPHISM



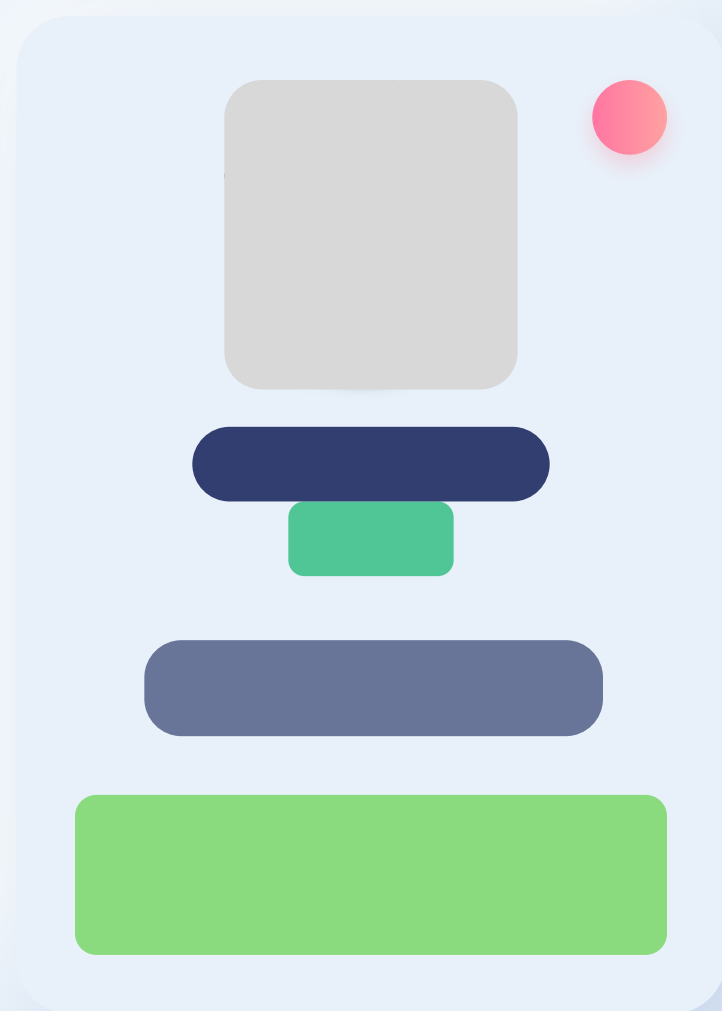
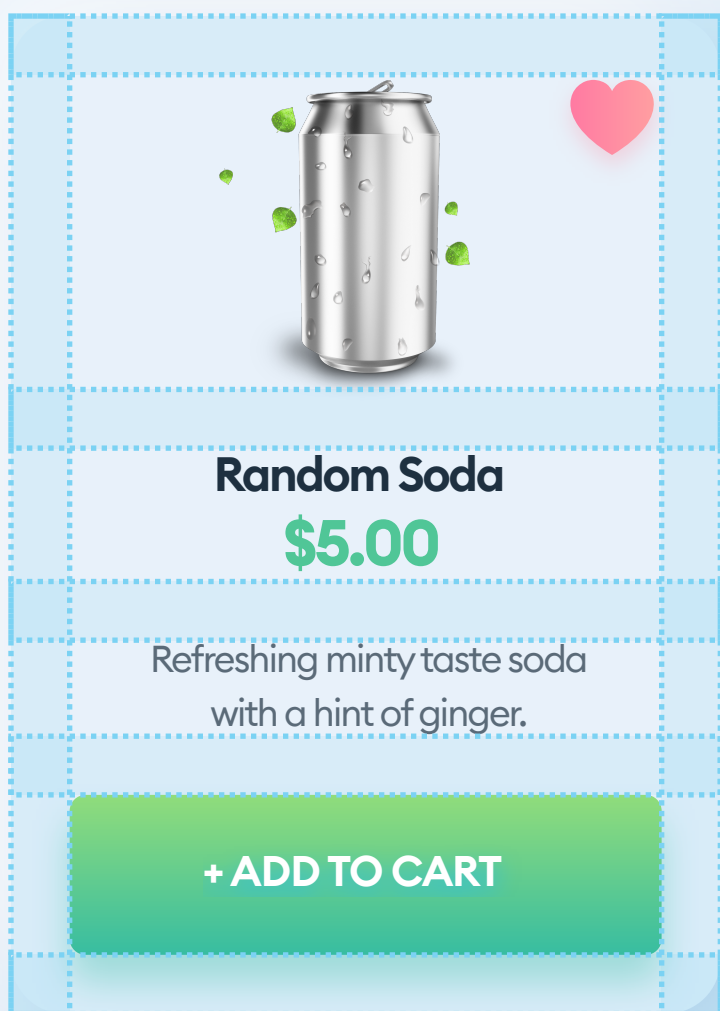
Neumorphism is a "new" design style that first appeared (under that name) in late 2019. What's interesting, Michal (one of the authors of this book) came up with the name and wrote the first long article on the style.

From then on, the illusion of soft, extruded plastic in UI began to spread under the Neumorphism name. This style works best with non-active objects and when it's not overused.

NEUMORPHISM IS TERRIBLE FOR BUTTONS

Avoid using Neumorphism for active elements like buttons or switches, due to shallow edge contrast. While inverting the shadows "looks cool," it's not usable and not accessible. The low contrast will make it problematic to use the interface by both the people with vision problems and people using older, lower contrast displays. Designing a button, make sure that your normal and active states are using underline or font-weight to differentiate. Otherwise, for some users, both states may look exactly the same.





CARDS

We believe that style works best as a background for info or product cards. While the mechanic of it doesn't make sense for anything you can enlarge or swipe (like cards), this rule can be bent if you want to use Neumorphism in your product. Start your card design by removing the background and creating proper spacing, hierarchy, sizes, and colors for everything first. It has to work seamlessly together and be perceived as a whole before you add the neumorphic card underneath.

If the card structure is still easy to discern, even without the background, it means it will also work for people who can't see the subtle shadows. It means that this style works best when it's not necessary - a pure decoration.

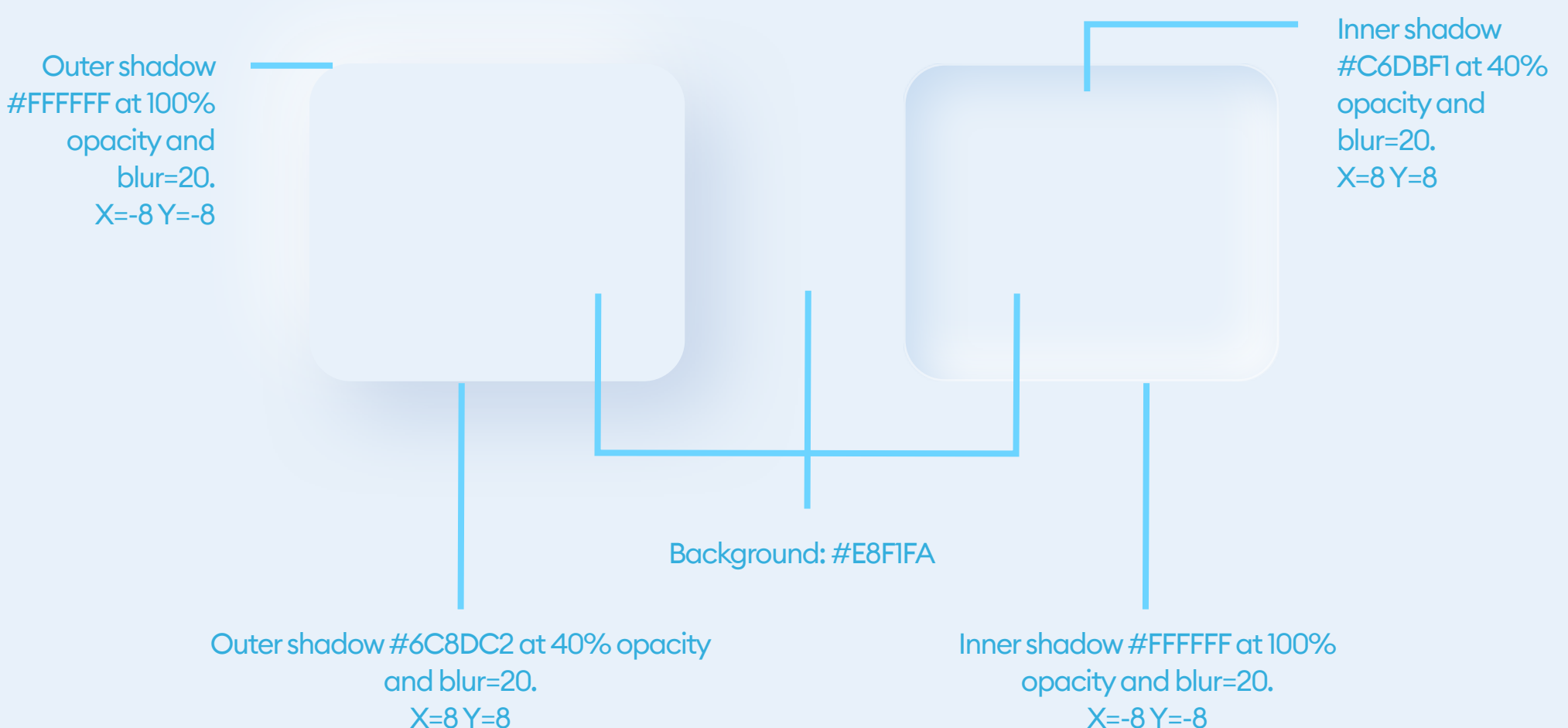
You can mix it with other styles (Material, Modern or even Flat), but avoid creating products that only use Neumorphism and use it extensively across all UI elements.

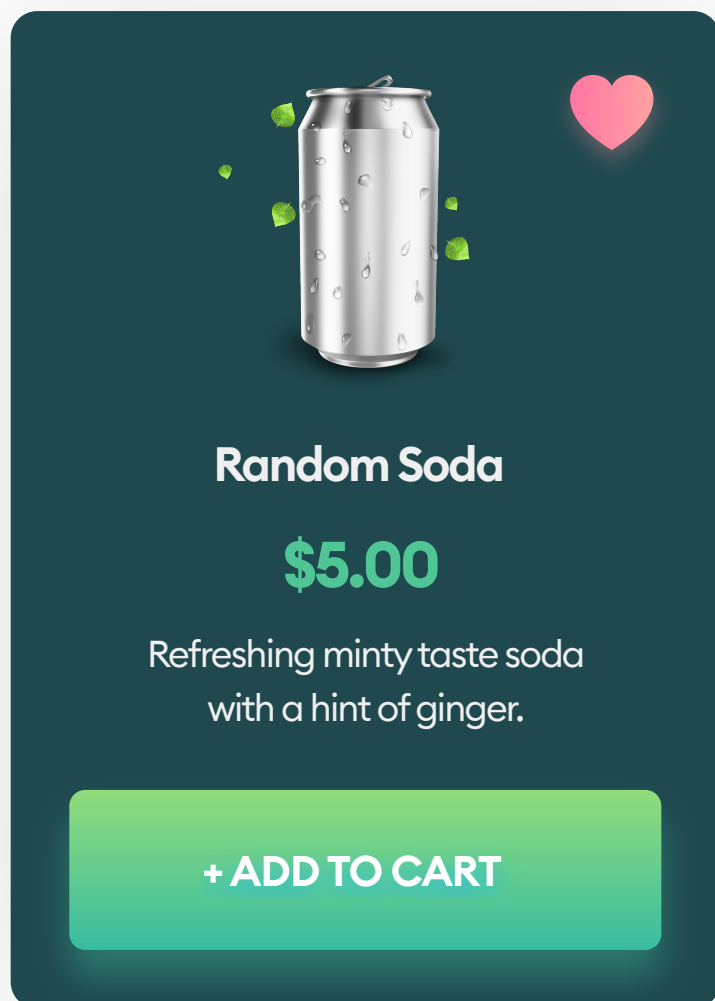
HOW TO ACHIEVE THE EFFECT

The idea behind neumorphism is to use inner or outer shadows as the only defining characteristic of objects. For this to work, the page background needs to be the same color as the object background. Because we're using dark and light shadows at the same time, the background cannot be fully white or fully black as it would render the entire effect pointless.

The background can be grey or have a hint of another color. In our examples, it's a very light desaturated blue. Create your object, fill it with the same color as the background and add two shadows to it, a lighter one on top-left, and a darker one on bottom-right. The blending of the two shadows will create that extruded plastic effect everyone is after.

Changing the drop-shadows to inner-shadows will result in that concave look.





Dark mode became popular with the gaining popularity of OLED screens in mobile devices. In most cases, it's a second interface style of an app or website, and the user can choose between dark and standard versions.

There is not enough data to confirm that dark interfaces are easier on the eyes. They can, however, in some instances, help reduce the energy consumption of your device.

Apple added Dark mode (as an option) to both desktop and mobile devices in 2019. At the same time, Google created a dark version of their Gmail app, and Instagram allowed the user to switch the interface to dark mode.

It's worth remembering that the energy-saving part of dark mode only works on OLED screens (in which dark pixels are simply not backlit) and only if you're using pure black (#000000). Combined with white text, it creates a very high contrast that may make it harder to use comfortably.

If you're using dark-mode based on dark-greys, it's purely an aesthetic choice, and the only gain for the user is the ability to customize the product. Some people prefer dark-themed interfaces, so it may be a good idea to design both in your projects.

ULTRA-MINIMALISM



Random Soda

\$5.00

Refreshing minty taste soda
with a hint of ginger.

+ ADD TO FAVORITES

+ ADD TO CART

Ultra-minimalism is most common in e-ink screen devices (e-book readers, minimalistic smartphones, etc.)

The interface uses mostly monochrome text (and sometimes divider lines) without colors, shadows, or decoration.

In the age of smartphone addiction, simple, minimalistic devices are starting to create a niche. They're tool-based instead of feed-based, with very simple, often text-based interfaces.

This style is also suitable for specialized, professional devices like counters, measuring systems, wiring, and more.

Ultra minimalism will never be the dominating trend, but it does not aspire to become one. It's reserved for niche applications for small, focused user groups. It's also very difficult to design well, so it may be an excellent place to start your design exercises. With that level of simplicity, all the spacing and fonts need to be flawless to work well and look good.

You can try to mix ultra minimalism with colorful photos to achieve impressive, modern-looking results. The main problem of this style, however, is that it's not suited for the mass market. Most consumers won't feel right using an interface based on text alone.

“

**The best approach
is to combine best
parts of different
styles for a unique
look & feel. ”**

TRENDS COME AND GO

Every style gets old after it's been around for a while. Designers get bored with it and start experimenting. In many cases, the experiments are like a pendulum swinging back towards something that used to be popular a few years before. The emergence of Neumorphism is a good example of that trend - it's merging Skeuomorphism with minimalism for something brand new (even if flawed). It's almost sure that the current style of soft, colorful shadows will not be dominant in a few years anymore.

But it's important to understand that it's not the style itself that matters. It's your understanding of the basic design principles of grid, typography, colors, and knowledge on how to put it together for the best result. You can create an excellent design in ANY style.

Even if your background is a wooden plank and your buttons have a metal texture - if the grid, typography, and colors are right, it will look good.

Picking a design style for your product requires some thought on who are your users exactly.

How tech-savvy are they? Are they young or old?

Some groups (like senior citizens) would still prefer to have their content on wooden shelves instead of abstract "floating paper" layers. That real-world connection still helps people with understanding the interface, even if it's not as trendy anymore.

What we suggest is for you to explore mixing and matching various styles to create something of your own. Nobody wants all the products to look the same after all.



25.

Design Process

Design Process

KEEP YOUR DESIGN FILES TIDY

Even when doing design exercises, you need to make sure your design files are organized and tidy.

When you make a rectangle, the common name for it - "Rectangle 1" appears on your layer list. Avoid having instances like "Rectangle 1 copy", "Rectangle 1 copy 1", "Rectangle 1 copy 2." etc. Of course, when you design quickly, you can let yourself go, but before you move on to the next screen - tidy up.

COMPONENTS

No matter where in the world you are, name your components and groups in English. It will make it a lot easier if a foreign person joins your team. The naming convention should be agreed upon by your team, but what works best is super simple, to the point options.



CTA - BUTTON - SMALL - GREEN GRADIENT
CTA / BUTTON / SMALL / GREEN GRADIENT

Use gradation to name your layers and groups. You can either use the "/" or the "-" to set the name and the general family for an object.

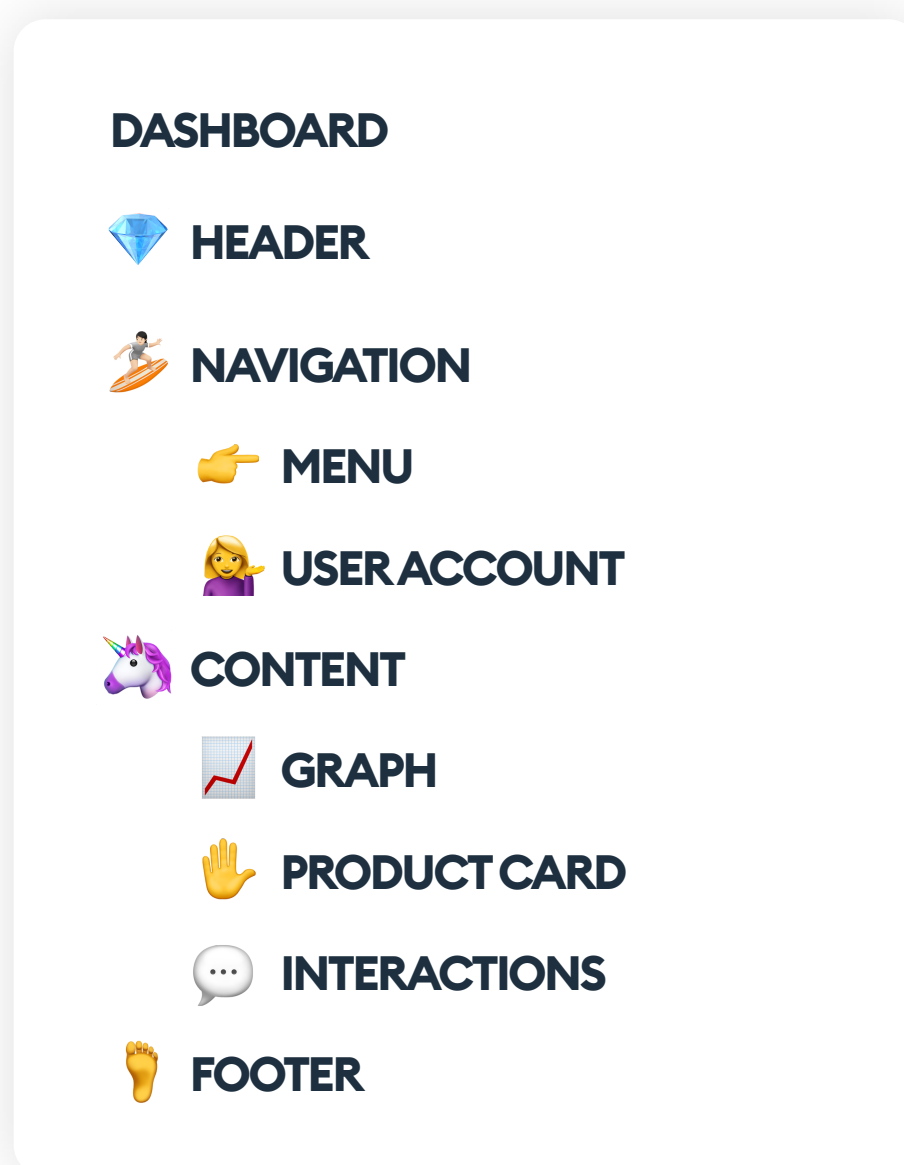
When you turn your group into a Symbol/Component, it often results in a beautiful folder-like structure in your component lists. It helps a lot with clarity and finding the right thing quickly.

“

**Naming layers and
groups should
become your
second nature.”**

TOP LEVEL SECTIONS

When you've named all the individual objects (like icons, buttons, windows), it's time to combine them all into groups. The most obvious group types are "Header," "Navigation," "Content," and "Footer. Of course, every project is different, so that you may need a completely different set. We found out that using emoji for specific group types will make it a lot faster to navigate even the most complex layer lists quickly. It's simply faster than reading text alone.

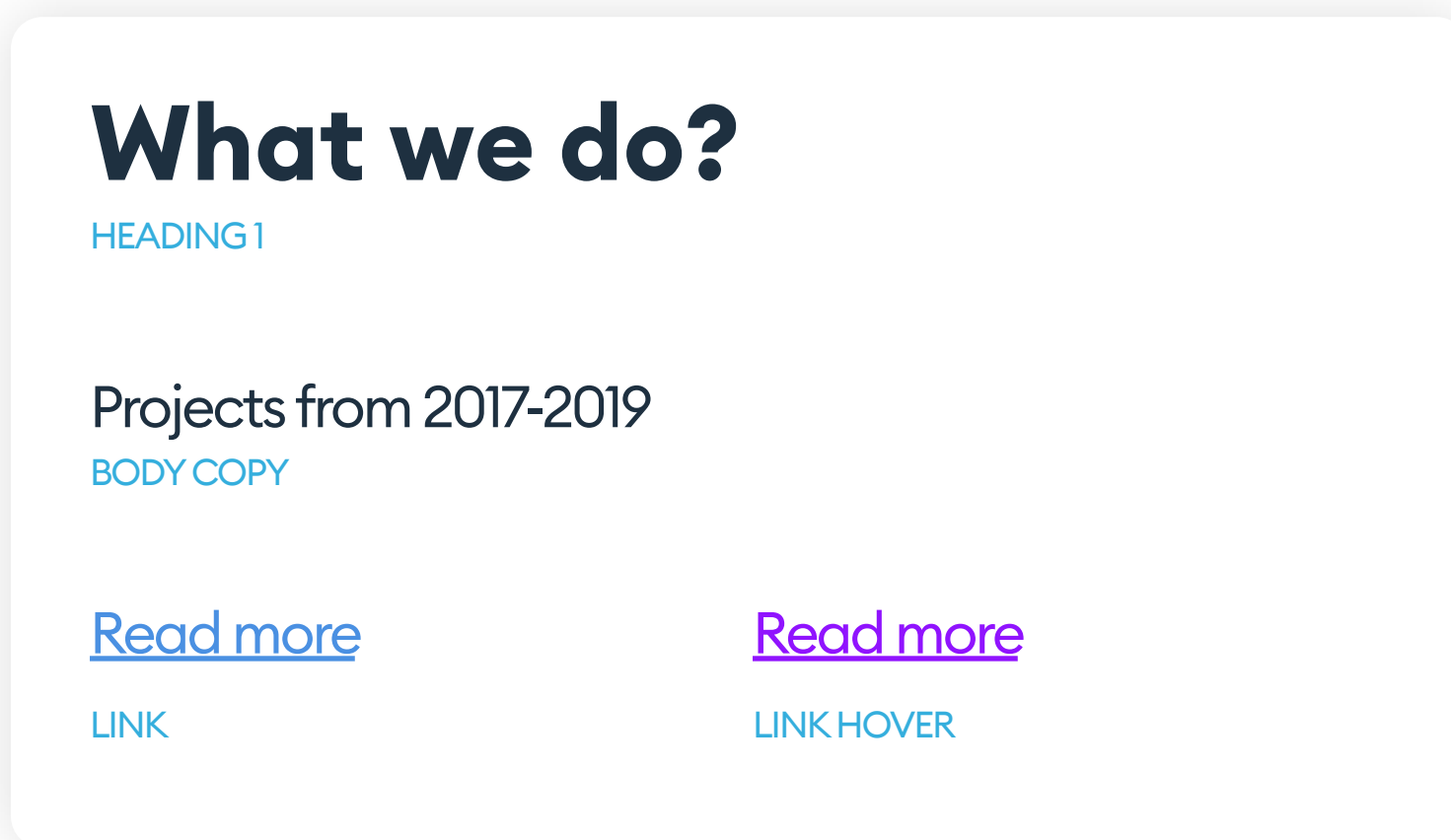


The emojis can be very literal (a graph, human foot for the footer), or more abstract. It's essential, however, that they are consistent across the entire project. Each high-level group gets an icon! Of course, the individual layers or smaller groups don't need an emoji - that would be overkill.

Once you have an emoji-system in place, you'll be able to adjust a large number of screens quickly.

TEXT STYLES

After designing the major screens of your project, it's time to start creating the text styles. Most design tools allow for easy naming of the text styles (like "Heading 1", "Body Copy" or "Link").



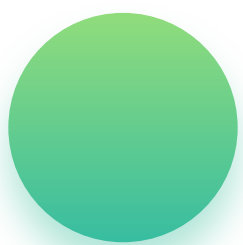
That way, while making future screens, you can choose the right style from a list and keep the entire project consistent. The first few screens should serve as a baseline - as this is your testing ground - but after you're done with them, it's time to define the rules and stick to them. It will also make it a lot faster to modify a text style across all the screens, by changing it in one place.

HOW TO NAME YOUR TEXT STYLES?

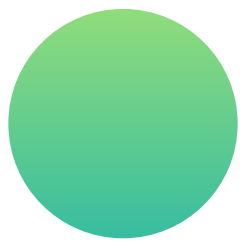
The best naming convention comes from HTML and CSS. Name your headers in the way they work in code (H1, H2, H3, H4). The same goes for your paragraphs (P), and your links. It will make it a lot easier for the developers to understand the font hierarchy.

LAYER STYLES

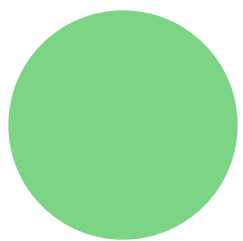
Layer styles save the properties of the object under a common name. They include the fill (color or gradient), stroke (or no-stroke), shadows, opacity, and blurs. You can apply them to all of your shapes (rectangles, ovals, icons) to define how they look.



GRADIENTS / GREEN - SHADOW



GRADIENTS / GREEN



COLORS / GREEN

LAYER STYLES

COLORS

 GREEN

 BLUE

GRADIENTS

 GREEN

 GREEN - SHADOW

 PURPLE

Try to create groups for your styles, so they'll neatly fold into an expanding tree-like structure. It will make it a lot easier to manage them, add new ones, or remove redundancies.

When building a new object, try to use already existing styles on it whenever possible. The more styles, the more chaotic your layout can get, so limit the number and avoid adding new ones unless they're necessary.

Sadly, most layer styles don't save the border-radius values, so we need to keep that consistent ourselves. Still, layer styles are the best way to hold shadow and fill definitions of your objects in one place.

You can showcase your styles with a bit of CSS code on a separate artboard, to have a very early attempt at a design system. That initial style guide will help your developers with implementing those styles as CSS classes. You can export those individual style guide-artboards as PDFs or upload them to a developer-friendly platform (inVision, Zeplin, Avocode, and more). Don't worry about the CSS code clarity. The design tools export way too many properties, but your developers should be able to only use the parts they really need.

Typography

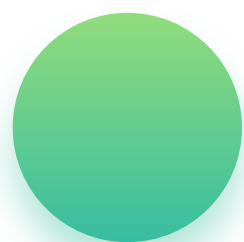
Heading 1

```
font-family: WorkSans-ExtraBold;  
font-size: 20px;  
color: #1E3040;  
letter-spacing: -1px;  
line-height: 15px;
```

Body Copy

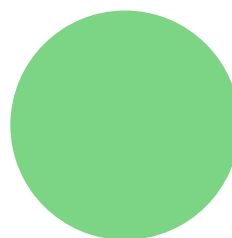
```
font-family: WorkSans-Regular;  
font-size: 11px;  
color: #1E3040;  
letter-spacing: -0.55px;  
line-height: 15px;
```

Colors and Gradients



GRADIENTS / GREEN - SHADOW

```
background-image: linear-gradient(0deg,  
#13B0B0 0%, #F4FF51 100%);  
box-shadow: 0 5px 7px 0 rgba(63,192,158,0.40);
```



COLORS / GREEN

```
background: #7CD584;
```



26.

Design Systems

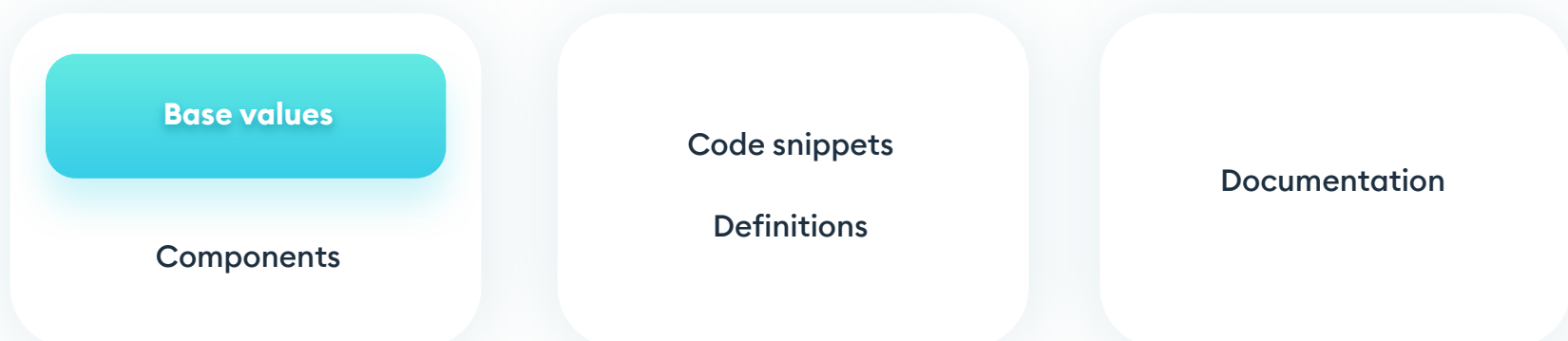
Design Systems

WHAT IS A DESIGN SYSTEM?

A design system is a collection of methods and processes that make the teams building digital products collaborate in an organized manner.

In other words, it's a documented library of design components, ready to use in code. These components vary in complexity and can be nested inside one another. They include things like buttons, form fields, but also basic layout settings (typography rules, margins, colors). It should also include code-ready definitions (either as code-snippets or CSS classes) for the developers.

It's a modern take on the trusty styleguide that comes at an intersection of design, coding, and future-proofing products.



Brad Frost wrote about the term Atomic Design, which for many, has become an outline for a good design system. Its premise is that starting small (with atoms, or base values), we define more and more complex structures, ending up with the final page layouts.



WHY?

The main advantages of a design system are much higher levels of consistency and faster, more precise implementations. In other words, the team can focus on actually building innovation without spending time figuring out which pieces to use for building a popup window.

Most digital product creation processes are an effect of the collaboration between a designer and a developer (with others around that core). If the team is small, it's easy for them to understand each other. Of course, some of the messages always get distorted. That's the nature of collaboration. When the team starts to grow, communication spreads outside the official, organized channels. Some decisions happen during lunch, some on a cigarette break, some on slack, emails, or in the company elevator. This communication spread can lead to chaos and cause consistency and quality problems like the one outlined below.

INTERNAL CONSISTENCY

E-mail

E-mail

Internal consistency is a vital element of a successful product. It means the same kind of object within the product should always look and work the same way. One huge brand that we analyzed has completely different styles for their login and registration forms—all in the confines of one single app. That causes confusion and frustration in users. They were expecting something and got something different in return. If you don't push consistency from the start, it will only get more chaotic as the product grows.

SOURCE OF TRUTH

Most design systems start as a component/symbol library in your favorite design tool. That's a great start, but to make your system genuinely work in your company, you need to document it. That documentation becomes your "Source of Truth."

You should update it regularly, as design systems are never really complete. New research, trends, and features can influence global changes within your system.

HOW TO DOCUMENT?

The first step of any documentation is all about setting the basic rules:

- ✓ What's the update process (who and how accepts a change?)
- ✓ What's the format and the language used (has to match the company)
- ✓ Who is responsible for improving the Design System

All companies are different, so the way you build out your documentation is relatively open. Make it fit your business and its needs. The most important thing to remember is being precise - document every element as thoroughly as possible, including its use-cases.

A Design System should match your company, not the other way around. Treat it like your internal company product and work on it with that mindset from the start.

“

**A Design System
should fit your
organization,
not the other
way around.”**

WHERE TO START?

When building a design system for an existing brand, you need to start by analyzing the brand book and existing digital products the company has. Find all the main defining points and write them down.

The things you need to define first are colors, fonts, and the grid. They are the top-level qualities of our design library. They influence every single component and layout you build.

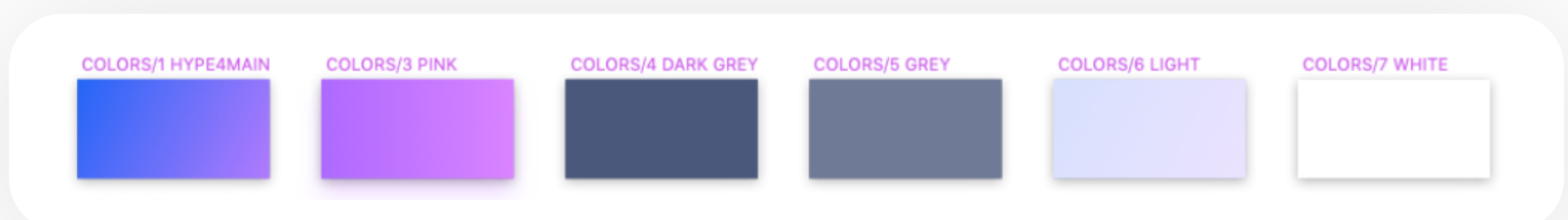
A button is a combination of shape, color, margins, and a font. Split it into its individual components and document each one separately.



The shadow style can have its individual name, or be a category group. An example of the description can look like this:

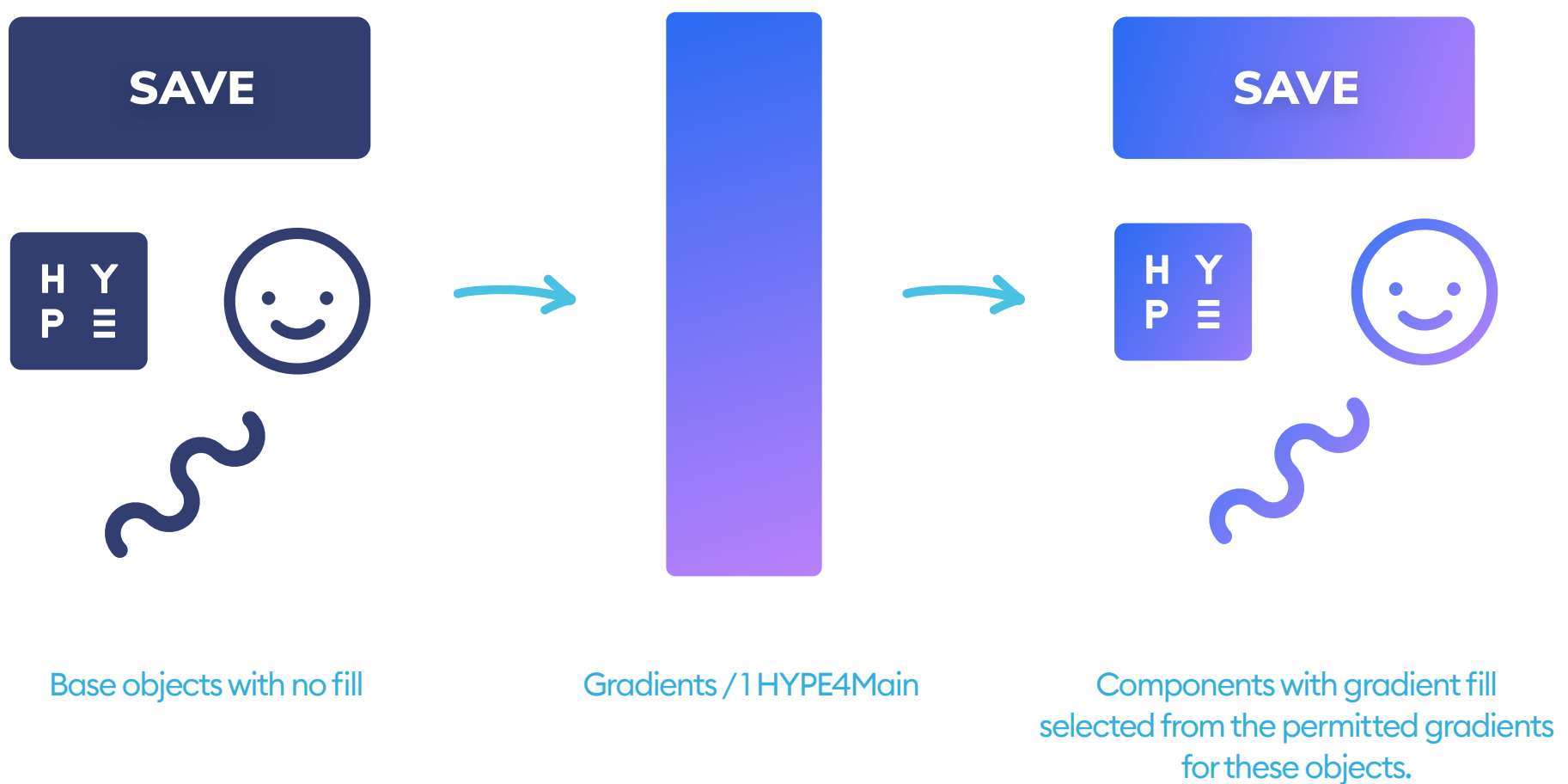
Shadow / CTA: Color, blur value, position.

Define your colors first so that they can merge with the shadow description as well. Give them consistent, easy to understand names. Here's an example of how we documented our brand colors and gradients.



NESTING AND SHARED PROPERTIES

When you define your color library and the ways they can be used (which color goes where), you can start assigning them to objects. That means both the obvious ones (buttons, backgrounds), but also others (like icons or decorations.)



Each UI element should have a set of clear rules on which colors and gradients it can use. You can reserve the primary gradient for CTA only, and some specific color just for the icons. Be precise.

Even in the design phase, try to use only colors coming from your library. Avoid individually setting colors of your objects by hand. They all should connect to the style guide and come from there.


Thanks to this approach, when you decide to update the system with a new set of rules, the objects will change automatically, saving you hours of work.

All members of the design system team must understand CSS. It's not difficult to learn, and you can copy an object's CSS code directly from within your design tool. The nature of CSS will make it easier to get the properties, even if they don't use CSS in their code. It became the universal design-to-code language.




Button Shadow

```
box-shadow: 0 4px 9px -4px #8C6EE8;
```



Colors / Main Gradient

```
background-image: linear-gradient(-49deg, #8E74FF 0%, #2781F0 100%); border-radius: 3px;
```



Global rules

```
font-family: Muli-Black; font-size: 10px; color: #FFFFFF; letter-spacing: 0.15px; text-align: center; text-shadow: 0 2px 4px rgba(0,0,0,0.12); line-height: 44px; padding: 0 24px;
```

COMMON PARTS

The next step is defining the common code parts. If your button line-height is always 44 points, you can call it "Line-height/button/standard" in your documentation. Or `$button-lh` if you're using design tokens.

Avoid mixing real, numeric values with code. If all the code documentation uses are pre-set numbers from a single source (design tokens), you can have a lot more control of the core of your Design System.

SOFT GRID AND DESIGN TOKENS

An excellent approach to grids in Design Systems is setting a soft-grid on your defined base-value (10 or 8 points). Then you can use tokens to describe the following values for use across the system. An 8-point base would look something like this:

Token	Size
<code>\$grid-base</code>	8px
<code>\$grid-small</code>	16px
<code>\$grid-medium</code>	24px
<code>\$grid-large</code>	32px
<code>\$grid-xlarge</code>	48px

DESIGN TOKEN

A design token is an overriding value that you define in one place and then use its token everywhere else. It allows for faster global changes in the future.

The best set up allows for both the wiki and code snippets to grab those number values from a separate JSON file. Avoid entering pixel values in both the components and the documentation wiki.



Properties

line-height: `$grid-xlarge`
height: `$grid-xlarge`

Margins

padding: 0 `$grid-medium`;
margin: `$grid-small`

Component definitions (for example, a button's size, spacing, margins, and line-height) should use Design Tokens whenever possible. You can implement them in more unusual places as well, or multiply them by other defined numbers (like the Golden Ratio).

Your documentation should be a structured wiki with versioning, change tracking, and notes. The history (previous versions) of each page should be easily accessible (a dropdown may be a good idea) from each page.

It seems obvious that the best approach here is to use a CMS, right?

Based on our experiences with both large and small systems and companies, it's best to build it out in code only. The only person able to make changes should be the developer. The harder the modifications, the less chaos.

It also means the developer has to be present at every Design System related meeting. It will help him understand the changes and rules a lot better, as he can ask specific questions when adding the changes to the wiki. The goal is to have the developer be one of the most active members of the product team.

It helps to conquer the chaos of too many admins and set precise rules on every modification to the system. All new versions should use the date as their versioning characteristic, as it makes them a lot easier to find.

We tried a couple of different technologies in our Design Systems but finally settled on ReactJS. Of course, you can build yours in any technology you want (including just CSS), but React is made for a systemic approach. Those React code snippets also have a CSS version separately available. That's because using React for every possible use-case would be an overkill. But having the CSS snippets allows you to keep things like animated banner ads consistent with the system as well.

If your product is Swift or Java-based, you can also add platform-specific definitions (Colors in RGBA instead of HEX values), but even then add the CSS as well.

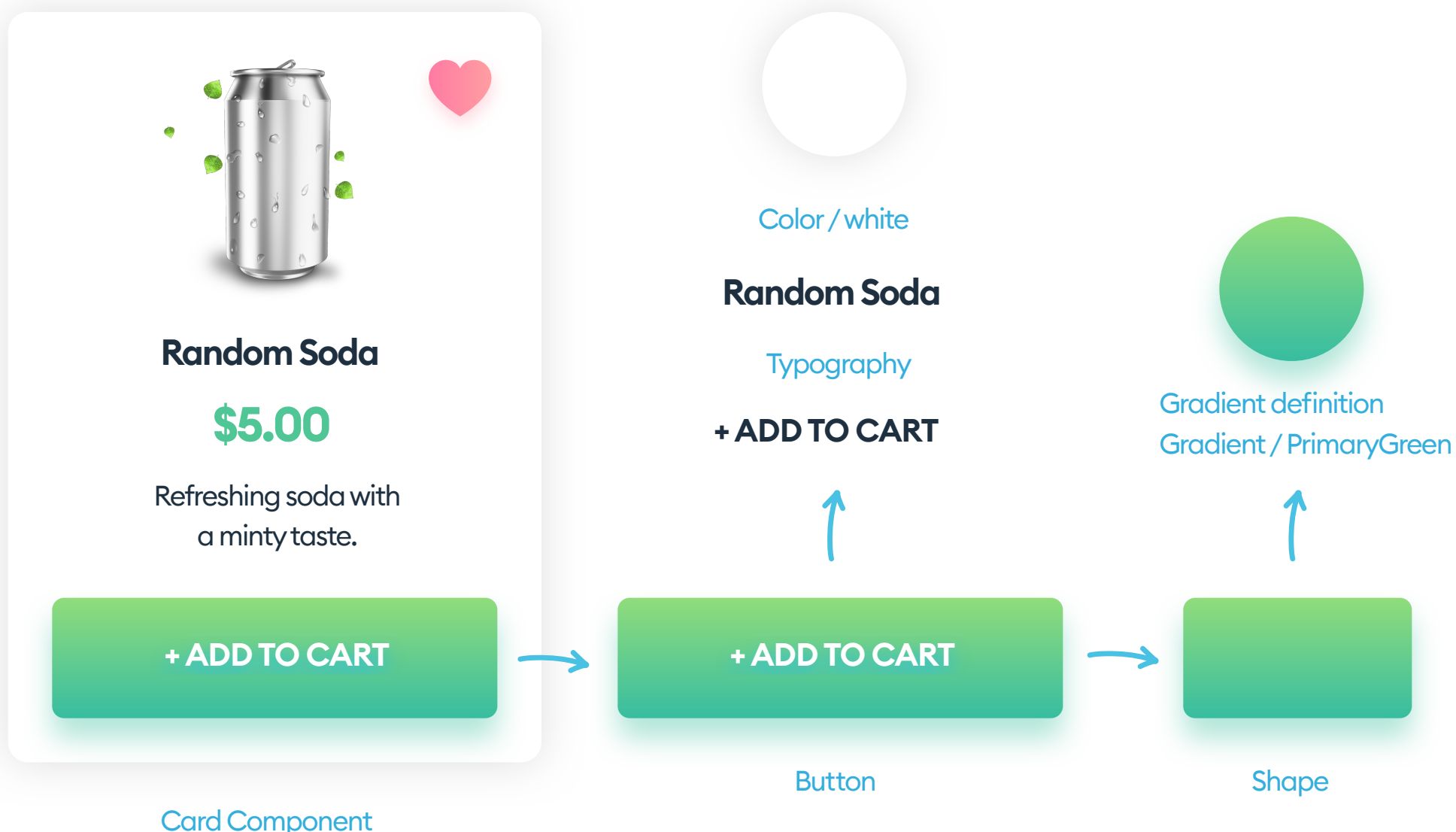
VISUAL DOCUMENTATION

Each page of the system wiki should also have a place for visual documentation - mostly in the form of screenshots with short descriptions. You can use this to show specific examples of each component in a real-world scenario.

The way you write or structure your documentation is up to you. Remember to be as specific as possible and always state the obvious as well. That last fact will help new employees quickly familiarise themselves with the system.

TREE-LIKE STRUCTURE

When describing larger components (like a product card), you specify (and link) all the individual objects the component is made of. They all should link to their own specific wiki pages.



That kind of documentation structure will allow you to track exactly how the element is built, down to its smallest non-divisible parts (atoms). It also helps with finding the relations between objects and planning future product changes.

You can name each step of that nesting structure based on the atomic design approach, but it's not necessary. You can use your own names as long as the team agrees on the same naming convention.

RESEARCH AND DEVELOPMENT

When the descriptions, versioning, and history are all in, you can add the ability to document your research. We do it through a separate wiki page that exists between our regular history pages. The best way to explain it is by example:



Let's say the radio-button component interaction analysis shows a large number of clicks fell outside of its area. It means that many people missed it at least once when they tried to use it. Further analysis showed that the course of action is making both the radio-button and the active area larger.

After implementing that change in an A/B test has proven that the number of missed clicks has dropped drastically. Users were able to access the component a lot faster.

The A/B Test results are then published on that bridging page between versions. If we're doing click-tracking (like in this example), we also publish them over the components to show the actual ratios. The research page serves as the reason for the change. It proves the change was made as an informed decision. When a new team member joins the project, they can clearly see the decision and what it was influenced by.



“

**A Design System
can help you focus
on innovating,
without worrying
about small details.”**

THE TEAM

To create a Design System team, you need at least:

- ✓ a UI/UX designer
- ✓ a product owner
- ✓ a front-end developer

The best developers for the job are the ones who understand design and have at least the basic UI design knowledge already. Same with the designer - it's best if they understand at least the principles behind CSS.

SUMMARY

A Systemic Approach to components and screens is the future of design. It allows you to focus on what's important (how things work) and not worry about the quality (as the components are pre-made and built correctly). Even if you don't plan on doing a large-scale Design System, at least try a systemic approach with documenting your design decisions outside of your design files.

It doesn't mean that each project and functionality can only use the same building blocks. You can use them as a base, but when designing completely new functions for the product, you can create a non-standard UI and simply add it to the Design System.

That approach still leaves enough space for innovation, even in the visual details of your components. The Design System should be your baseline, but feel free to expand it and build additional components as long as they bring value to the project.



27.

UI Audit

UI Audit

HOW TO ANALYSE THE UI?

When we look into redesigning an existing product, it's best to start with auditing the UI. It helps to define what we know about the design and outline the areas that need the most help.

Even if we're the author of the design, it's essential to audit it every few months. This practice helps with staying up to date on the most exciting design concepts and best practices. Auditing often allows us to find and fix consistency problems that would otherwise build up, causing design-debt.

The audits role is not to match the current hot design trend - they change way too often for it to make sense. The goal here is to check consistency, readability, colors, accessibility, and general style. Each audit should end with a short document outlining the most critical areas to improve.

QA - QUALITY ASSURANCE

Quality Assurance is an often-overlooked process. Its underlying goal is to check how consistent the initial design is with the coded product.

There are many methods to achieve this, but the most important thing to know is that QA is essential to UI quality. You should always find the time to go through this process.

AUDIT COMPONENTS

The first step of the audit should be documenting all the screens and interactions our product has. That includes states of the screen (like error messages or invalid form inputs). The next part is counting and analyzing all of the interface elements. Try to be clear and concise in your notes. The goal is to answer questions like:

- ✓ How many typefaces and font sizes, weights, and styles our project uses? How many different ones are on each of the screens?
- ✓ Count the colors and gradients in your product. Try to define the purpose of each one.
- ✓ Check the grid and layout of the project. What are the margins, paddings, and safe spaces?
- ✓ What kinds of shadows, borders, and border radii does the product use?
- ✓ Do all the essential elements (both interactive and informational) adhere to the WCAG accessibility guidelines? Or do you have a special interface version for people with impaired vision?

In the next pages, we'll go through each of these steps in detail and show you how to solve these issues the best way.

DOCUMENT EVERYTHING

While documenting, make sure you can do it in a tool to which most of your team members have access. Having many sets of eyes do the audit allows for a broader perspective. Use Google Docs, Dropbox Paper, or other, similar tools.

“

**Nearly all digital
products get more
chaotic with time.”**

TYPOGRAPHY

Once we have our list of screen designs, we start writing down all the fonts styles, sizes, and weights for each screen. For example:

LOGIN SCREEN

Open Sans

- 11pt, Bold
- 17pt, Medium
- 14pt, Regular
- 20pt, Bold
- 22pt, Medium

Pacifico

- 14pt, Bold
- 14pt, Regular

As you can see, we have eight different font styles on just one screen alone. Try not to exceed 3-4 styles per screen, as more than that causes processing delays and makes your design harder to understand.

We use that method to analyze all of the remaining screens. You can put the table into a spreadsheet, with font styles in the rows and their usage counts in the column. This approach will help you find which styles and fonts are the most common, and which are only present once or twice. You can safely go on and get rid of the rare font-cases by replacing them with their nearest common style.

Open Sans

11pt, Bold
17pt, Medium
14pt, Regular
14pt, Bold, Underline
19pt, ExtraBold

Screens using it

17
1
15
2
11

TYPOGRAPHY

Based on the gathered data, we then define the fonts, styles, and sizes for the essential UI elements. Try to re-use as many styles as possible - in case of typography, less is more.

For example, a text link with the underline removed can also work great as the button label - both have the same function after all. Go on and remove as many styles as you can, while keeping hierarchy and structure.

Consistency is the key.

Heading (H1)

Subheading (H2)

Paragraph text (P)

[Text Link](#)

Button label

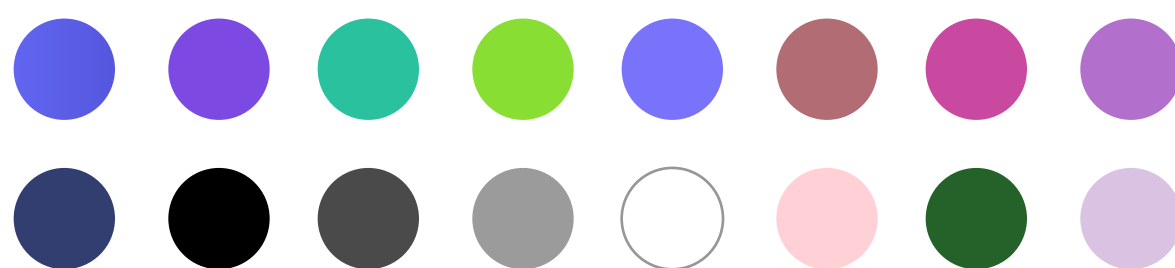
Meta Label

COLORS

Go through every screen and grab the colors and gradient values with a color picker tool. Try not to miss any elements - all backgrounds, buttons, text, icons, and borders.



Colors defined on project start



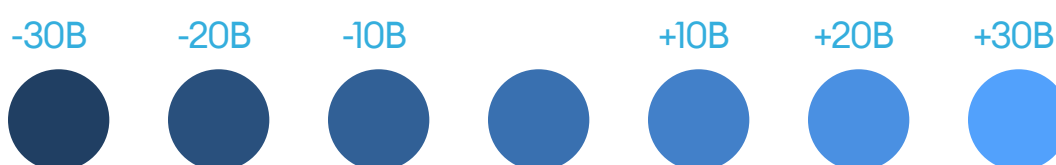
Additional, random colors found in the audit

The example above shows that after a while, we tend to deviate from our initially precise set of color guidelines.

Before we start removing the excess colors, however, let's focus on trying to find out if they had a purpose. If an additional shade was necessary for the UI to make more sense, we keep it and add it to our initial list. Try only to remove colors that seem random or unnecessary.

COLOR VARIANTS

You can use variants of your base colors if you need more shades, but prefer to keep the interface consistent. Try experimenting with higher and lower brightness values to define those variants.



Base color

GRADIENTS

The gradient audit works the same way. The main difference here is that we also have to write down the angle and the type of our gradient (linear, radial or angular)

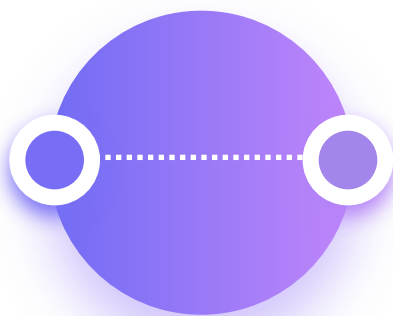


Gradients defined initially



Inconsistent gradients found

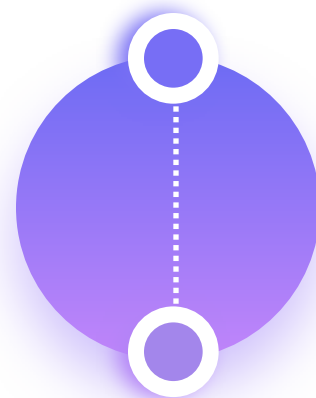
As you can see in the example above, even the same gradient can have different angles that negatively affect consistency. Each instance of inconsistency should be documented and outlined.



90° Angle



45° Angle

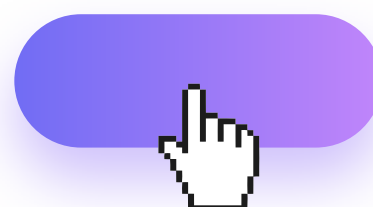


180° Angle

Don't forget to test the hover effects of your items (both cards and buttons are an excellent place to start). Sometimes an effect can only be seen once you hover or activate an element.



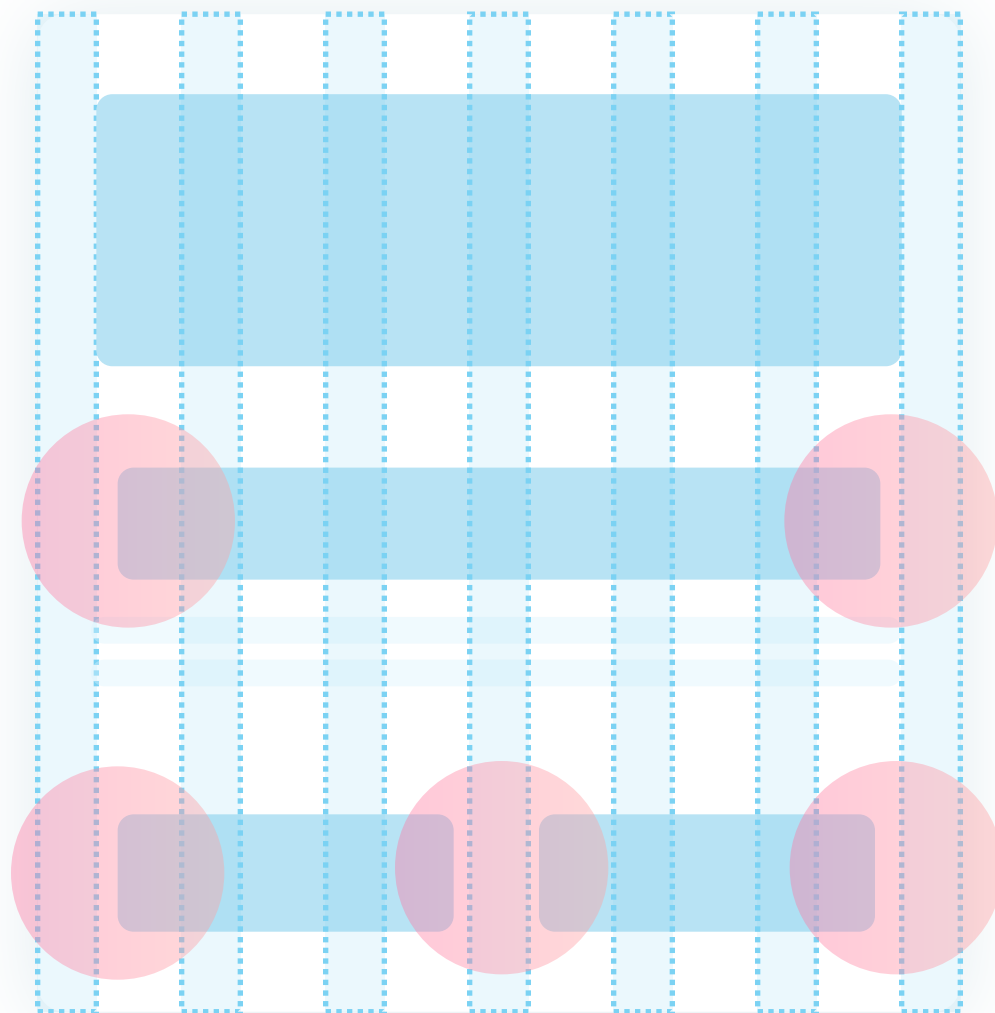
Button



Hover effect

LAYOUT AND GRID

If you have a consistent, defined grid (let's say 12 columns with 32p gutters), the first step is to apply it again to every screen. Then find all the places in which our layout doesn't match the grid layout. Take screenshots and write down the displacement amount.



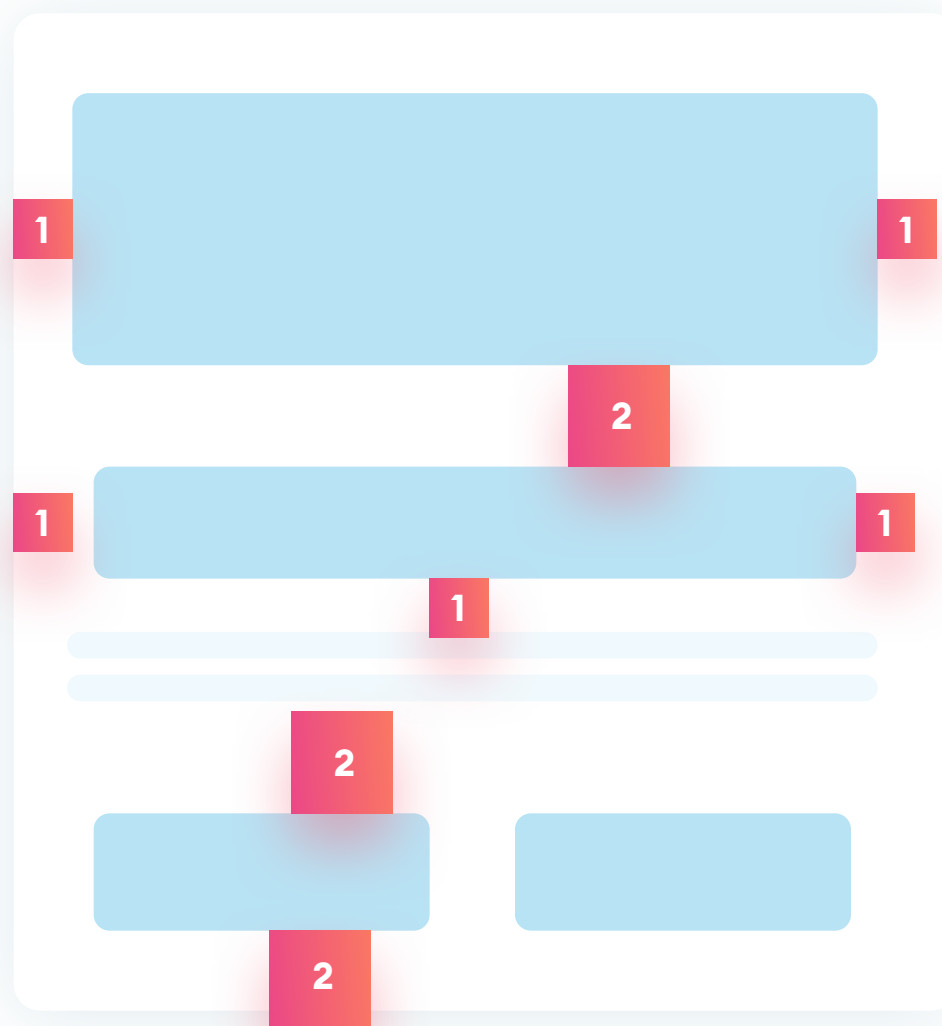
In this example (a 6 column grid), we see that one of the headers and two buttons below are off the grid. Write that down and proceed to the next screen.

If an element has a similar out-of-line position with the grid on more than one screen, it's likely the easiest one to fix. Please write it down on top of your layout audit list and take care of it first thing. In most cases, a global size or position setting will do the trick.

The best practice in implementing design audits is to focus on the easiest fixes first and then go down the line towards the hardest ones. This way will yield measurable results in less time.

CUSTOM, SOFT GRID

When you don't have a defined grid and it's not easy to find a structure, we can define a soft grid instead. The best way to do it is by using the "Red Square Method" which we cover in more detail in the grids chapter.



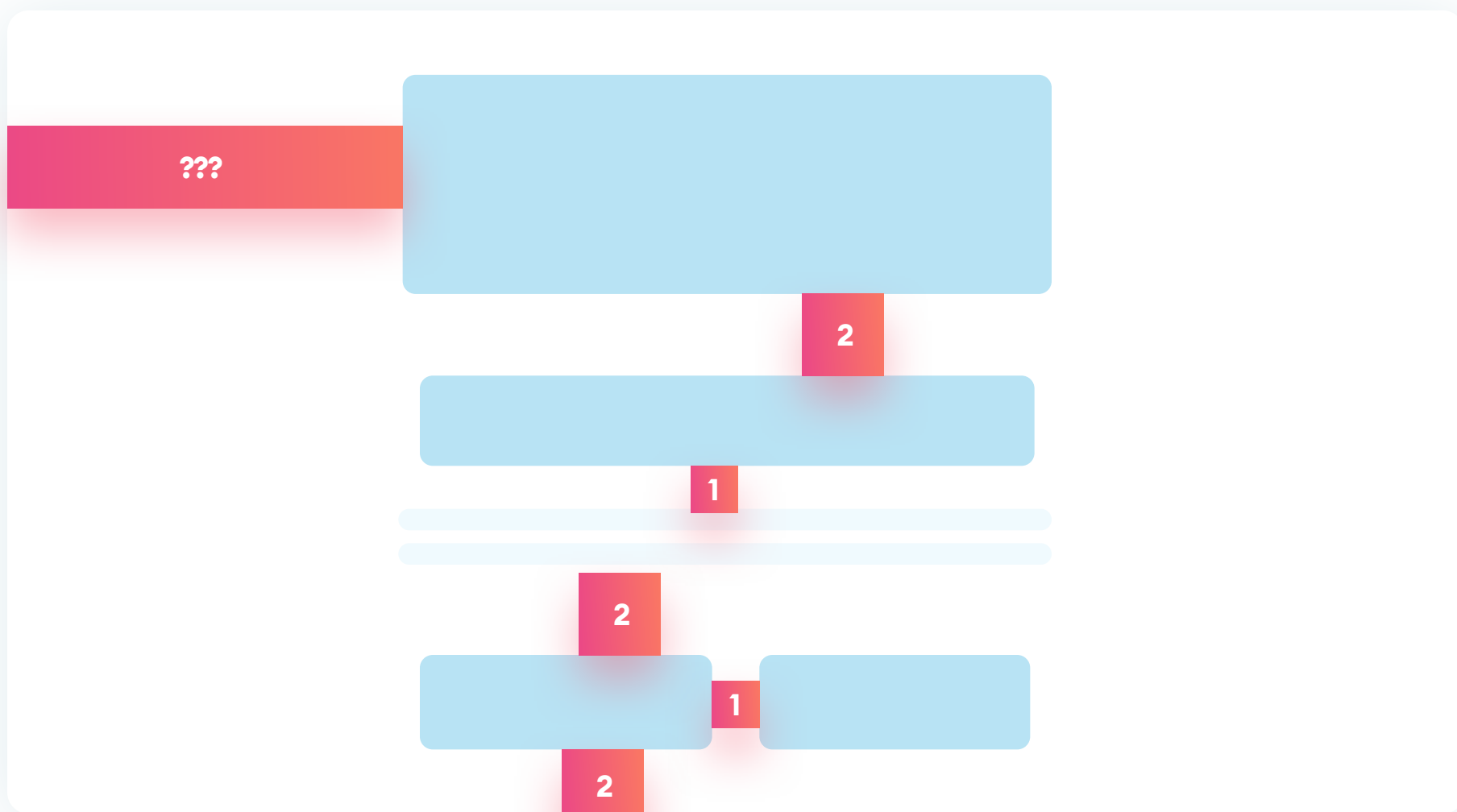
While defining a soft grid for a mobile app, start by measuring the most often used margin from the sides of the screen (box 1). If the number doesn't look random (like 11 or 13), we may have found our grid base. Values like 10,12,15,16, 20 or 24 are the most common.

Keep in mind the recommended value for mobile apps should be 16 and above. After finding the base, we look for the second, larger spacing value (2). If that number comes directly from the first one (like being a multiple of it), we have found a rule.

The example above shows some margins are uneven - especially on the right side and the bottom. Note those out and schedule to fix it.

SOFT GRID ON LARGER SCREENS

While designing for larger screens (like desktops and TVs), there's no point in finding the margin from the sides - it can be different for every device. In this case, try to find the soft grid by the spacing between the elements.



With complex designs, this process can take a significant amount of time. Try to use the most content-heavy screen to start, as it's usually easier to find a rule with more elements.

The most popular spacing should become your soft-grid base.

All deviations from that number (especially if they're only a few pixels off) should end up on the list of inconsistencies.

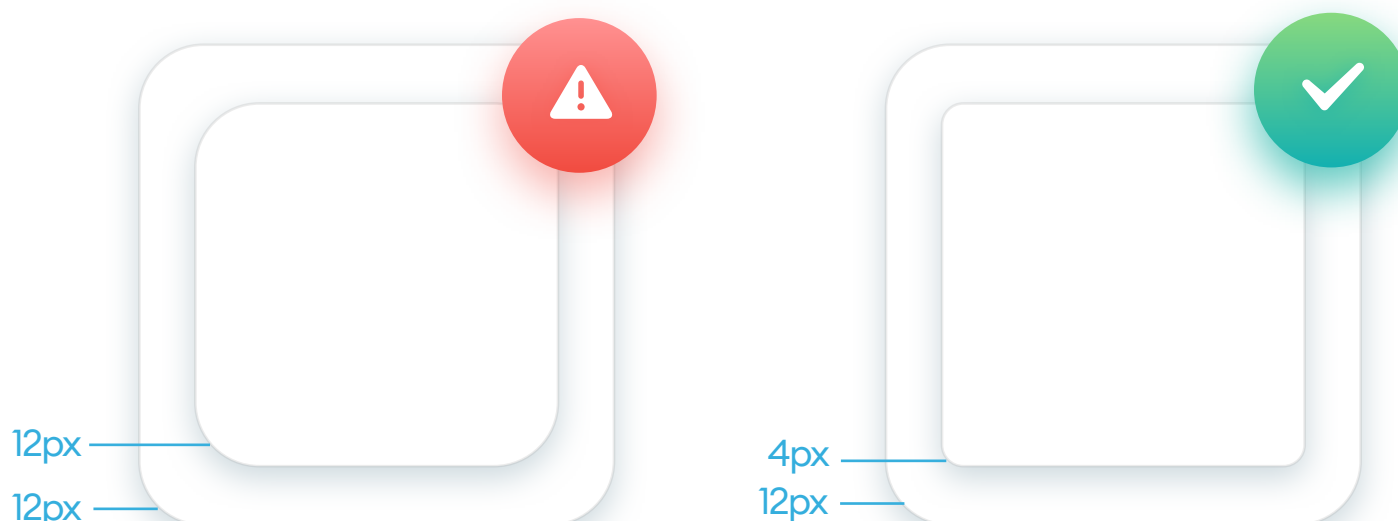
OBJECTS

Each object you design has a couple of characteristics that define how it looks. These range from the most obvious ones like background-fill, through border thickness, border-radius, shadow components (direction, alpha, blur), and more.



These rectangles all share the same background and border. They differ, however, with border radii (4, 0, and 6) and shadow values (different Y value and different blur).

Look at every text-field, checkbox or button, and write down its border value (thickness), border-radius (in pixels), and shadow values. The shadow is one of the primary hierarchy setting components so that it can have a more varied set of values than border-related ones. More than one border thickness and more than two radius values should raise suspicion of inconsistency (unless they come from a conscious design decision).



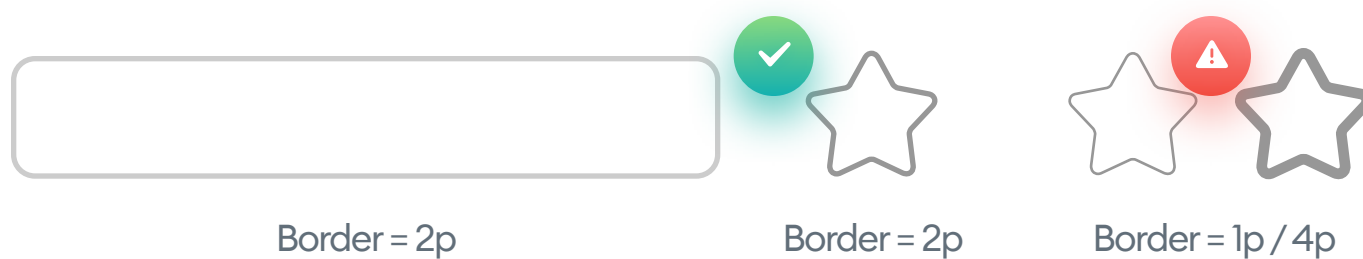
Two border-radius values are OK with nested elements. When both objects have the same radius value, the diagonal space between their corners is different than the area on their sides. Try adjusting the border-radius for the shapes to match more evenly.

OBJECTS

Border thickness is especially vital while designing forms. All elements of equal importance should have the same border weights. An active (focused) field can have a stronger border to emphasize it, but all the other elements should be consistent.



If you're using outline icons, you should also check their border thickness against your form fields. If your form uses 2p border widths, try to make your icon border width consistent.

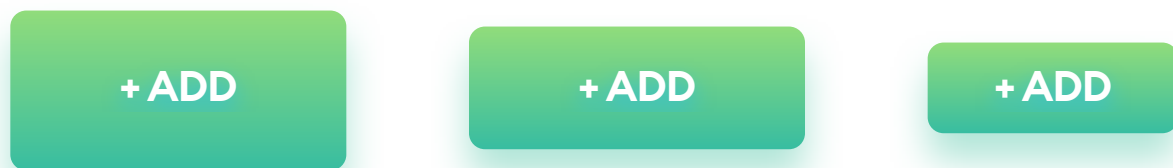


The border-radius of our form should also define the radius of our icon shapes. In this case, you don't need to count the pixel values but rather choose similar roundness of these objects. A rounded text field should always accompany rounded icons. With complex, multi-shape icons, you can't make every angle rounded, but it's the general icon shape that matters.



INTERACTIVE ELEMENTS SIZES

Button and form fields should have a consistent height value throughout the entire project. Try to find interactive elements that are too small first (smaller than 20p height on desktop and smaller than 40p on phones). The size test in the audit is a two-part process - finding elements that are too small (or too big) and finding items being inconsistent between one another.



The smaller the height of a button, the harder it will be to click on it (desktop) or tap on it (touch screen device).

Our project can have similar size buttons, with subtle differences between them or differ only with internal padding around the button label. As active elements are an essential part of any design, try to keep them as consistent as possible for clarity.



Various stroke-widths in radio buttons



Two icon styles in checkboxes



Small differences in button height

Different inner padding around the button label.

“

**Most common UI
problems, are often
the easiest ones
to fix.”**

ACCESSIBILITY TEST

Accessibility analysis is an important part of the UI Audit. It's good to leave it for after we have the list of most common font sizes, weights and our color palletes created. The simplest way to test accessibility is by using a contrast checker plugin for your design tool. There is of course a lot more to accessibility than just contrast, and we recommend you explore the subject in more depth. But contrast is a good start given the sizes and clarity of the interface are already spot on.

Every place in which your test elements don't meet the WCAG rules should be pointed out in the audit. (That's unless you have a special interface version for vision impaired users)

Example

 1.69

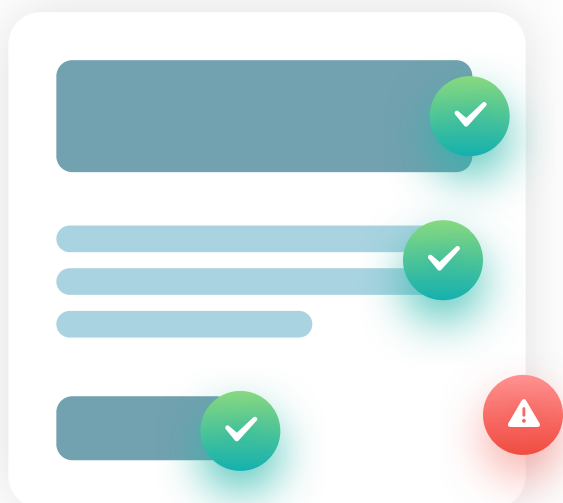
Contrast too low

Example

 13.53

Contrast in the right range

NON-ESSENTIAL ELEMENTS



If your design hierarchy is easy to understand it defines the relation between the elements. In that case decorative, non-essential objects can sometimes fail the contrast test. (Like light, soft-shadows under a product card). But keep in mind that it's only accepted, when all the card elements have clear enough hierarchy so the background is not needed to define its bounds.

Purely aesthetic objects (like low-opacity decorations in the background) can be skipped during accessibility analysis. Their visibility (or lack thereof) shouldn't affect the ease of use of the product.



Decorative background shapes can have low contrast.

While testing coded components try to check if all your images have an ALT value (alternative text). All form elements should also have all the necessary labels and placeholders that help with them being understood by voice interfaces.

Accessibility is a sophisticated, always-growing field so we suggest you explore it in more detail. In many cases doing it right makes the interface easier to use by everyone - not just the people with vision impairments.

“

**Auditing yourself
will help with
creating good
habits for future
designs.”**

UI Audit

DIY

Aside from the apparent quality improvements, a UI Audit also helps junior designers to find weak spots in their projects. The next pages contain a special checklist that serves as a base for auditing a digital product. You can print it out or copy it to your task management software.

This simplified checklist is also a great way to show the results of our audits to stakeholders and product owners. When you start your report with easy to digest findings, it will be much easier to convince the right people to act upon the problems and fix them.

The method works just as well for both work-in-progress designs and fully implemented products. In the second scenario, it's good to have one of the developers to be a part of the audit team.

GOOD PRACTICES

Try to make a habit of checking your every design. It'll help you avoid many of the same problems later on for a more consistent, refined experience.

Even if the eye for the right color or font choices comes with practice, consistency, contrast, and the right size of the objects in your designs will bring them to the next level.

UI Audit Checklist

Typography

 Consistent, defined typography.

 Too many styles, fonts and sizes:

fonts

sizes

styles

Interactive elements

 Consistent sizes, borders, shadows etc.

 Inconsistencies in objects definitions:

sizes

border radii

line weight

shadow styles

UI Audit Checklist

Grid / Layout

- All the screens are aligned to the grid.
- Alignment problems:
 - Desktop inconsistencies
 - Mobile inconsistencies

Colors and gradients

- The design uses only previously defined colors, gradients and gradient types.
- There are too many colors and gradients:
 - all project colors
 - base (defined) colors
 - gradients
 - gradient angles and styles



28.

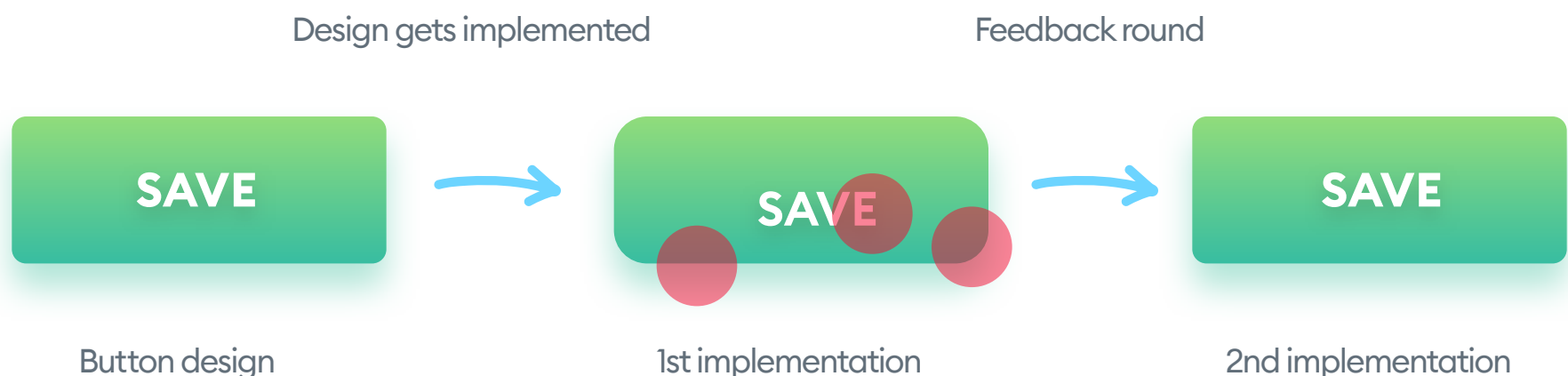
Developer Handoff

Developer Handoff

HOW TO WORK WITH THE DEVELOPERS

After carefully designing all our views and building the clickable prototypes, there comes the time to start the actual implementation. In many teams, this process means that the designer cuts out the assets (icons, images, etc.) and provides some basic guidance for the developers. Then they start coding the design, and the designer can focus on the next project.

We don't recommend this approach. Designers should play a vital role in the QA (Quality Assurance) process by evaluating the precision of the implementation, testing, and tweaking the microinteractions.



This chapter contains the best practices, useful processes, and some additional knowledge on how to work with developers. We also try to tell the story from the developer's side, so they can work with designers faster and more efficiently.

When we test a real, coded screen of our app or website on the actual device, some tweaks may be necessary. That's why collaboration between designers and developers is essential even after the design process is complete.

THE BASICS AND THE PROCESS

The development consists of two main parts (using a vast oversimplification for the sake of clarity):

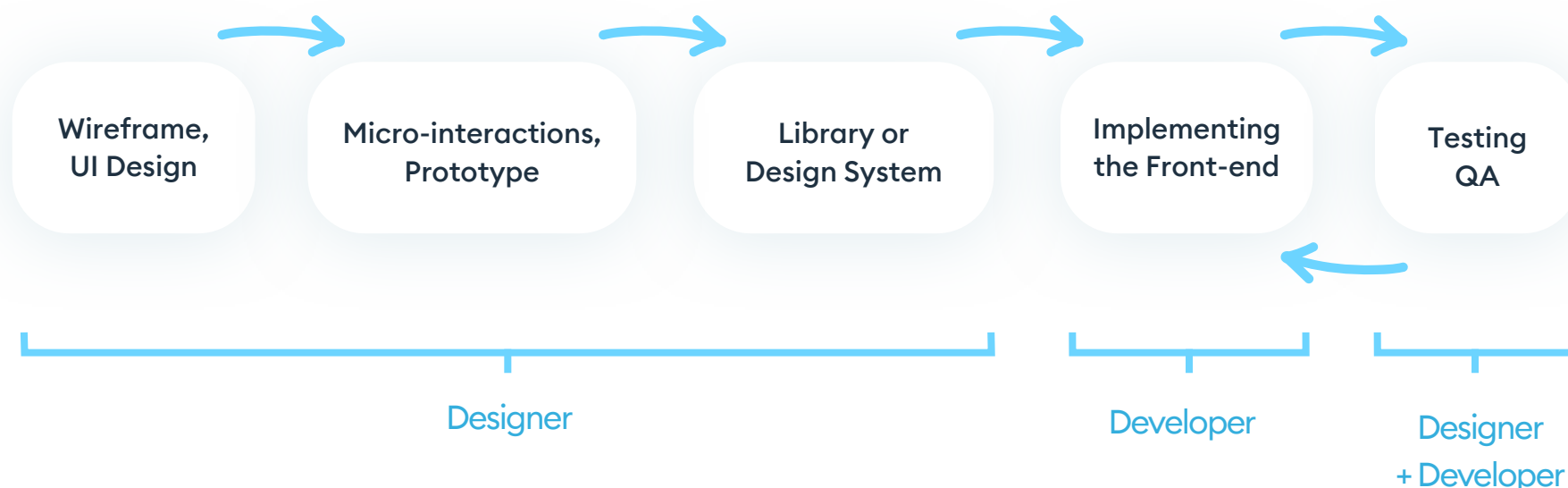
Front-end: This is about coding everything the user can see. It contains the UI, animations, transitions and all kinds of calculations that can happen in real-time)

Front-end developers should be the designer's best friends, as they're the people we work with the most during the implementation.

Back-end developers usually code things we can't see - the engine of our app and the processes happening in the background. They create the databases, connect everything, and make the product functional. They're often very far from any UI related issues. Still, sometimes their input is helpful, as they can provide valuable information about how design decisions can affect the performance of the engine.

A full-stack developer is simply a combination of the two fields above - someone who can do both front-end and back-end.


The team responsible for implementing a design, however, is always the designer and the front-end developer. A typical process can look something like this:



Even if you don't plan on building a full design system for your product, it's still worth starting with a Design Library. That approach (even with the most basic library) will significantly increase the precision and shorten the time required to implement it.

You can create your component library at the very early design stages. It's worth doing early, as it will make it easier for the designs to be consistent as well. Some prefer to do the library near the end, once all the types of components are already done. Choose the way that works better for you.

The design library is a separate file, with all of the components from the main project. That includes things like buttons, inputs, dropdowns, and larger items like popups and product cards. They all should be defined and annotated in the design library file. A good idea is also to add the CSS snippets next to each object (within your design file).



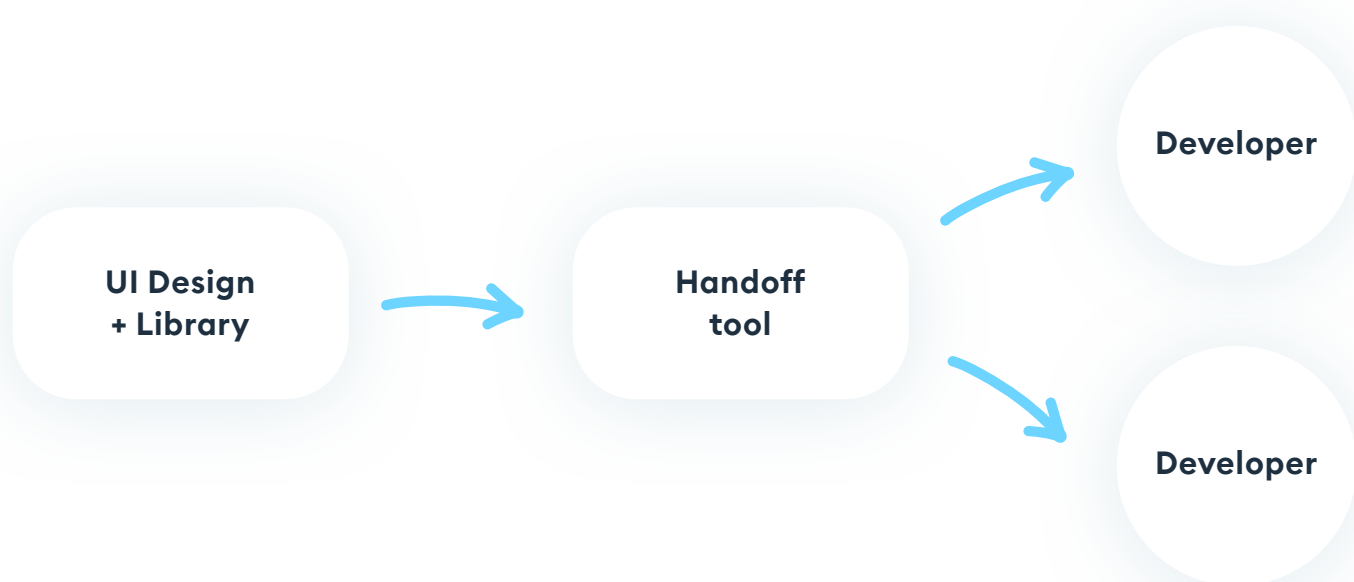
CSS of this button

```
background-image: linear-gradient(0deg, #13B0B0 0%, #F4FF51 100%);
box-shadow: 0 5px 7px 0 rgba(63,192,158,0.40);
border-radius: 3px;
padding: 0 24px;
margin: 8px;
font-family: Work-Sans;
font-size: 10px;
color: #FFFFFF;
letter-spacing: 0.15px;
text-align: center;
text-shadow: 0 2px 4px rgba(0,0,0,0.12);
line-height: 44px;
```

Most developers will optimize that code and won't use it in the form shown above anyway, but it's good to give them all the properties separately just in case they need them. You can get the CSS snippets for each object from within most design tools available today.

THE TOOLS

The next step is transferring both our library and all the individual screens into a developer-friendly tool. These include Sketch Cloud, Figma, Avocode, Zeplin, inVision Inspect, and more. Once these tools process your design files, they'll be able to get precise measurements of all objects, get their properties and relations, and more. These tools make the developer handoff process a lot faster and easier.



Most of them require you to drag and drop your design file, and then they can do their magic. Once the processing is complete, your developers can access all the project information in a developer-friendly form. That can include image assets as well. Because these tools are exact, if you mess up your grid, the developers will copy the wrong rules and code the entire project wrong. Remember to double-check everything before uploading.

If you need to make those 1pt adjustments do them and then upload.

You can also prepare the developer style guides yourself, without relying on external handoff tools. They make it easier, but together with your developers, you may come up with a dedicated process that works better for your specific needs. In any case, you need to be as precise as possible when designing and documenting your ideas.

“

**A designer that
knows CSS can
communicate
with the developers
a lot easier.”**

As we already mentioned in the chapter on Design Systems, we believe all designers should know CSS. It's not a real programming language, but rather a collection of properties and values that are easy to memorize.

Each style you define has a similar pattern of *property: value;*

background-color : #FFFFFF ;

Property Value

The easy thing about CSS is that most of the commands are in plain English. The example above defines a background color for our object to be white (#FFFFFF in hex). CSS also understands some primary colors by their English names (white, red, orange, black, blue, etc.), but it's best to use the HEX or RGBA values for greater precision.

Most design tools provide you with entire CSS classes for all the design objects, so you can quickly learn by simply creating an object (like a rounded rectangle with a gradient fill and a shadow) and checking how the CSS for it looks like. Knowing CSS will make it a lot easier to communicate with the developers and will give you a background on how the UI is defined.

DISPLAY : NONE;

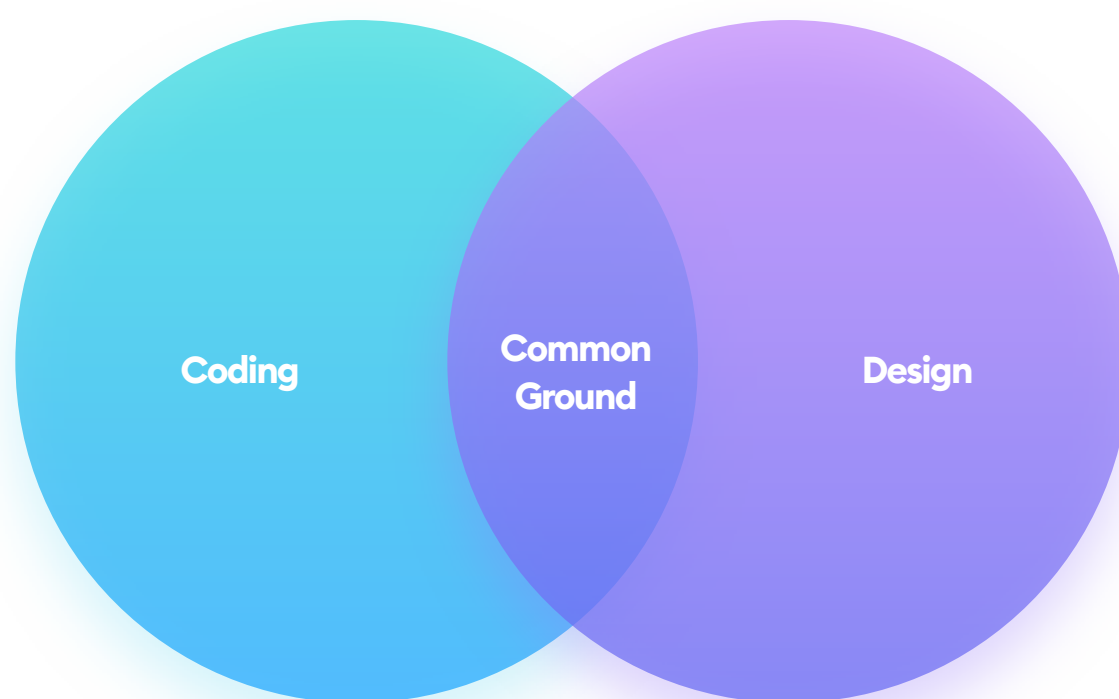
One of the most common front-end developer jokes is "fixing something with a display: none;" which, of course, means hiding the object from view. They often joke that way about the most complex problems that they need to solve, so now you know why everyone is laughing :)

DESIGN BASICS

When (as a designer) you learn the basics of CSS, it's worth to convince your developers to learn the basics of design as well. It will make your cooperation a lot easier and will help you eliminate many of the most common problems front-end developers face nowadays (mostly alignment issues.)

When the developer can understand, that the design is not a work of art, but rather a precisely planned set of rules, it will help him transfer that into code with higher accuracy. The fewer QA rounds needed, the faster (and cheaper) the project is to build.

Try to communicate the design process as a language, so they can learn it just like they learn programming languages. The design also has rules and methods and won't work if only randomly put together, just like code.

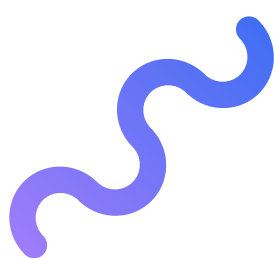


A developer who understands design and a designer who knows a little development is the best combination for creating impressive user interfaces. You need as much of that common ground as possible, and then you won't believe how much faster and better everything works.

ASSETS

Most of the simple shapes (like buttons, carts, input fields) can be generated directly in code. Even more complicated effects like double, colorful shadows or custom strokes can now come to life with good CSS alone.

All non-standard shapes, icons, backgrounds, or photos are assets that we need to provide our developers separately. There are a couple of possible options on how to handle assets. Let's dive in.



SVG



PNG



JPG

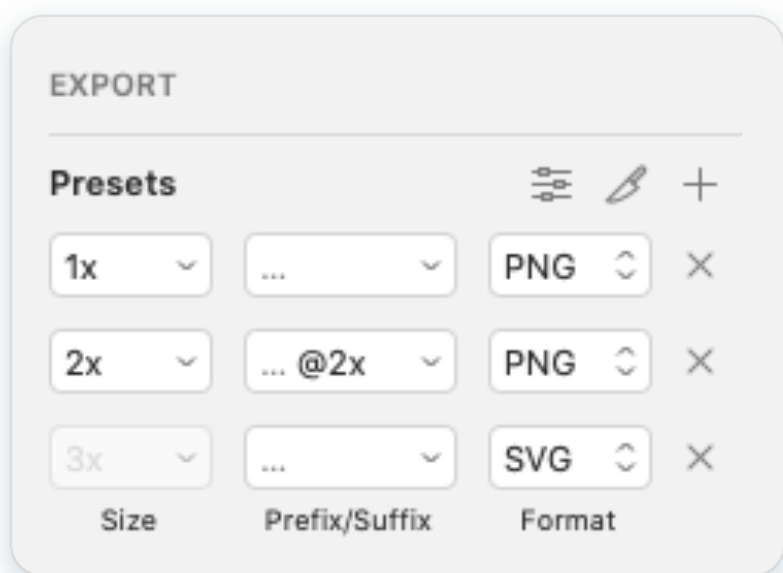
Simple, flat shapes or icons work best in SVG. It's a vector format that's web-ready and allows for easy scaling of objects without any loss of quality.

Use PNG's for more complex images with transparencies. If you're designing for high-density displays (and trust me - you likely are), also remember to make an @2x (double the size) version of your file. It will make the image a lot sharper on newer, more modern screens.

JPG's are still the best choice for most photographs. In cases where the photo is quite "flat" and only has a couple of colors, you can try experimenting with PNG as well. Remember to find the right compromise between loss of quality and smaller file size.

CUTTING OUT THE ASSETS

Most modern design tools provide you with the ability to export assets in various formats and versions at once. You can export an SVG, 2X PNG and a JPG in just one click.



Sketch (v. 60) Asset export panel.

You can use presets to create your own combinations (like 1x svg, 2x png) and export them with one click. The slice icon (knife) allows you to create areas covering all the layers inside them.

The essential part of this process is making sure the resulting files are small in size (kilobytes). That, of course, comes with some quality loss, and it's your job to pick the right compromise. The lower the file size, the faster and smoother the coded design will work. Before you hand over the actual asset files, you can optimize them manually.

Optimization often means compression (which results in some quality loss), removing unused colors or meta-information (like the EXIF data which stores location and camera information for a photo and is not necessary.)

Some developers optimize the assets themselves. We believe this is a job for the designer, as sometimes too much compression can negatively affect the interface. There are various online and offline optimization tools. The ones we use are TinyPNG, Compressor, ImageOptim, and ImageAlpha. Most of them are free - you can find them online.

↩
Proto

to
typ

29.

Prototyping

ping

Prototyping

PROTOTYPING BASICS

Interactive prototypes are a simulation of how the finished product will work. There are two main types of prototypes:

- code-less (inVision, Sketch, Figma)
- fully coded (UXPin, Framer, HTML/CSS prototypes, Axure)

The code-less prototypes most often rely on image-maps, an old HTML technique of creating invisible links on top of static images. Those links lead to other images, giving an illusion of a working interface. In most of these, you can only interact with them through links, so all input or select boxes will become just a static representation.

The fully coded prototypes are a lot more advanced, as they have most of the objects built-in code with full interactivity. In some cases, you can also apply additional logic to them or even use external data sources like Google Maps, YouTube videos, or music players.

They can have full, validating forms and be almost indistinguishable from the final product. In some cases, after tweaks, they can transition into being the front-end components of the finalized app.

Prototypes also used to be divided into **low-fidelity (wireframes)** and **high-fidelity (visual UI designs)**. Still, given the growth of design systems and componentization of design, lo-fi prototypes are very rare. They're also not that good for testing purposes, as low fidelity gives your users a feeling they're not using a real product, and that can affect the test results in a big way. That's why it's best to stick to high fidelity prototypes for most projects.

WHY SHOULD YOU PROTOTYPE?

When designing new, exciting features for a product, the "how" of the entire process is mostly in your head. Sure, you can document it, but that can lead to the famous "game of telephone" problem:

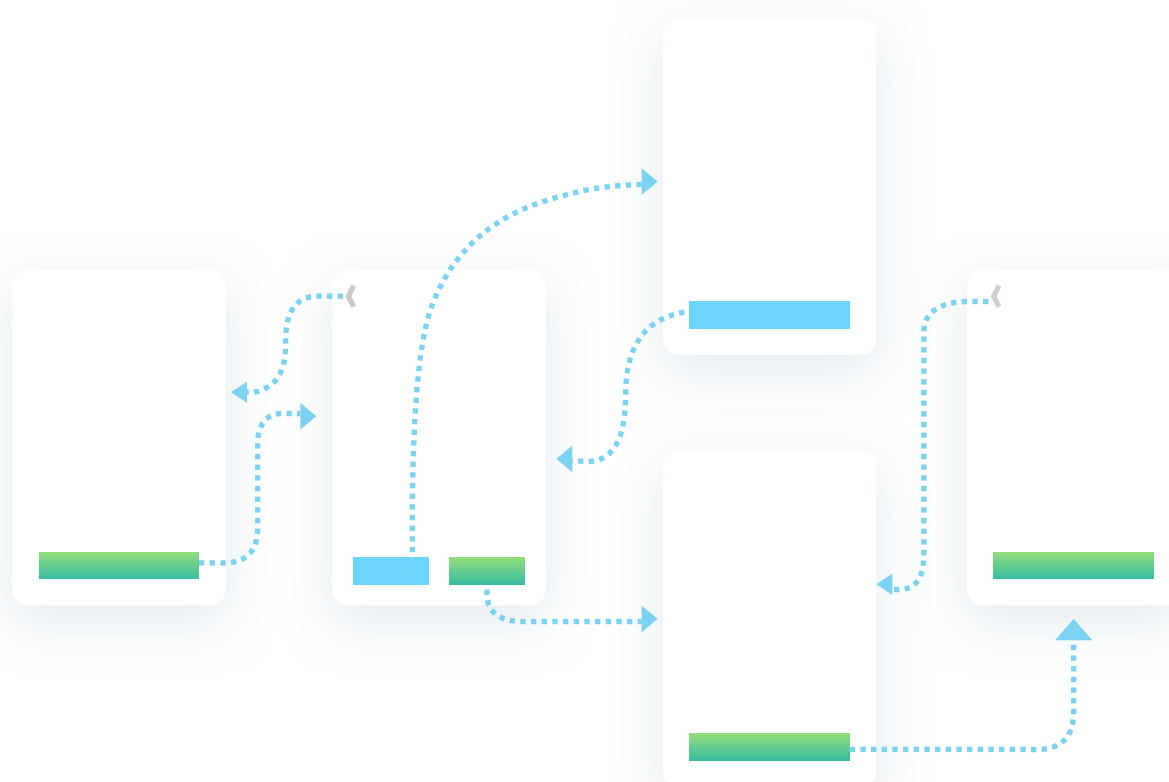
not everyone understands you the same way. Messages get distorted and skewed. Whether it's stakeholders or your developers, having a "clickable" prototype is going to be a massive help for practically everyone on your team.

If you're practicing speed-designing, (a technique of rapid iterations), a prototype can help you identify potential problems and choke-points a lot faster.

FLOW DIAGRAMS

Most code-less design tools display prototypes as flow-diagrams. The idea is to connect screens with individual components by a series of arrows.

A flow diagram can also help give a birds-eye view of the navigation to your developers, so it's a good practice to add it to your project documentation along with a clickable prototype.



“

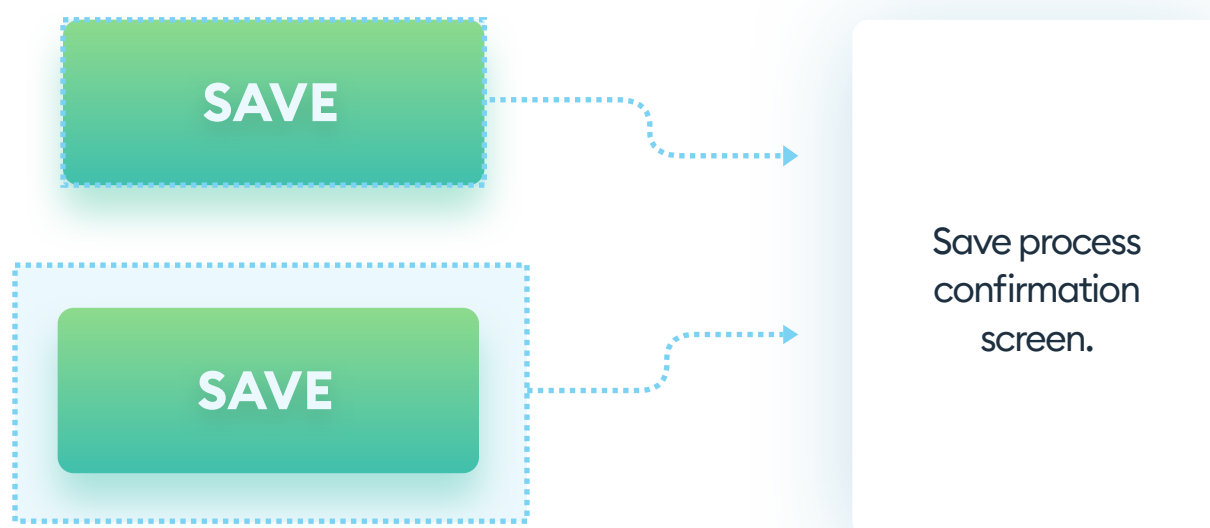
**A good prototype
is worth a thousand
meetings.”**

Ideo

CODE-LESS PROTOTYPES

Code-less prototypes are very fast to build and a great way to show the product to stakeholders while the design process still goes on. There is a quote, attributed to the company IDEO, that one prototype is worth a thousand meetings. That is true. It's a lot easier to show someone how a product works than to explain it in writing. It all started with paper prototypes, where you connect screen representations and create a non-digital flow diagram. The natural evolution of this approach was the code-less prototypes that inVision has brought to the masses. The idea behind them is to use image maps linking objects on a screen to other screens.

You can either link an object directly - for a button it would only use its size as the clickable area, or draw the image map yourself and make it larger. From our experiences, the second option, while slower to build, allows you to make every active area a lot bigger, which will help with ironing out the initial navigation problems. The first option, however, is closer to reality and can lead you to update your touch target sizes if things turn out to be difficult to click.



You can either link an object to a screen directly or draw the linking rectangle yourself. No matter which option you choose, the result will work precisely the same way. The second choice gives you additional precision for linking and can allow making the touch areas larger.

CODE-LESS PROTOTYPES

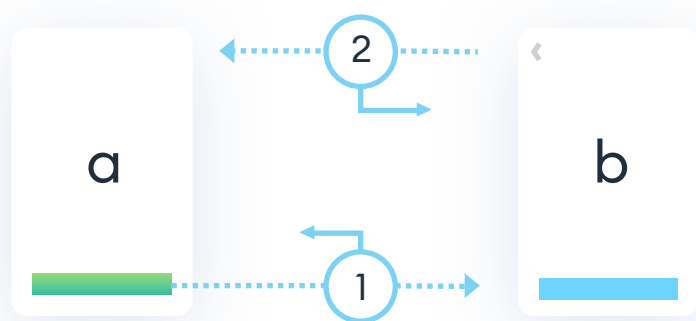
Most tools allow you to create the web-based prototypes without leaving the tool itself. Others use special plugins (like Craft for Sketch to sync with inVision seamlessly). That makes the connecting and linking process so quick. It'd be a shame not to do it.

Aside from connecting screens to each other, you can also select an action for "back" that always takes you to the previous screen. It's especially handy when there is more than one way leading to a particular screen, and you want to keep the flow intact.

TRANSITION TYPES

When just sketching out the flow of your product, you can stick to the default transition types. It's excellent for internal testing to have the "instant" transition where one screen replaces another.

Before testing on real users or stakeholders, it's good to pick the appropriate transitions. The general rule is for the transition to show where each screen exists in space. So if you're going to the next screen in a series, it should move in from the right side. When going back, the back arrow is usually on the left side, so the previous screen should slide in from the left side.



When transitioning from screen **a** to **b**, using transition 1, the actual movement of the screens is opposite to the transition arrow. Screen **b** slides in from the right side.

When going back (transition 2) from **b** to **a**, screen **a** slides in from the left side to show it's been there, behind the display the whole time.

Do those transitions right, as they can cause a considerable amount of confusion, especially when they are opposite to what the user is expecting. We like to understand the order and relation between screens, and transitions are an excellent way to enforce that relation.

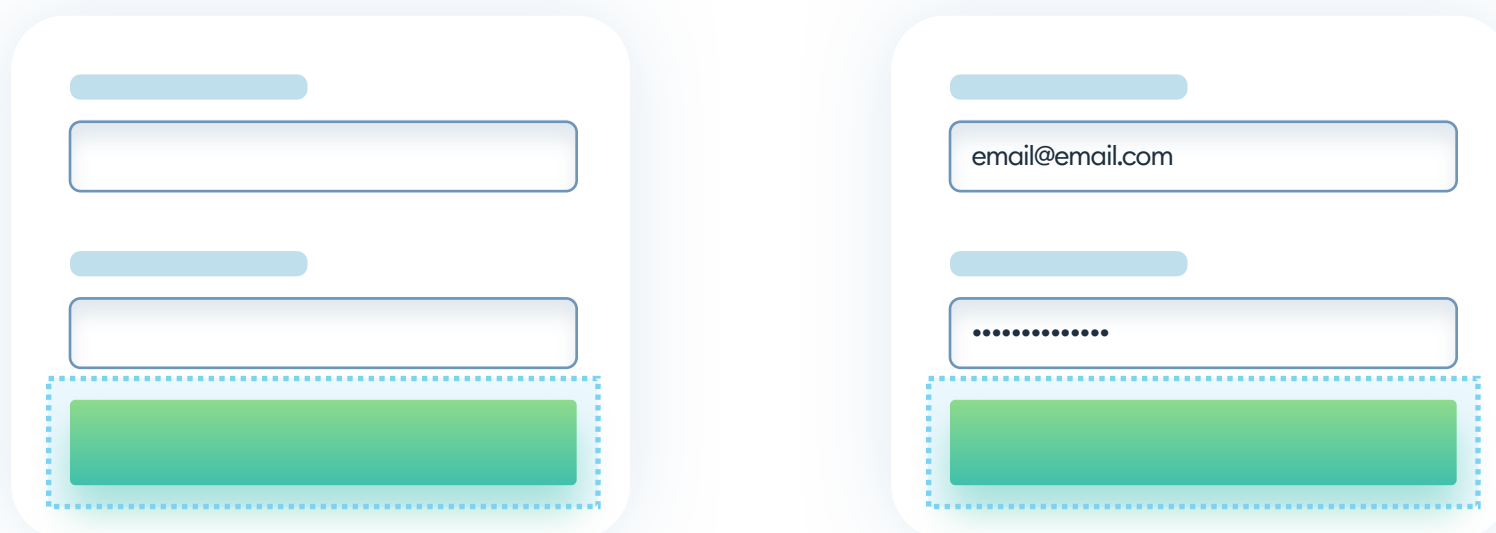
CODED PROTOTYPES

Coded prototypes are a lot more challenging to build, as they require you to code all the components, interactions and transitions. Whether you create them in tools like Framer or UXPin or put together some HTML, CSS, and JavaScript, the general idea is to have an unprecedented level of interactivity.

All forms should validate, all interactive objects should cause the desired effect, and all actions should have direct, micro-interaction based consequences.

The aim is to be able to take those already pre-built high fidelity components and use them in the final product. It's blending the line between prototyping and front-end development, and with design systems, we believe this may eventually be the leading approach.

A design system based prototype is a lot faster to build than coding every little object from scratch. They're easy to organize, consistent, and expandable.



Code-less prototype (left) only has one active area - the button. You cannot enter anything into the text-fields as they're merely a static image. A coded prototype (right), on the other hand, is, in essence, a real form (often with validation) so you can fill all the fields and then press the button.

BLENDING COMPETENCIES

While code-less prototypes are easy to do for anyone, the shift to coded ones requires a small competence shift in designers. We often advocate at least learn CSS if you're a designer, and in many cases, that's enough to help the developer create the working prototype.

Front-end development is so close to modern design nowadays that learning a bit of React, or just plain JavaScript can be an excellent way for you to improve the quality of your work.

This is why designers should understand front-end, while front-end developers should know design. That seamless collaboration allows for much higher fidelity and robust prototypes, which leads to better user testing and superior end-products.



Once you establish a design system, choosing a pre-made button component will be quick and easy for the developer and will make the prototyping process much faster.

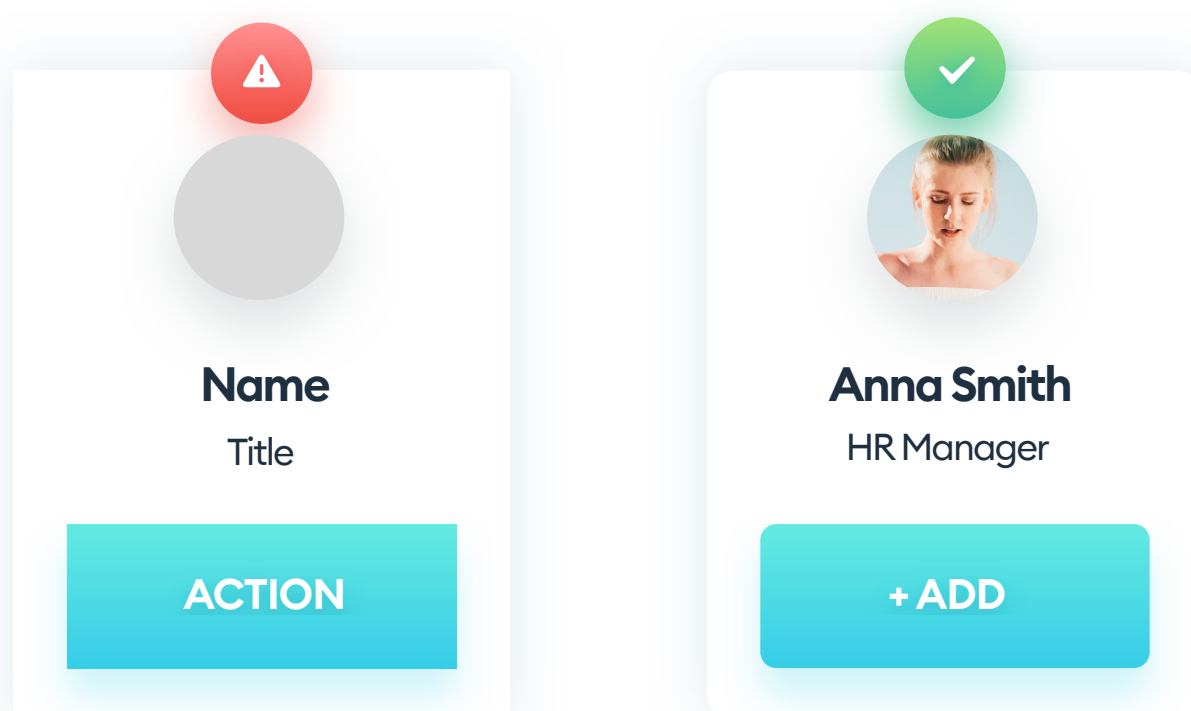
Treat your coded prototype like a door to the final product front-end. The best teams consist of design-driven developers and designers who know a bit how to code. In most cases, one or two people from each category are enough at this stage.

The important thing is to approach this with all the seriousness of a full product. You also need to test thoroughly both internally, and when possible, on real users.

REAL DATA

When creating prototypes, avoid Lorem Ipsum and dummy images. If you're showing a list of users, give every person a name and a photo. There are handy generators that can do that for you.

By "real" data, we don't mean you need to hunt down your future users and force them to fill the prototype with information. What you need to do is to simulate real data, so when a list item is a first and last name, don't write "First Name / Last Name" but rather create a virtual human to fill that spot.



Don't duplicate the same person on the list, either. The whole idea of working with a prototype is to have it reflect reality as closely as possible.

Try and use the longest name and the largest numbers in your prototypes as well (at least in one place). That way, you'll be sure how it looks when you transition between screens. Is it still readable?

Or is an adjustment needed?

“

**Use "real looking" data
in your prototypes.”**

RESEARCH

There are two main reasons for prototyping. One is to make the stakeholders (like the CEO, or investors) understand how the product works. The other is to have a model of the product, on which you can conduct research. It's best to test on real users, but even internal, office-testing can bring forward many issues you were not able to predict while designing.

There are many research methods you can use, ranging from in-person interviews, screen recording, or heatmap generation.

Some of those methods can help you identify flow problems early on - judging by how long it takes the user to accomplish the main task.

Build your prototypes in a way that you can gather the research data, quickly adjust the prototype, and test again. This approach is crucial for a functional, problem-free interface at every stage.

Avoid using dummy data and wireframe-like looking prototypes while testing, as the users don't treat wireframes the same way they do a real product.

The whole concept behind prototyping is to be quick and agile.



It's best to iron out all the possible problems before they go into the development of a fully functional product. Pulling back on that dev work can be both costly and time-consuming.



30.

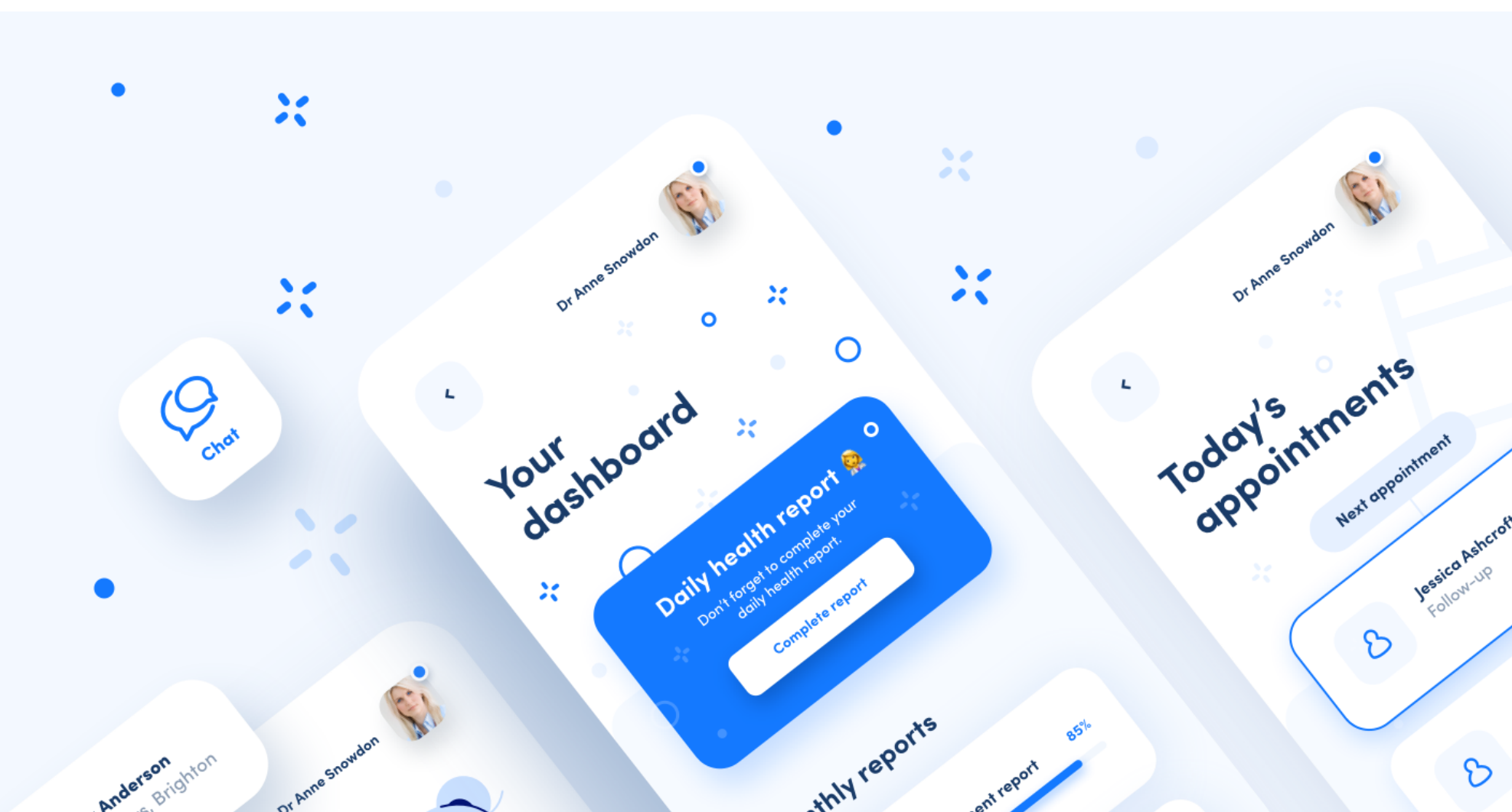
**Show your
work**

Show your work

THE POWER OF PRESENTATION

We are living in a very fast and noisy world. As of today, there are 14 Million shots on Dribbble. That means that 14M projects compete for the eyes and attention of viewers. A big part of success nowadays is having the ability to sell your work. And by sell we don't necessarily mean a transaction in which money changes hands. Selling is also getting people interested in you and in what you do. Selling your work can land you a great freelance client, or a great, steady agency job.

We still believe that great work will speak for itself, but sometimes it simply needs a little boost to be heard over all the noise. This chapter will cover all the main elements of presenting your work in a beautiful, eye-catching way. The idea is for your projects to shine, so you can be more successful at what you do best. Obviously the first and most important step is to have a great design to begin with, but let's assume we already have that part covered. So where do we start?



PEOPLE BUY WITH THEIR EYES

We all know the old saying that you can only make the first impression once. This is just as relevant in the design world. This can be the difference between a potential client closing the tab with your portfolio after a few seconds, or scrolling through it's entirety a couple of times and as a result getting in touch with you.



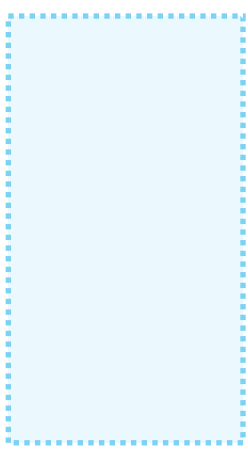
Now, keep in mind, that if you're an UI designer that also does the broader UX work (like some basic research, prototyping, flow-diagrams etc) your portfolio should reflect all that. It should show all of your skills. It's not only about presenting some app screens in a nice way, but having a general, consistent presentation of your work. It also needs to have some form of "user delight" that's deliberately yours. In other words it needs to be both beautiful, and unique enough in some details that you are going to get remembered.

“

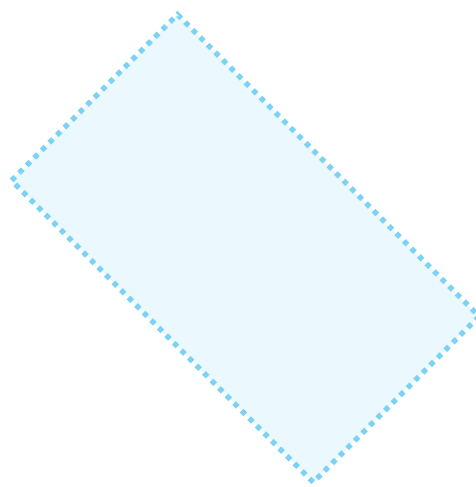
**Every great UI design
needs a great
presentation to reach
a wider audience.”**

WHEN IN DOUBT - ROTATE

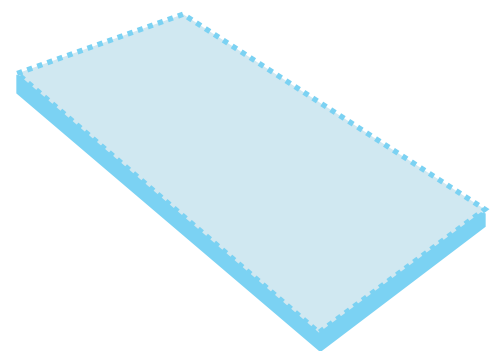
When actually using a design, the best possible orientation is to have the screen facing us directly. But that is not the case when you're browsing images. Both rotation and 3D perspective are huge factors that influence how good we perceive a project to be. There are three main static ways to show off a design.



"Flat" rectangle



2D rotation



3D transformation
& perspective

Just showing off a flat, rectangular design could still work well, if we give it a little bit of context or a nice background.

But even a 2D rotation will actually make the project look a little bit more appealing for most people. Adding a well defined perspective can make an even better impression, but for it to work the design needs to be done in a way that works well in a 3D perspective. That means the diagonal lines resulting from the perspective transformation should not be distorted in any way or pixellated at the edges.

Rotation and perspective are also a little bit more forgiving to some (small) grid or alignment problems in the actual design. By changing the perspective these problems are less likely to get noticed.

CONTROLLED CHAOS

When you're designing a digital product, user focus should always be guided and enhanced. A chaotic design will result with the attention shifting endlessly between buttons, form fields, images, icons and navigation without a clear, guiding pattern on where to start. This is bad. However, a presentation of the design doesn't have to strictly follow the same rules. That, of course, doesn't mean that total chaos is allowed. There needs to be some sense and reason to the presentation too. One of the ways of using controlled chaos is the popular grid of screens. Instead of showing just one or two shots from the project, you show a lot of them in a 2d structure. It's good to add a little bit assymetry to the layout, which will make it interesting.

A grid view can also be rotated (the most popular angles right now are between 9° and 13°) which is currently one of the hottest trends in presenting digital products. The best thing about this approach is that it shows both the scale of a product, and how you designed various functionalities and screens, all in one image.



“

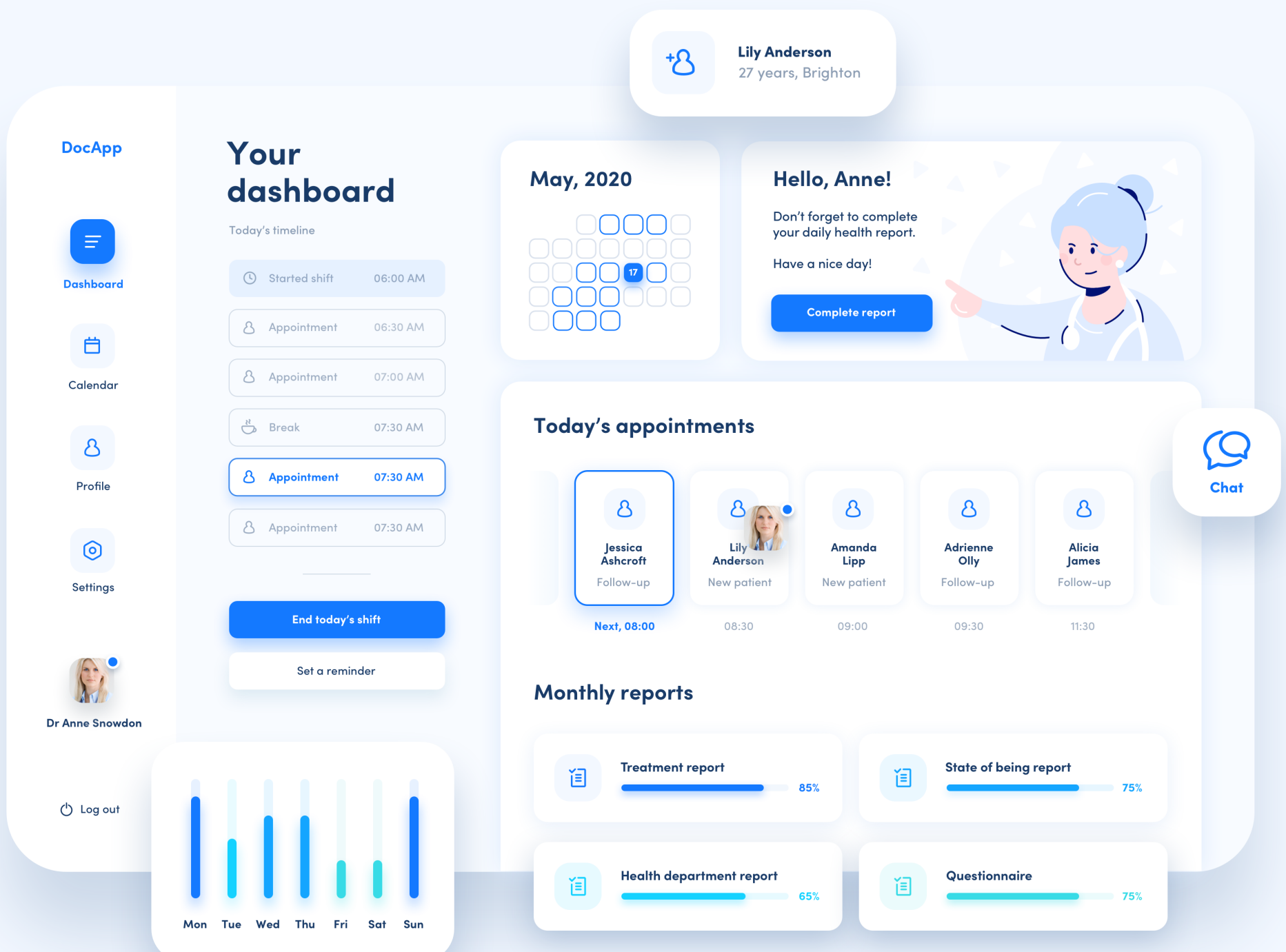
Both 2D and 3D rotation have a huge effect on how well a project is received.”

DEPTH AND ORNAMENTATION

An illusion of depth and a showcase of the functionalities can be a great way to create a good first impression. This can be done on a single image, or on a grid.

The two main approaches are to either add depth to some of the screens (in a grid), or create additional overlays that extend beyond the edges of your screen. This can be as simple, as showing more of a card carousel, that would normally be cut-off by the screen edge.

But you can get creative and also put notifications, or even some single icons on small overlays and place them on a layer above your design with the right amount of shadow to show depth.



DYNAMIC PRESENTATION

Unless you're designing for print or for a PDF, there's nothing stopping you from trying motion in your design presentations! They attract attention and also give the impression that a lot of work has been put into the project. Given how easy many animating tools are right now, that is not really the case most of the time, but it's definitely something we can use to put our presentations on the next level.



One idea is to keep the screen intact and animate the device itself in 3d space. You can do it in tools like Rotato Camera. It allows you to set up the keyframes and it will animate the phone movement into a beautiful shot.

Obviously, the video flows full of nice interactions and transitions of a product placed on a 3d-perspective will have the most impact, but they are also the most difficult and time consuming to prepare. You can however use simpler forms as well.

One idea is to simply animate the screen grid assembling itself, possibly with some background shapes moving in parallax as well.

You can also add some extra moving layers on top, like the carousel cards, or a profile card that slides into place on one of the screens. There are no limits to creativity here, but keep in mind that these animations should not exceed ten seconds. They need to briefly catch the attention, showcase the project and let the viewer go on to the next project.

“

Experiment with the presentation style, but always keep it "on brand" with your personal touch.”

PRESENTING NON-UI WORK

When you worked on a project, that had UI handled by someone else, you can still use UI / Visual principles to design a beautiful experience around the non-visual UX processes you did.

One example is visualising stakeholder interviews as a card layout with a description next to it. You can put quotes from the example stakeholders on each card to show that the process had actual, actionable insights and outcomes.

The same goes for user testing results, workshops and other parts of the process you took care of. Just use the visual principles to present parts of each process based on this chapter, and have a short description with more details.

Stakeholder interviews

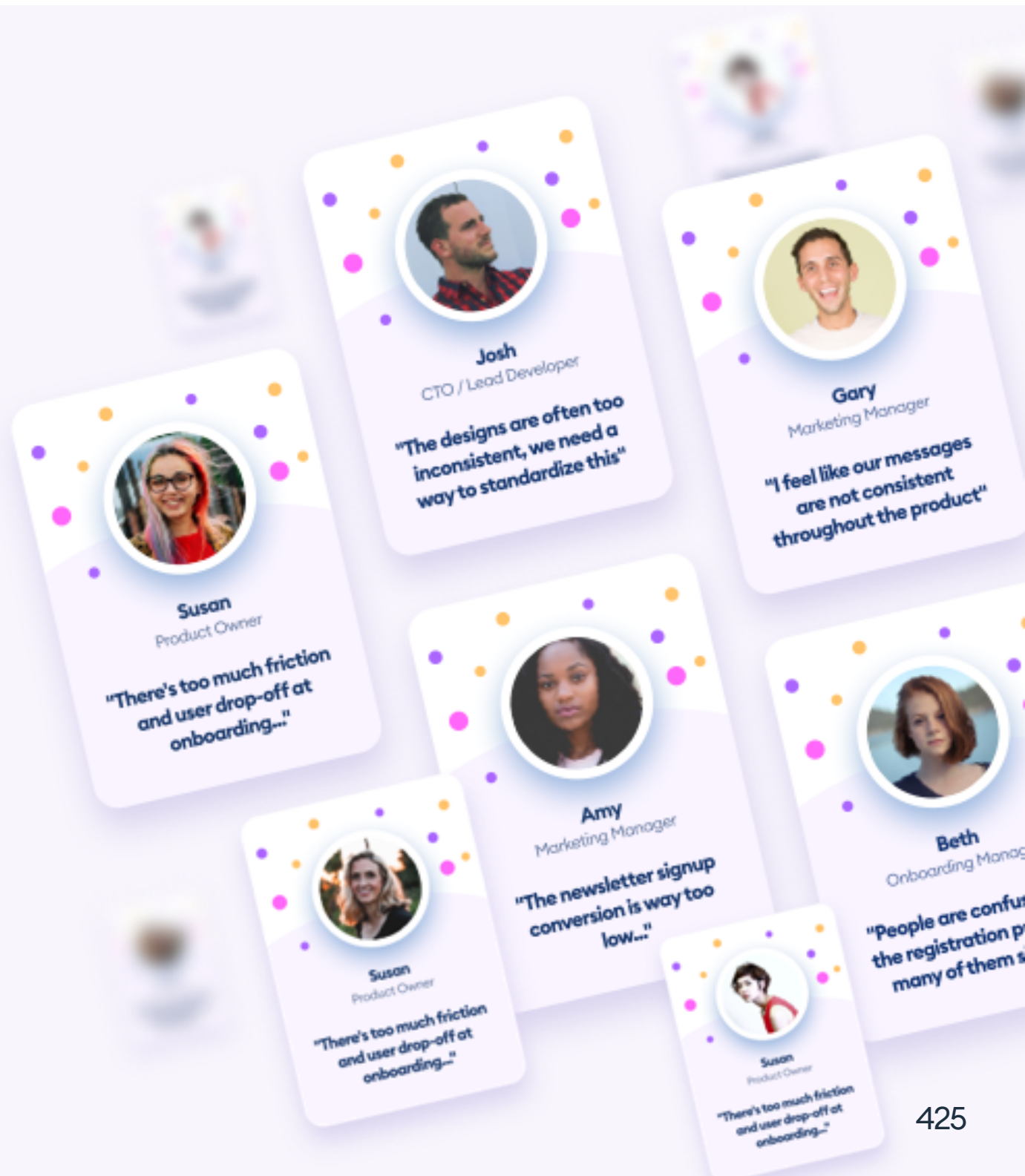
2 weeks

In person

via zoom

I conducted a series of stakeholder interviews over a course of two weeks, trying to find out common identified problems with the product and specify which areas are the most important to which group.

The goal was to find overlap and identify the most important goals for the first design sessions with key stakeholders.





31.

**Starting a
new design**

Starting a new design

THE BASICS

When you're just starting out, it's good to do the "project init" screens from scratch. That will give you the practice and confidence you'll need at that early stage. Many junior designers are scared to begin a project, with dozen of questions going through their heads.

- What resolution should I use?
- How high is the tab bar? The status bar?
- How big can the font be in a mobile app?
- How small can it be?
- Can I have active elements smaller than 44 points if they're not as important on mobile?
- How do I set the basic grid for a desktop website design?

And a lot more. But they all translate to: How do I start?

In this chapter I will help you create your own starting templates and define the good practices when starting a new design project from scratch. These rules are universal, and you can use them in any design tool you like, as they're not limited to just Sketch, Figma or Adobe XD.

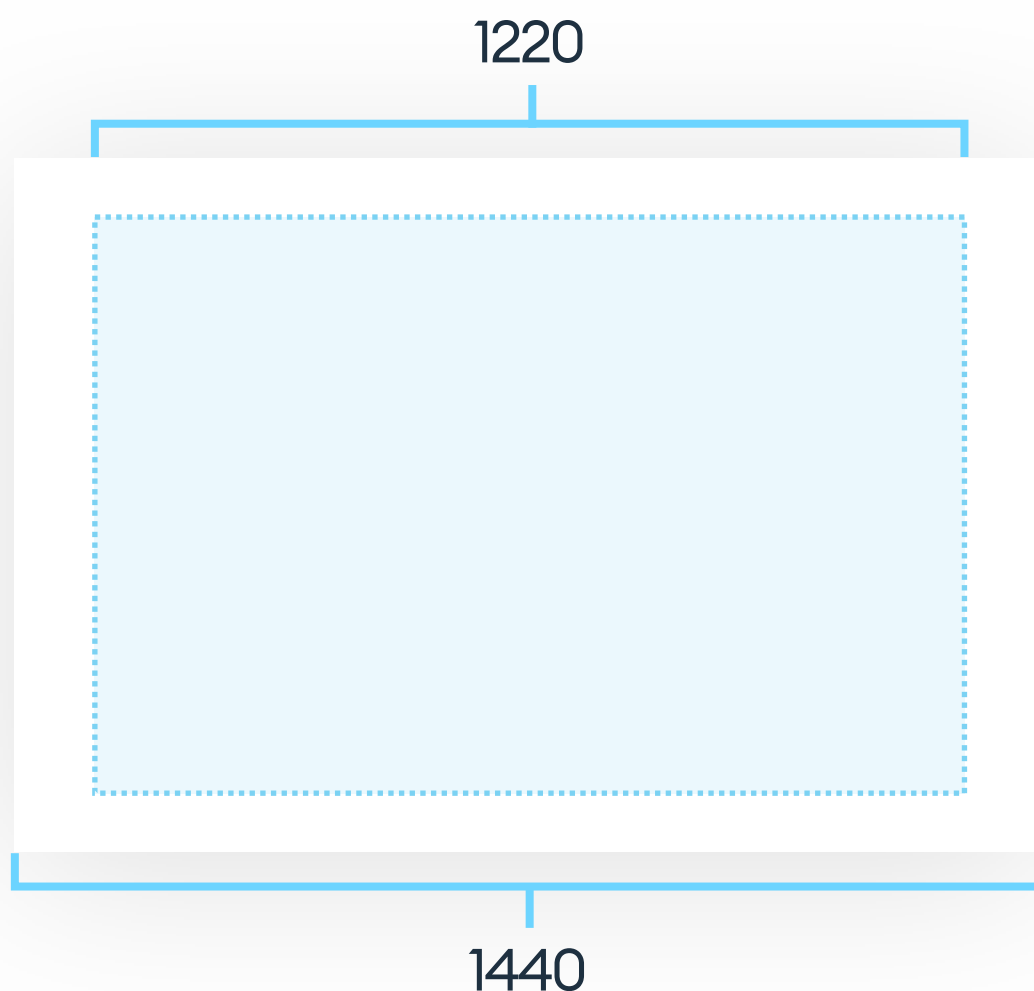
I will cover both an app design process, and a website design, including setting safe grids for landing pages and dashboards.

If your first designs will be portfolio projects and you haven't decided what to start working on first, start with mobile apps. The smaller canvas makes it slightly easier when you're a beginner to achieve a satisfying result.

Let's start with projects for the web, as they're a little bit more complicated. If you're designing a dashboard or a landing page, you need to start by choosing the proper artboard size.

There are many suggested sizes out there, but looking at the most popular screen resolutions I think the best choice for beginners (especially working on a portfolio project) is 1440 points wide. Obviously the height of the artboard will depend on how much content you're planning to show and can range anywhere from 900 points up.

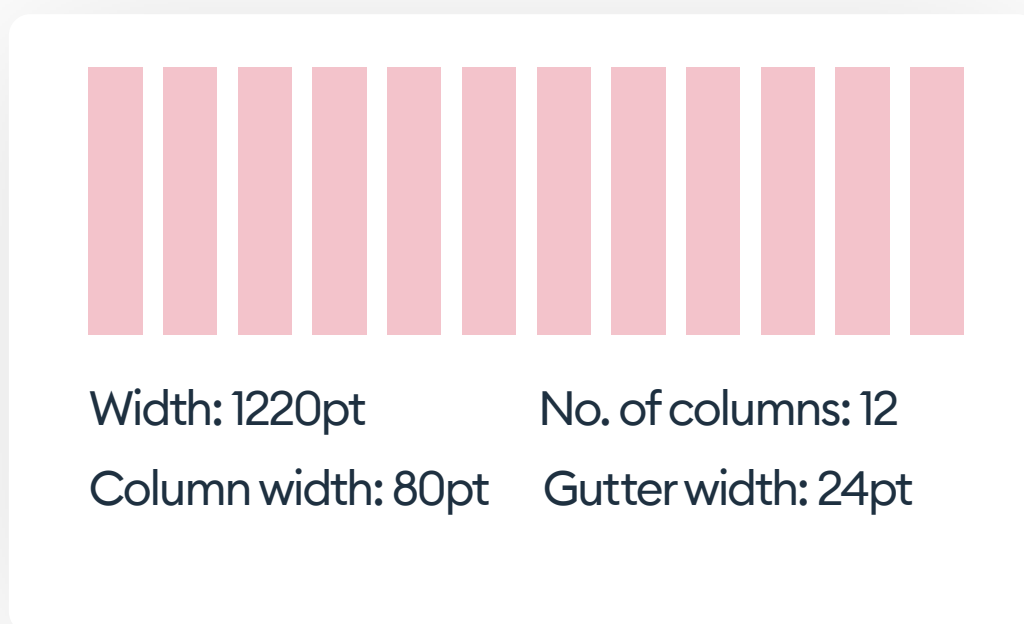
Treat those 1440 points as "safe space" for some layout / background elements to occupy, but define a narrower space for the actual content. I usually go with 1220 points. If you want to have a feel for how big the content will be, simply create a rectangle that's 1220 points wide and place it on your artboard. It should look something like this:



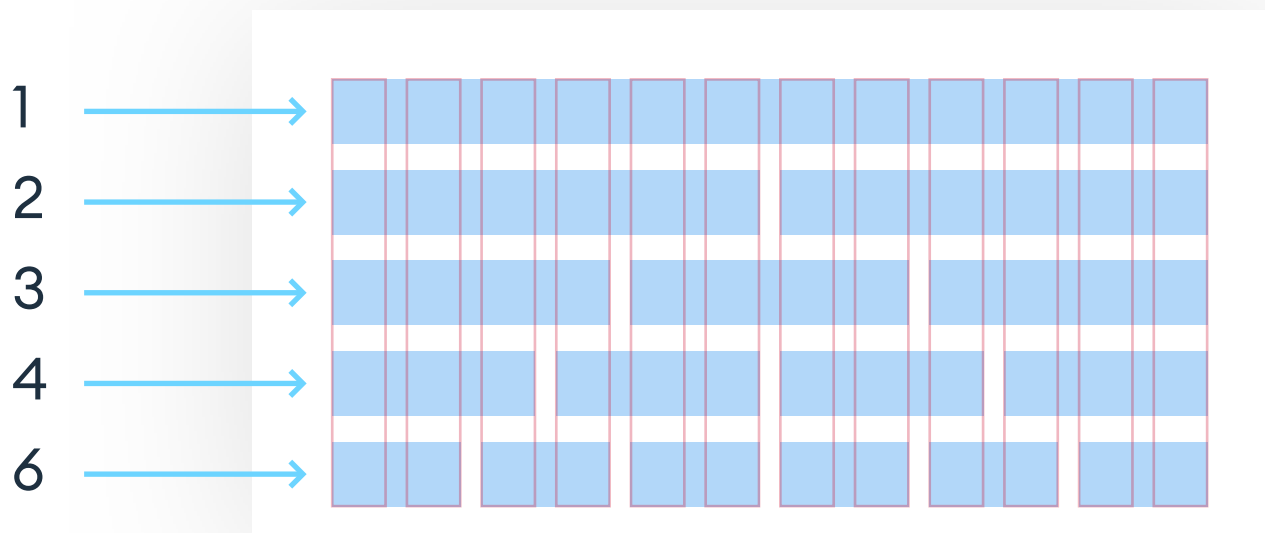
Why 1220? The reason is simple, it's the easiest number to work with if you want to use an 8-point grid with 12 columns. You can use a 24pt gutter with 80pt column size.

THE GRID

Starting this way will make it easier to visualize how much space you have for the content, and how much for background elements, full-width headers and more. So let's create our 12 column grid based on a 1220pt full size.



It's also important to understand why we're using 12 columns. It allows you to create sections on the website in the most possible configurations (1,2,3,4,6 possible equal section widths)



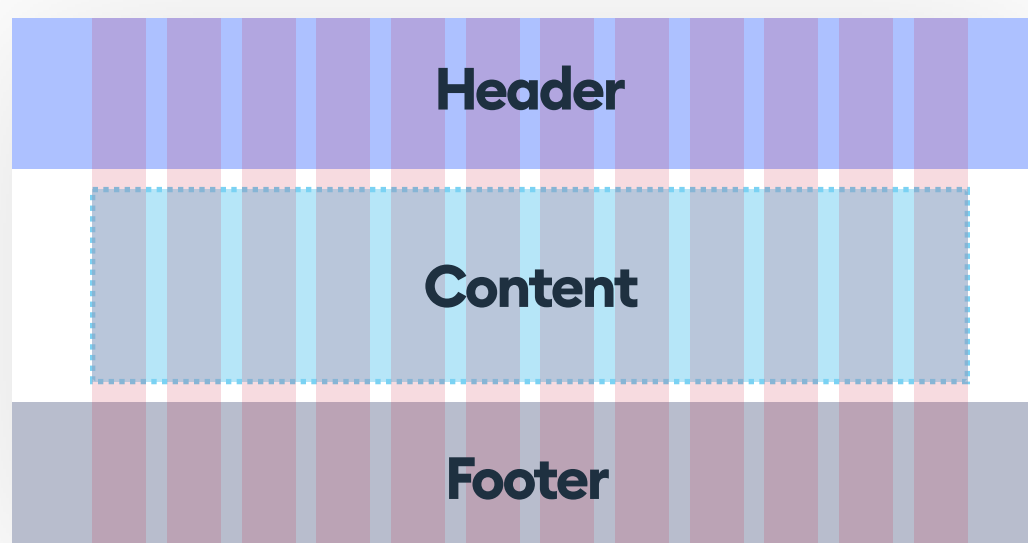
This is by far the easiest way to set your grid for a portfolio desktop project. Of course a "real" project may need a grid or layout dedicated to the type of content. (for example very complex, data-heavy designs may need more than 12 columns).

“

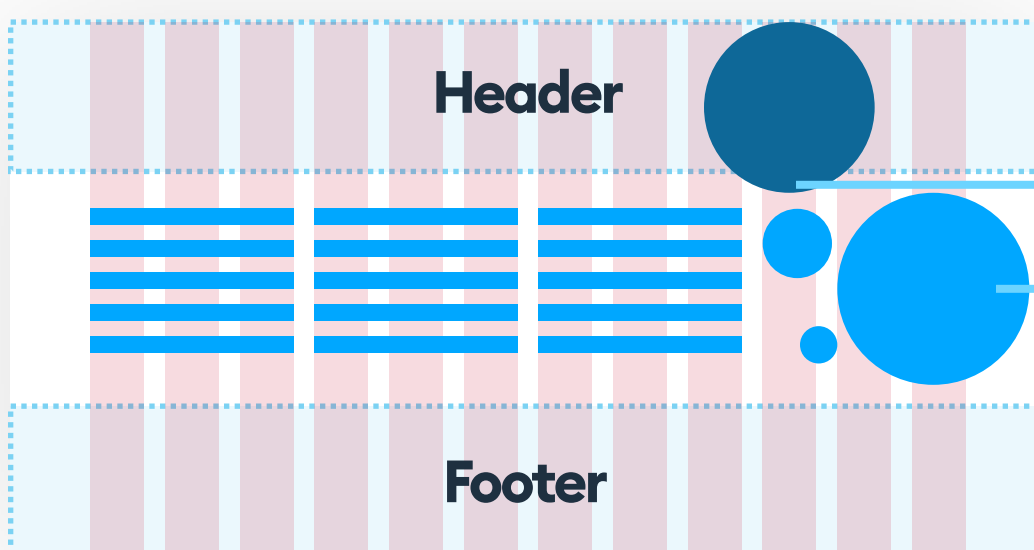
**A 12 column grid
is the easiest starting
grid for desktop
projects.”**

SETTING UP

Once you have your grid set up you can start by creating an outline of the information architecture, by simply placing the building blocks onto the artboard. For portfolio projects and landing pages it's good to create a header and footer that extend the full width of the artboard (and even beyond) while keeping the content within the boundaries.



The most important rule is to keep the content within the grid boundaries and the content sections aligned to the grid. However, all decorations and background elements can break this rule to make the layout more interesting.



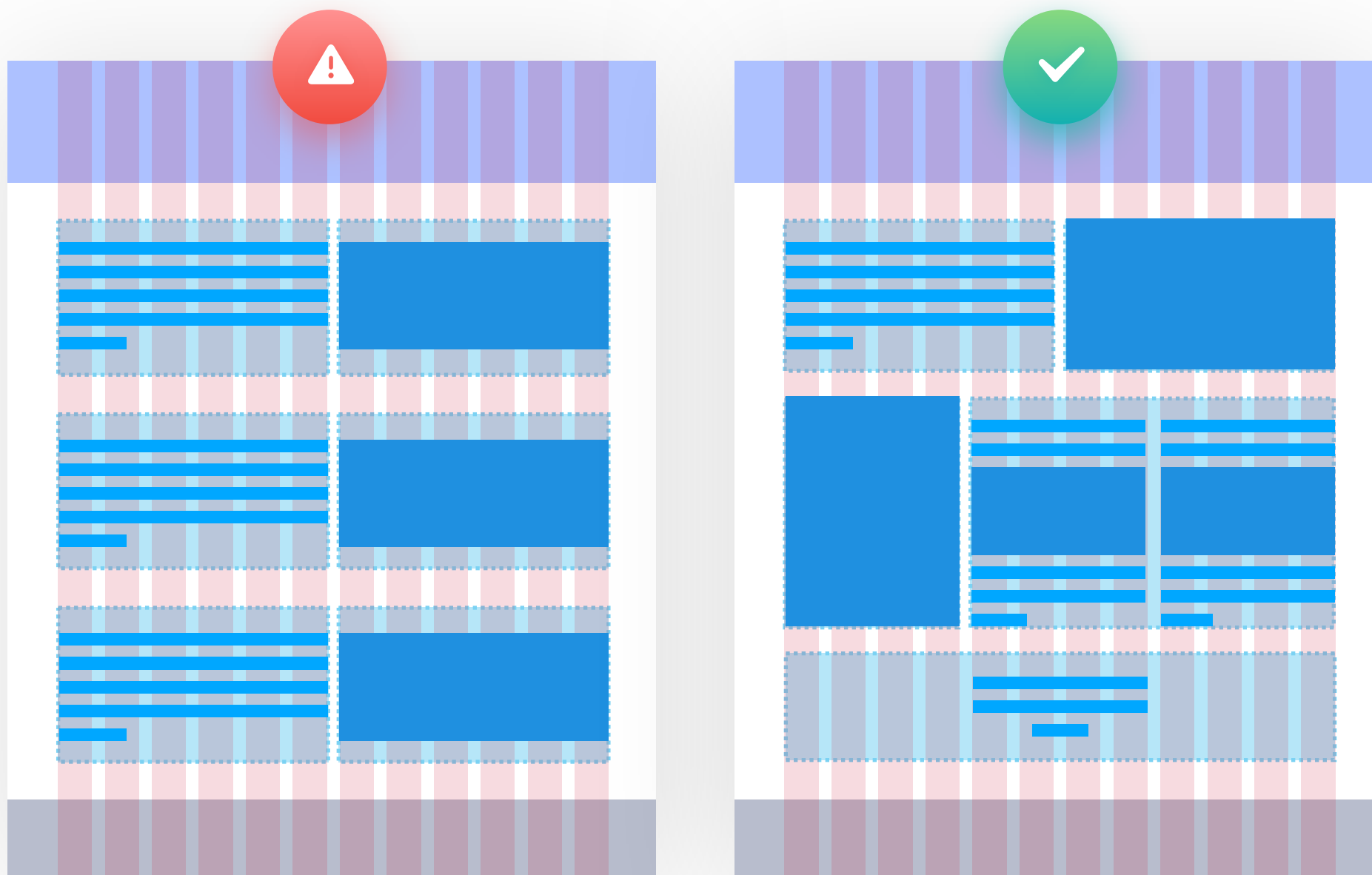
You can also extend separate objects to drop beyond the header constraints.

Visual elements like illustrations, photos and background shapes can go outside of the grid, as long as the main content fits within it.

Layouts with objects consciously breaking outside of the grid tend to be more interesting and engaging to the user. The most important rule is to only have a couple of elements like that on a page, as they will naturally attract the users focus.

LANDING PAGES

If you plan on having a lot of sections on your page, try to avoid using the same, boring pattern for each one. We need some variety and some structure to focus and keep the engagement.

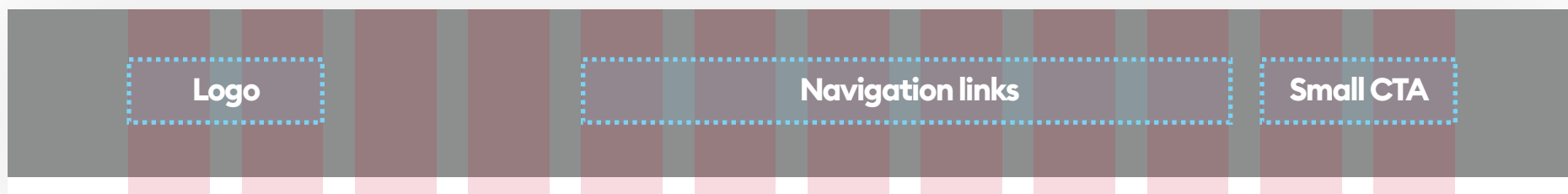


The example on the left has three main sections that end up looking boring and repetitive. Changing the background color in each section is not enough to achieve engaging content. Try to differentiate the sections in a way that make sense to show the content. If a section has less text, add a larger image next to it. If there's more text, you can separate it into columns, with the shorter call to actions completely centered with nothing on the sides.

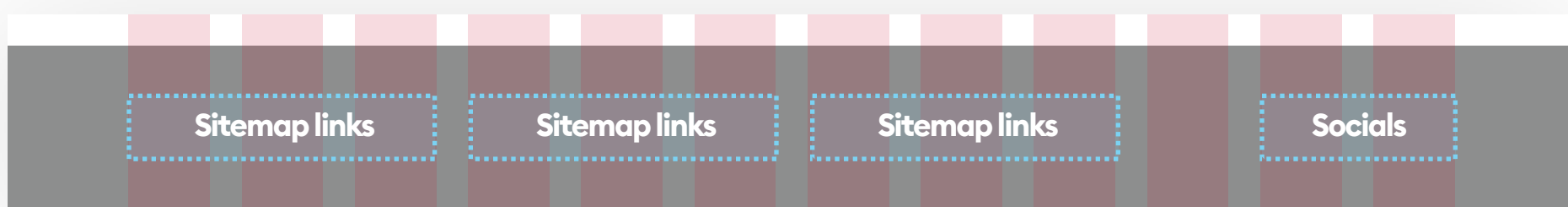
These varied layouts work especially well on landing pages, where the idea is to excite and sell an idea (or a product). For content browsing and other table-like forms of data it's best to stick to the same section style for each one.

SAFE PATTERNS

There are some safe, well-known patterns you can use if you're not sure how to structure your first landing pages. Before experimenting you can try using these to get more comfortable.



The most common (and thus most understood by the users) header structure is having the logo on the left side, but not larger than two layout columns, so it doesn't overtake the header. Then the smaller Call to Action should be right aligned within the grid - it can be a register / sign up button, or a purchase button. It sits naturally at the end of the F-pattern so if it's visually prominent it will definitely get noticed. The main navigation links can exist within the space between the logo and the CTA, in our example they're aligned to the right, but can also be centered or left aligned. However, make sure the logo has enough safe-space on its right side if you plan to left align the navigation.



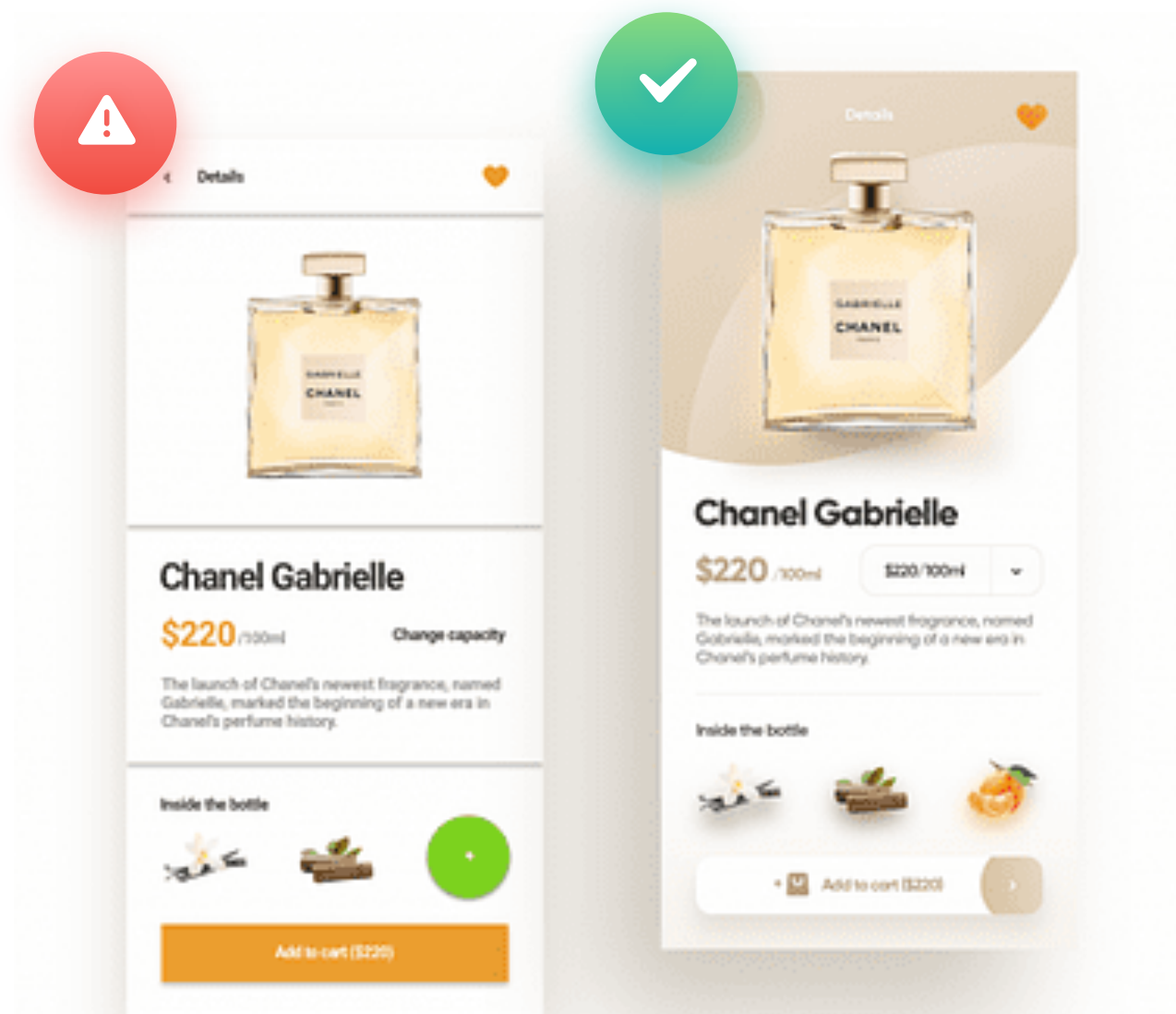
For the footer you can simply place the main menu again horizontally, or if you have more pages you can arrange them in vertical columns. Then keep the social media icons on the right side, as that's the place the users are typically looking for them.

As usual, look at other landing pages before you start and try to get inspired for colors, patterns, section dividers and more. The best way to learn is trying to copy something you really like and add your own twist to it.

MOBILE APP SCREEN

First thing to do, is deciding on the device your design will “live on”. Again, if you’re aiming to build out a UI portfolio first, it’s likely up to you, whether you’ll use an iPhone or and Android as reference.

If you want a truly unique, engaging showcase of your work, you shouldn’t really use Material Design or iOS components all too much anyway. They make the project look generic, and in the fast-paced modern world people will simply scroll right through them without pause.



These two projects are of the exact same screen, yet one of them uses generic, boring patterns, while the other tries to make it a bit more unique and interesting. In portfolio projects you should almost always go with the option on the right, as you want your project to be noticed, remember and liked.

“

**For portfolio projects
avoid using the generic
patterns, as they won't
stand out enough
to get noticed. ”**

Android and iOS, although very different as platforms, are constantly moving closer to one another in terms of UI norms. There are some things that should likely still be platform-specific (like textfields in forms), but most of the design can be made platform-agnostic.

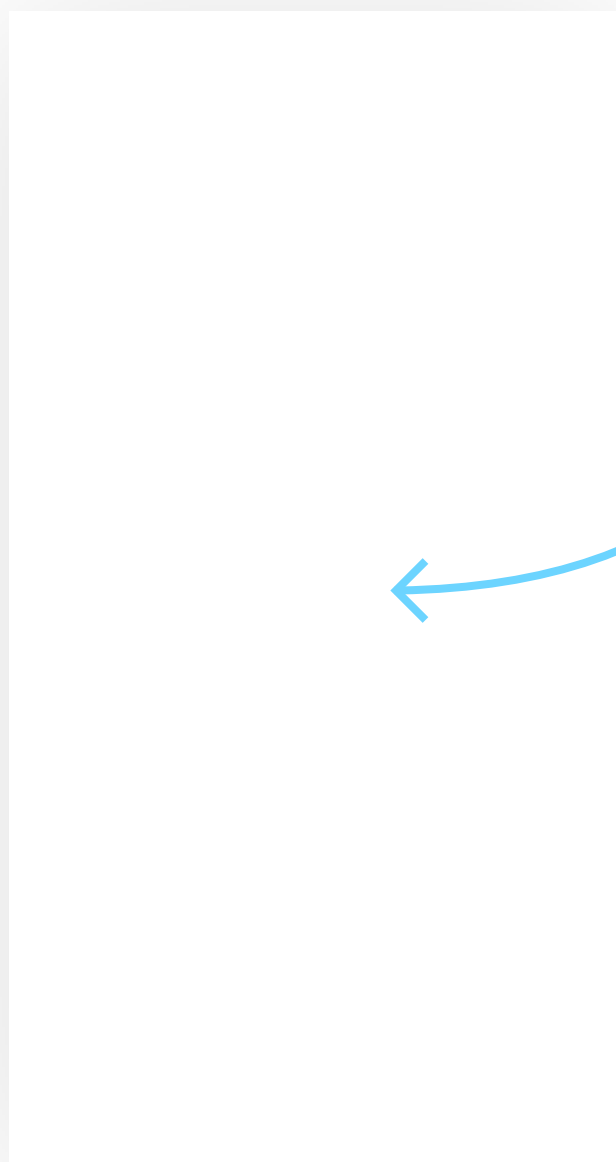
Pick your phone

So if you're a daily Android user, design for Android. Try to pick a phone with a 16:9 or taller aspect ratio. If you're in the Apple camp, go ahead and start with the regular iPhone 12 Pro (not Max).

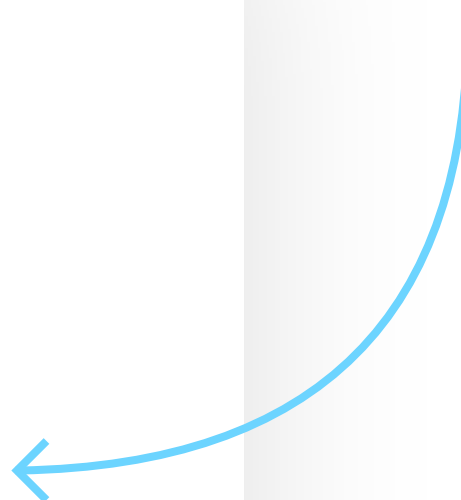
Start with the basics

For this exercise, I'm going to pick the iPhone. The first thing to do, is adding an Artboard by pressing A on the keyboard. That shortcut is the same for most design tools. Find the right device on the list and select it.

iPhone 12 Pro

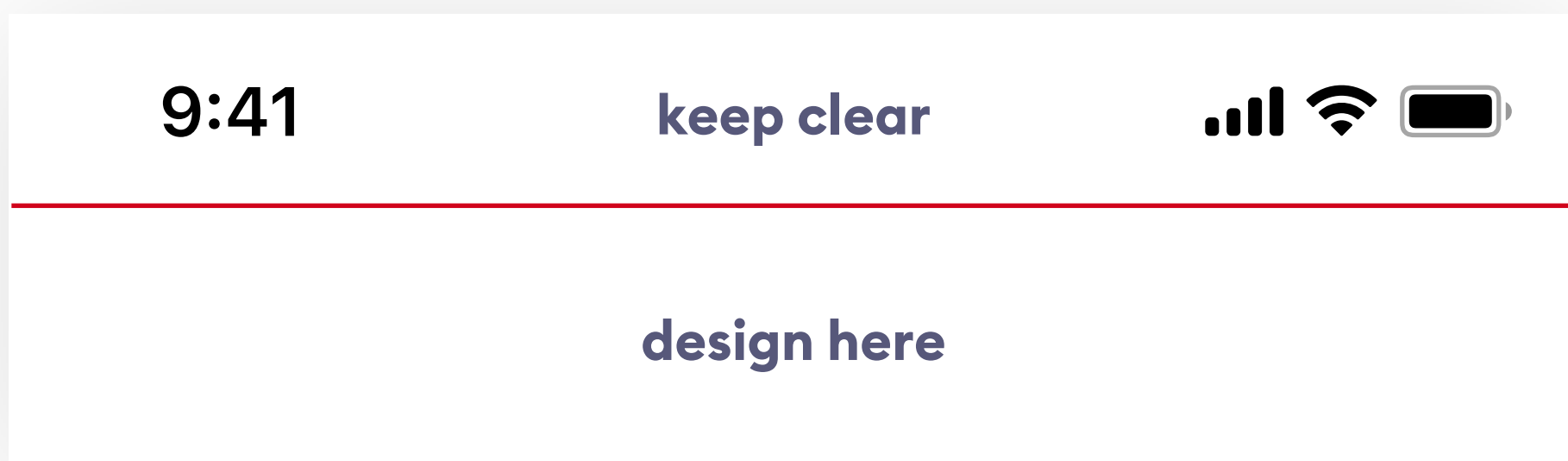


Time to make something awesome over here.



STATUS BAR

The next thing is to add a status bar – you can do it from the build-in iOS library in Sketch. Figma also has a similar option. Then make a guide at the bottom of your status bar and hide it. In showcase, portfolio designs they can be an unnecessary visual clutter, so they're often not shown. But the guide is there so you know which line not to cross.

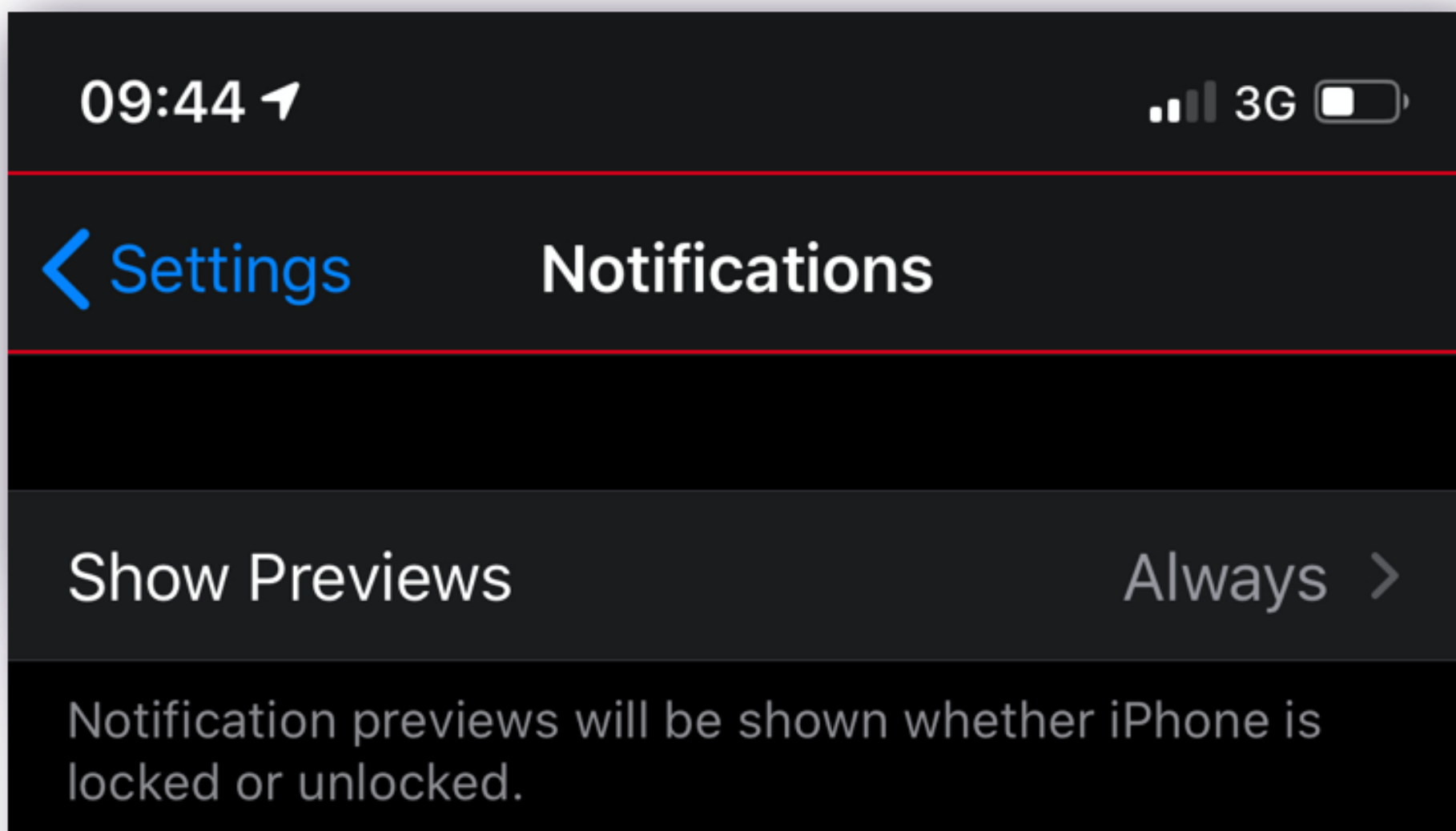


Navbar

Navbar, or Android's Action bar, is the title bar just under the status bar. It can have a couple of different heights, and you can check the proper one in the guidelines. They do change every now and then, so here's a different idea, that will help you get the right height instinctively.

Go to your phone and find an app that has the navbar and background separated. It just has to show the division. Most of the built-in Apple apps are a good pick here. I picked one of the Settings screens. Then take a screenshot and send it to your computer.

Drop that screenshot onto your artboard, set another guide where the navbar ends, and then remove it. Now you know how high it should be, based on a real, visual product.



That will make it easier to remember and recall than remembering a number that changes every few years. It will also give you a lot more context of how things fit on the navbar, than simply entering some number into a box. You should end up with two guides nicely dividing your artboard into three main sections: status bar, navbar and the rest of the design.



SET THE MARGINS

Mobile grids can be scary. Coming up with the right one, and then sticking to it is something that doesn't come easily. So instead of spending a lot of time on drawing grid-lines, make a soft-grid starting with setting the side-margins. These are "lines you don't cross". They're also something most designers don't pay too much attention to.

When the left and right margins are even slightly different, it shifts your entire design optically to one side and has a negative impact on readability. That's why I believe it's a good place to start.



The main two soft-grids right now are an 8-multiple and a 10-multiple. I covered them more extensively in the grids chapter with both the pros and cons for each one. It means that 8 and 10 respectively, are usually the smallest margins your content can have. Then 16 and 20 are one step higher, with 24 and 30 being the next set of values.

Design needs to breathe, so for most mobile projects I set the margins at either 24 or 32 points. For dense projects with a lot of data in column views, it's best to go with 24, and for simpler, less condensed screens let's use 32. It's a safe way to end up with a nice looking result.

“

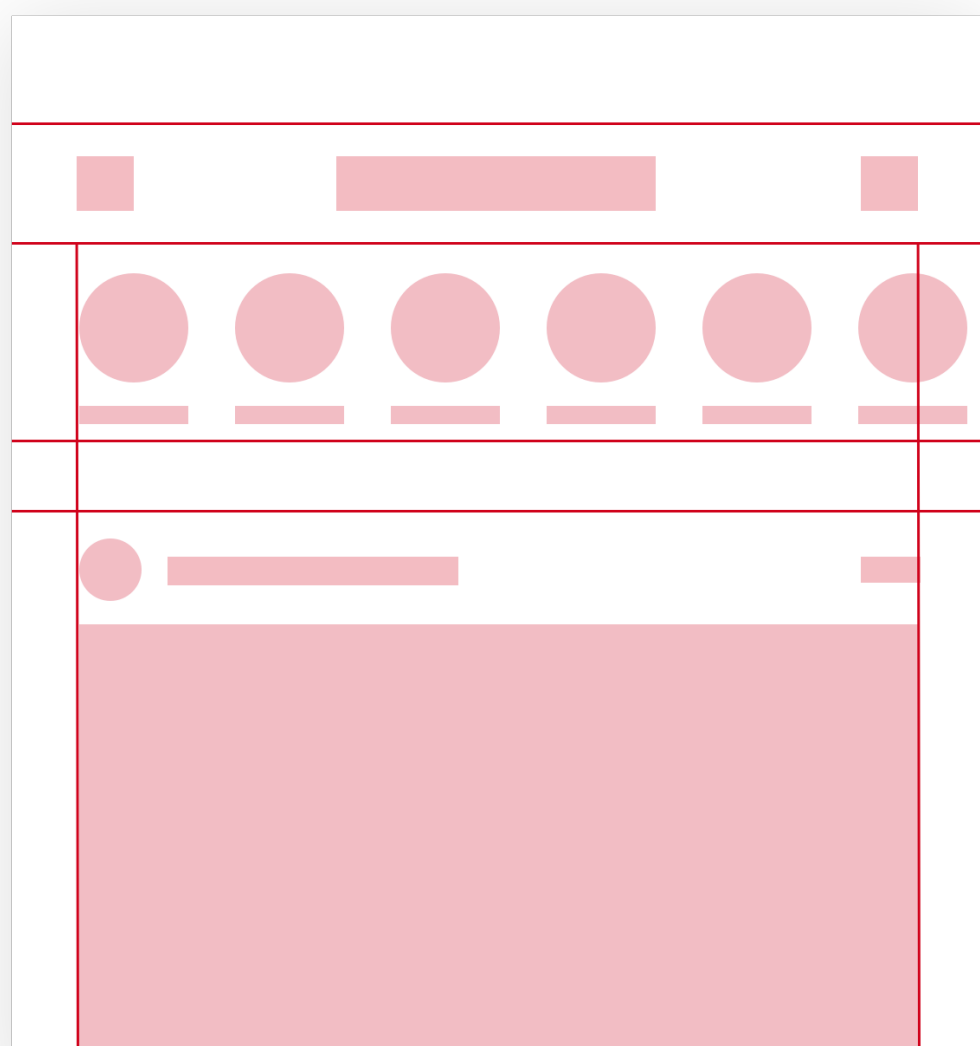
24 or 32 are the best starting margins for your very first mobile app projects.”

SAFE MARGINS

For your very first few projects start with 24 points as the margin and make sure no content (except for the background and decoration) ever exceeds that space. Then use that number and smaller numbers to set up the right hierarchy in your design.



If you have a rough idea of what you're trying to design in mind, you can start by blockframing it. It's a useful technique of working with "bounding-boxes" instead of content. Just make sure they're all properly aligned. For example, if you wanted to recreate a popular photo-sharing app it would look something like this:



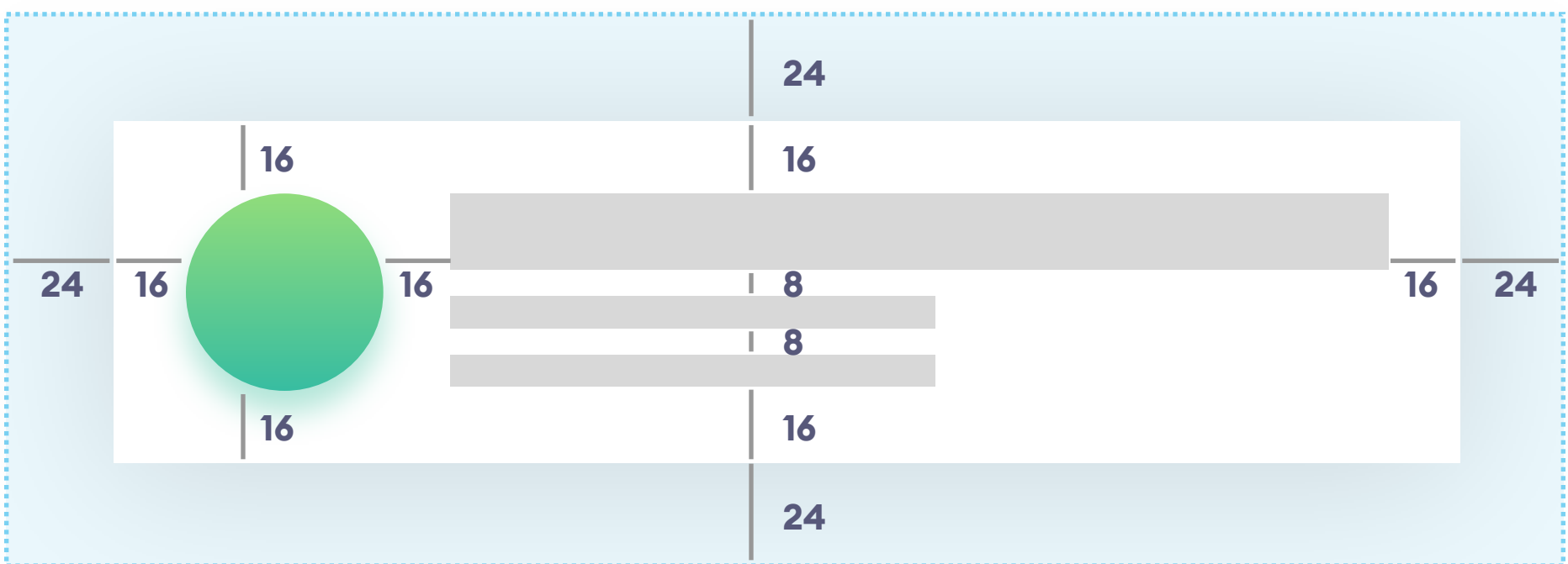
Blockframes are a good way to understanding the rigid rules behind a good design. Because of these uniform shapes, it's easy to get a feel for proper alignment before you start adding real, visually varied content. Just make sure the margin between the elements also comes from your grid setting.

“

Each element's height, width and position in relation to other objects should be thoughtfully calculated and planned. Don't leave anything to chance.”

BLOCKFRAMING

Rule of Proximity is essential here. The closer the objects are, the more they are perceived as a group that belongs together. When designing a card layout, obviously the content inside the card will be closer together, than the cards themselves. The main benefit of blockframing is that it's a lot easier to apply that rule. Let's look at it from a perspective of one single piece of the layout - the card.



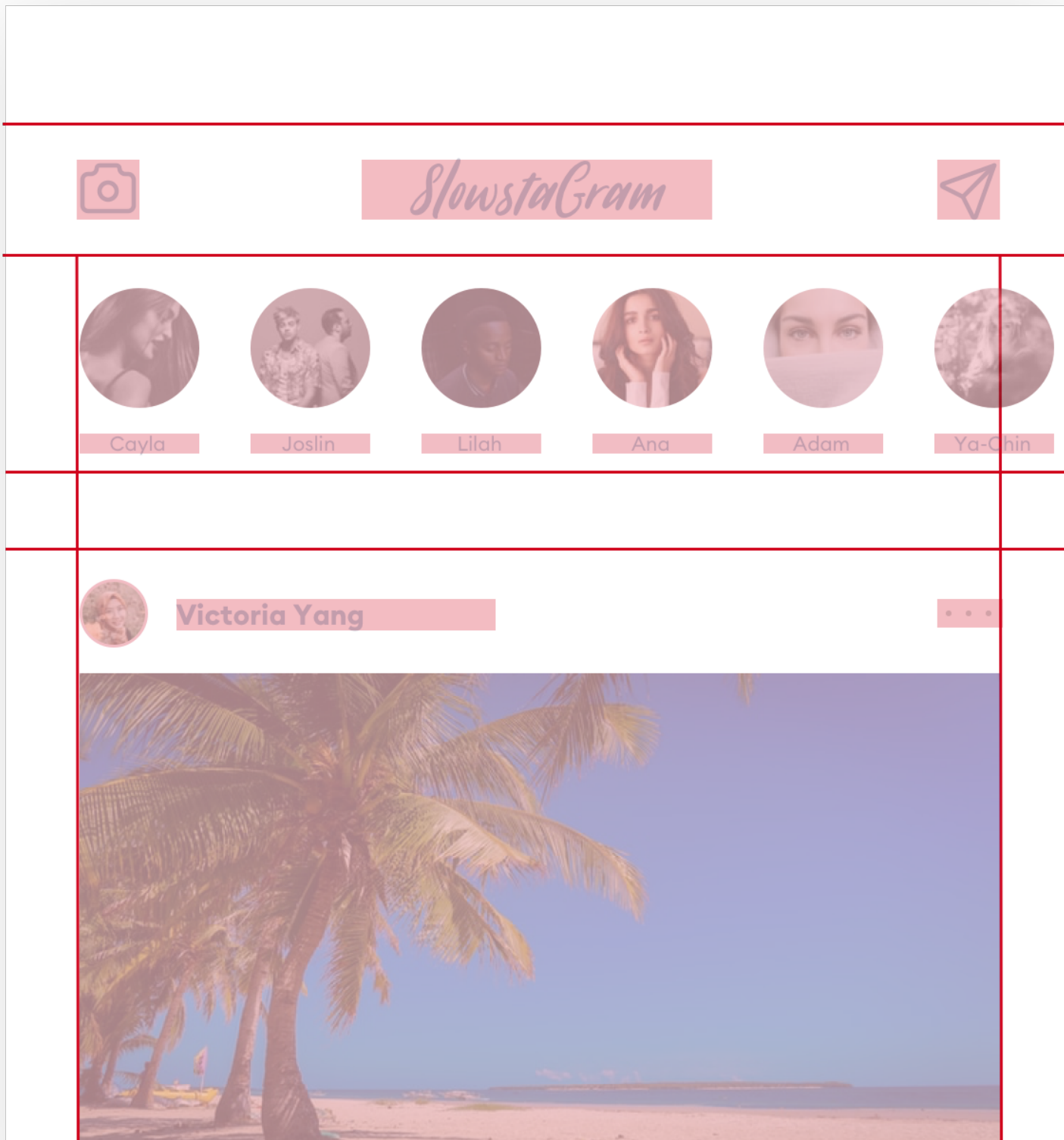
The card here is the main "group" so it's positioned 24 points from each side of the screen. The circular photo is another group, but it resides inside the card, so it's positioned at 16 points from each side. The text content blockframes are yet another group, so they're positioned 16 points from each side, but the space between them is 8, to show they belong together. Use that rule to easily group common sections together.

When you replace the blockframes with real content, some of the perception may be lost. That's ok, as long as that content sits within the boundaries set by the blockframes. You can find out more about the purpose of blockframing in this wonderful article by Jon Moore:

<https://medium.com/ux-power-tools/blockframing-and-31-free-sketch-ready-layouts-using-auto-layout-by-anima->

BLOCKFRAMING

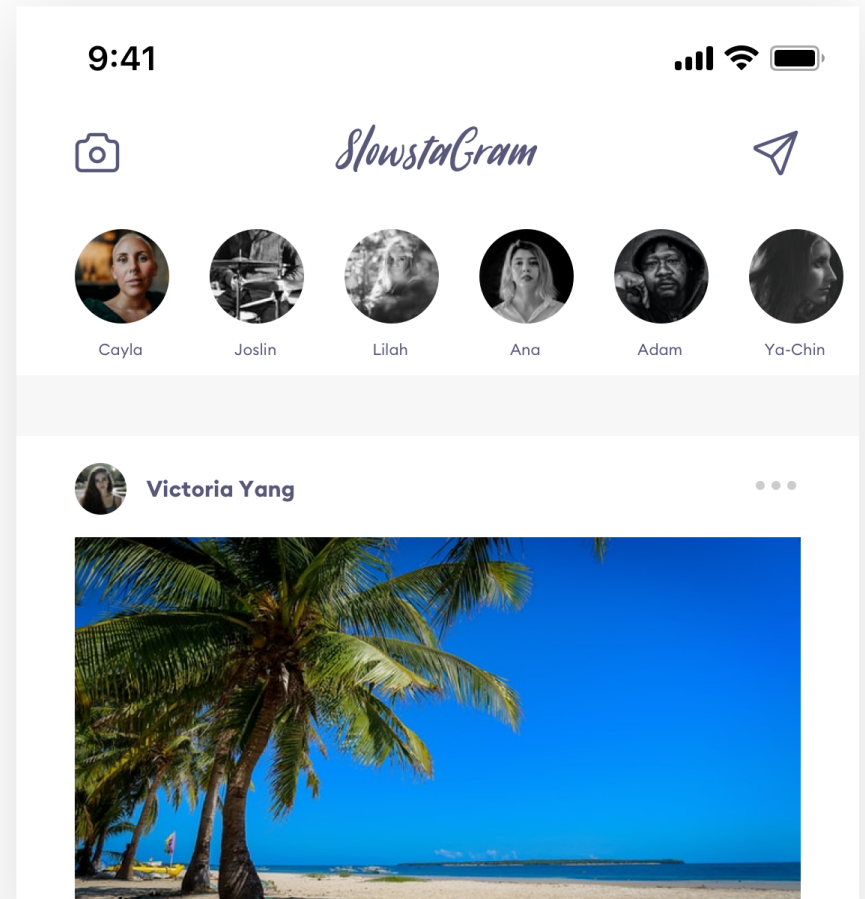
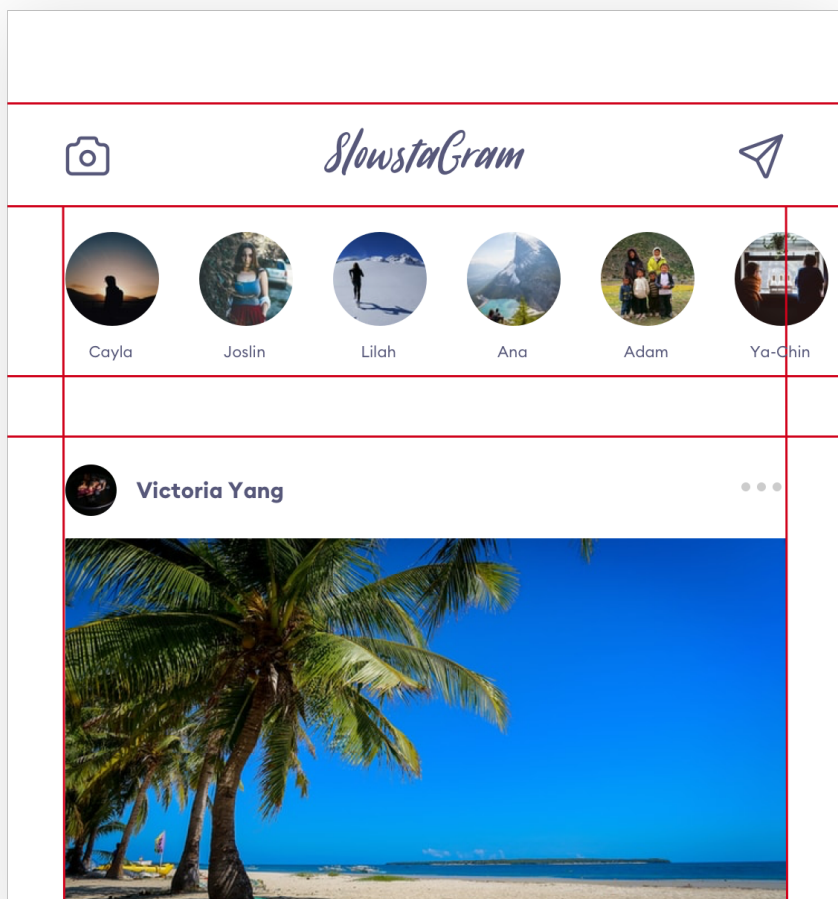
Now back to our example photo-sharing app. Once you have the blockframe complete, you can group it and lock it with reduced opacity. Ten percent will be enough if you use the right color that won't blend in with the content too much.



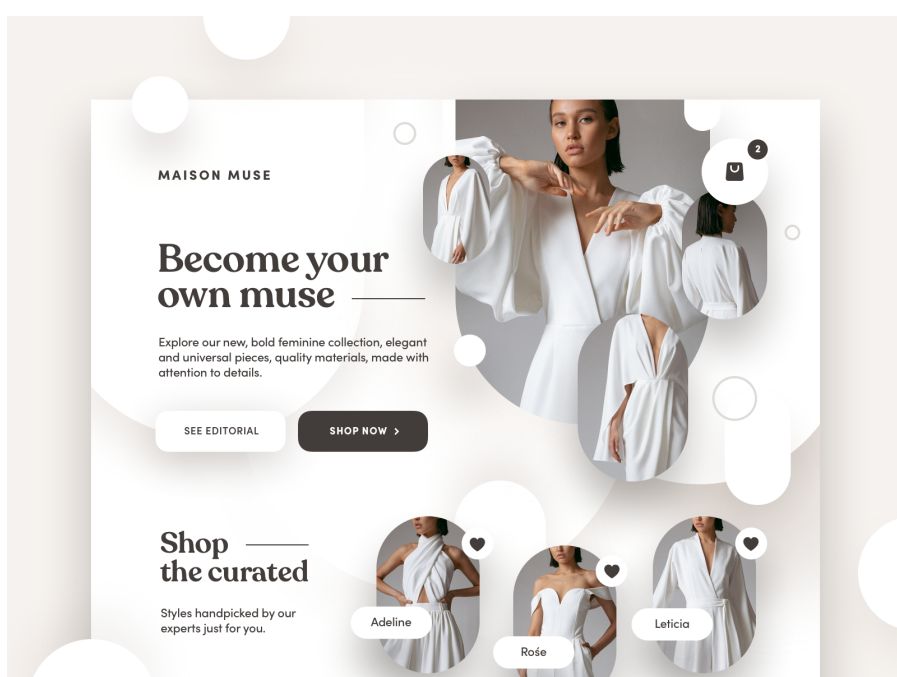
Then start adding the real content but fitting it within those strict guidelines. You can in some cases get the descenders of lowercase letters (y, j, g and the like) get outside of the blockframe boundaries - if they do that's also OK. As long as the CAP-height is aligned to the blockframe the layout will still work well.

BLOCKFRAMING

After you're done with this part, simply hide the blockframe group and you're all set. You can then also hide the guides and notice how the layout starts to "naturally make sense" without them.



Of course you can still refine the grid and spacing using the Golden Ratios and other sophisticated design rules. Yes - it can be made even better, more readable and easier to process. But for your first couple projects this rule will save you a lot of headaches and will allow to get to satisfying results much quicker.



The same rules also apply to more complex, organic layouts with background shapes and patterns that fall outside the rectangular grid. Strictly focus on aligning the content not the backgrounds.

“

Spacing alone can set a clear, visually attractive hierarchy of on-screen elements. Combine it with the right sizing for the best effect. ”



32.

**Success in
design**

Success in design

STAND OUT FROM THE CROWD

The goal for this book, was to condense our years of experience into a form that you can easily use to become a better designer. I am happy to say that we succeeded (judging by the countless emails from our readers). But knowing how to set the typography, how to pick the right color palette or how to make nice buttons is only one part of the story.

What I want you to have at the end of the day is success as a designer. Stories like this are the ones that I like to hear the most. All those emails saying how after reading this book you leveled up, got a promotion, a raise, or launched a successful startup are heartwarming and make the time writing this completely worthwhile.



Success doesn't come from the design knowledge alone. In this chapter I want to give you some tips on how to use the knowledge from this book to your advantage. You can help your "luck" and directly influence the chances of success. Whether you want a more exciting, better paying job, or start your own thing. This chapter is going to be a lot different and more personal than the rest of the book. Let's go!

Working hard vs working smart has been a mantra of many self-help books for years now. They often try to give you "hacks" that you can use to "work smart". It's like a get rich quick scheme, but the sad truth is that it likely won't work. But don't worry.

The "working smart" part here is the fact that you're reading this book. You could obviously learn all this on your own. Successful designers did exist before we released *Designing User Interfaces*. But finding the right sources that also make it easy for you to understand them takes time. A lot of time. So in this case you can already check-off the working smart part.

Now it's time to work hard.

If you finished the book, you got through the first part of this process already. You have a solid base, a foundation on which you can grow as a successful designer. But that success is not going to happen overnight. There are a couple of factors that can determine whether an employer will pick you, or somebody else, and design knowledge is only a small part of that.

Things like charisma, confidence, portfolio, ambition or even being fast on your feet, can all determine your chances for success. People (employers included) are not an AI, that determines an outcome based on the input (knowledge). And while I can't teach you some of these things, I can point you in a direction that definitely helped me.

Big part of my life I've been an introvert, afraid to speak up, afraid to step to the front and afraid to take chances. Some parts of that you can modify, for others you can simply find a workaround. But don't worry, you don't need to become as charismatic as a talk-show host overnight.

“

**You've got a solid
foundation. Now it's
time to work hard.”**

BUT WHY?

Let me tell you a little secret. Designers are one of the luckiest people in this whole digital industry. As most of IT we can of course work from anywhere, and photos of a laptop, running Sketch or Figma with a nice beach view in the background aren't that uncommon. We can travel the world, enjoy it, make some good \$ and experience freedom.

But there's also one other part of this that's often overlooked. Design (especially interface design) is becoming increasingly agile. Combining that with practice, experience and some speed designing techniques you can actually switch from an hourly rate to a per-project basis. With some practice you can work 4-5x faster, completing all your daily work in 2-3 hours and enjoying life. Or using that extra time to do another project to enjoy more money. It's all up to you.

Developers don't really have that advantage, as there aren't really many ways they can "code faster". The sheer complexity of what they do simply makes it nearly impossible. For us, designers it's a world of opportunity that is only waiting to get noticed.

And sure, getting that experience to be able to work that fast requires a grind at first, but what's a year or two of grind compared to what comes next? I've been there through the hard-way. Both at various agencies that I worked for, and running my own company, grind and crunch were practically a daily thing. But now looking back I see that I wasn't having the smart approach to hard work.

It's important to prioritize things that are actually beneficial, instead of doing everything, overwork yourself and fall asleep on the keyboard dreaming of working even harder the next day. Hard work has more to do with discipline and internal consistency, than working crazy hours, neglecting your health and wellbeing. Let me share some tips with you. It can actually be a fun ride.

IDENTIFY YOUR STRENGTHS

Charismatic? Me? No way! That's what many introverts say and it's something that can't be pushed upon you. Some people can become more charismatic, while for others it's simply something they'll never be comfortable with. But don't worry, because you can still be successful! It's all about identifying your strengths (and the already mentioned hard work).

So you don't need to be a swift talker, if your portfolio blows away everyone looking at it. Or if the sheer number of portfolio projects you've completed shows ambition and dedication.

Challenge yourself.

I think the best way to start is with things you definitely can control. One of the best ways to grow as a designer is to actually design a lot. Put a daily design challenge upon yourself for at least a month. They don't need to be big, complex projects. Anything between 30 minutes and 1 hour to complete is good. But make sure you actually commit and do it every single day. Yes, that includes the weekends. Think of this as a workout for your design skills.

I know it's hard. I know that after work or school, you often don't have the strength to do anything but Netflix-and-chill. But trust me with this.

Relax is important, but having a robust portfolio will actually lead to a lot more relaxing and a lot less stress later on. That first year or two of hard work makes it way easier in the next stage, when you stop being a junior designer and start to work on your own terms.

Besides you don't have to "find the time". You need to make the time. It has to be a conscious effort to sit down for at least 30 minutes each day and work on your future.

SWEET PROCRASTINATION

It's not easy. I know. Even writing the last page I took like 8 breaks because I couldn't focus. We have worse days, we have moods, we can feel under the weather. We're only human. But the reward of actually completing even a small exercise to better yourself is going to stay with you a lot longer.

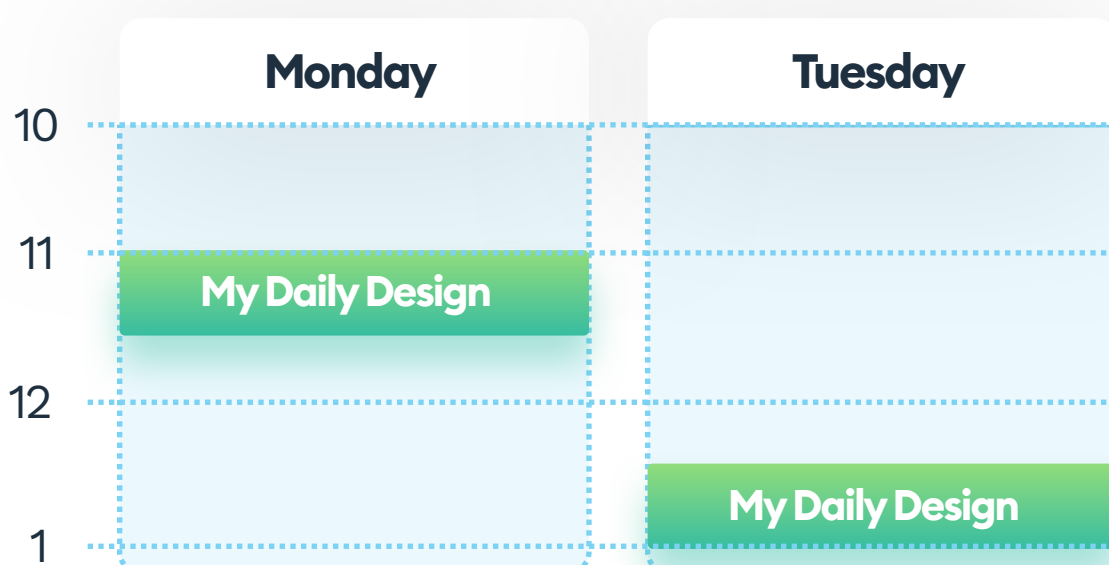
So how to beat procrastination even if for thirty minutes every day?

I have a few tips for you, that I tested on myself. Try and see if they work for you. Some may be helpful to get motivated to work more than just 30-60 minutes a day of course. It's just a starting point for skill, experience and a better portfolio.

There are two techniques that I found the most effective. You can do them both together, or pick one that you prefer. These are:

- Time blocking
- Notebook based, paper to-do lists

Timeblocking is simply creating a calendar appointment for yourself. Pick a time you know you'll be available at. Put your phone on Do Not Disturb and simply dedicate that 30 minutes to do as much as you can. If you can prolong it, do it. If not, then just stick to the half-hour blocks first.



You don't need to do it at the same time each day. Just make the time between other activities, before them, or after them. Plan it for a week ahead and stick to it.

“

30 minutes a day for a month compounds to achieving success much faster. ”

Timeblocking works, because it creates an urgency of a meeting with yourself. If it's scheduled for a specific time and you won't show, it creates a dissonance. Our brains hate that inconsistent feeling so by doing this we actually trick ourselves not to procrastinate.

The fact that we are living in an absurdly digital world makes pen and paper hold an especially important place in our minds. That's why a Paper To-Do list proves to be very effective. I use a small notebook as a to-do list for the most important things I want to accomplish on a day.

The layout is simple, the date at the top, and little squares with tasks next to them. But because it's written in pen, I can't erase it. I can't "move" a task to another day. It's "set in stone" and it feels really bad to actually skip an item on a day.

To make the experience more fun you can also add some other, smaller and easier to complete tasks next to your main thing on each day, so checking them off will give you that little boost of endorphines. That in turn gives you motivation to tackle the more difficult things next.

And that way, day by day, you'll be getting that "extra" practice. It doesn't matter if you're just hoping to start in this industry, or if you already have a design job. Going at it daily, even if for a short period of time will transform you for the better either way.

That's why the advice in this chapter is not for beginners alone. To stay on top of the techniques, trends, good practices and patterns you need to explore new things that are outside of what you normally do at your job or at the course or bootcamp.

30 DAYS LATER

I'm keeping my fingers crossed, that these techniques and your own motivation will lead you to actually complete a 30-day design challenge. Keep in mind, that most people quit around day 6. If you make it to day 10, you're well on your way of completing the challenge.

During those 30 days you should focus on designing the most popular interface elements there are. You know, the things you see in most products. Start with forms, there's plenty to do there. Do a login, registration, no-password login, forgot password form, or a profile edit screen. Then move over to dashboards, data visualisation (graphs), chatbots and more.

But every now and then do something less "design'y". Do a floating, glassmorphic credit card on a nice, blurred background with faux-3d spheres around it. Or a bunch of product cards floating in space. These creative, more-visual outbursts are what will help you find your own "style" of interface design. That style is a mix of various design trends and patterns with your own personal touch as a cherry on top.

Finding that style will take time and effort, that's why you need to consistently do a design every single day. You can follow tutorials if you wish as well, or if it's easier. Just find ones that can be fast and easy to complete. The goal of the first 30 days is to gain confidence in moving rectangles around.

Ok, but what's next?

Once you realize you can accomplish something like that, you can apply that dedication and motivation towards the next steps in your design career. And that by no means is an invitation to stop doing the challenge. The more you do it, the better you'll be. 30 days is just the beginning, but passing that mark will make it a breeze to continue and your future self will thank you for that.

“

**Make sure there are
designs in your design
portfolio.”**

Aside from the personal traits (that you can find plenty of resources on how to train online) there's one other piece that instantly sets you apart. I'm talking of course about your online portfolio.

But before we get to that, you need to understand that the landscape of the wide genre of "UX design" or "Product design" is so chaotic and convoluted, that the word "portfolio" can mean very different things to different people. I want to share with you my vision, based on my predictions of where the industry is heading. In short - the multitude of job titles will be simplified to researcher and designer.

That's why UI's are an important part of a design portfolio (also because potential employers and consumers also "buy" with their eyes). But you also need to show that you understand the overlaying "experience" element. That means that the best portfolios are a good mix of beautiful visual design, with descriptive thought processes explained in text or video. You can add a little bit of research (like stats from surveys you created) to show versatility too. It's extra points when you, as a designer, can clearly communicate with researchers in the team.

I believe the era of "UX portfolio without UI" is coming to an end. Only some companies still work with that detached, multi-function way, but more and more are embracing generalists. If you feel better with planning flow-diagrams or information architecture, then of course you should highlight this as your strength. But make sure to also show that you're able to make a semi-finalised design out of them if the need arises.

The best portfolios are a self-hosted, simple websites with short case studies. Knowing a typical product design start with ideation phases, and ends with a high-fidelity prototype, show this exact process in yours.

PORTFOLIO

A good portfolio should be as unobtrusive as possible. What works best in presenting your work is going as minimal as possible. Same goes with your Resume. Simple layouts, not too many font sizes and readability should be at its core. If you have a lot of projects, it's best that the interface of the actual portfolio is not overtaking the designs. After all this is what people are looking in portfolios.

But at the same time, using a pre-made template can ruin that "first impression" that you can only make once. Of course if your portfolio is a work of beauty (but still readable!) in itself, it'll likely make an impression even before the clients or employers see the designs. If you can make beautiful 3d transitions, amazing visuals and so on - go for it. But the problem is, that most junior designers aren't yet on a level to deliver that portfolio "wrapper" well enough. That often results with a very visible "trying too hard" vibe that falls flat with the viewers.

The safest way is to go minimal, but show the scale of your work. You can even state the number of individual projects on the page near the top to reinforce that. Of course don't do it if the number is lower than 10, because it can have an opposite effect.

Make sure you write a short bio and include a photo. The bio should simply explain your story with design (why, how and when you started), your future goals and tools or techniques you're exploring (the more the better). Avoid having "progress" bars next to the apps you know or skills you have. If you're confident you know Sketch and Figma, just write that you know them.

Potential employers often look at the tools you use first (before looking at projects) to find out whether you're a good fit to how the company works. If they work in a specific tool, that you've never used, chances are the onboarding process will be longer and that may result in them picking someone else.

CASE STUDIES

Unless you're applying for a purely UI designer role to some of those oldschool companies who still use that division, you need to have a bit more than just images. Case study sounds very sophisticated and likely over-the-top for what you have in for-fun projects, so let's say the proper term would be mini-case-breakdowns instead. Getting people interested is strongly connected to the storytelling element. If you've made an app project you'd like to show off, write a "fake bio" of that fake company and their fake product. One or two short paragraphs is fine, and makes a nice intro to the project. Just make sure to state that this is all a part of the process, so viewers understand it's made up.

That description should also cover the general problem the product is solving, how it's doing it and what other choices you've explored and decided not to take. (These can be bullet-points). If you've created some twitter surveys (great way for portfolio-research examples) add them here too.

The next step is to come up with a flow diagram at least for the parts of the screens you've designed in high-fidelity. Show the connections between them and try to make it in a beautiful, visual way - even using card components from the actual app for that flow for consistency.

After that some people do low-fidelity wireframes, but with the way design tools work now, they're likely going to be less and less important in the years to come. Mention creation of the base, high-fidelity components (like form fields and buttons) as the next step instead (call it a pre-design-system component library if you want some extra points ;). Of course show a couple of them visually too.

And after that show the visual designs in the best possible light and you're all set!

“

Low-fidelity wireframes are slowly becoming less relevant in portfolios. Show other parts of the process instead.”

TAKE THE (VIRTUAL) STAGE

Aside from the personal traits (that you can find plenty of resources on how to train online) there's one other piece that instantly sets you apart. That part is being active in the community on various meetups, panels and even solo presentations.

For years there has been a problem of facing stage fright and wasn't even considered by many introverts. But now, in the era of online events it's easier than ever!

If there are any junior-friendly events out there, volunteer if you can. Many beginners are afraid of speaking in public, because they feel that they don't have anything to say yet. And while obviously years of experience can help with having an interesting keynote, any of your current experiences are also worthy of sparking interest.

Imagine - many people on your level are going through similar struggles, problems and patterns as you. Sharing your path, be it a case study, or discussing a failed project, can lead you to becoming a voice that's known in the industry. So share what you know already and be honest. You'll quickly see that it's enough to be interesting.

Make slides for your main points, practice a lot before the event, but keep in mind that with online events you can have an entire printed "script" of what you have to say in front of you. That should decrease the tension and make it easier - you can always look at the script for help.

When I was just starting to go on stage years ago, my technique was to prepare a lot of slides. At least three slides for a minute of time. That led me to always being able to improvise something when you can only discuss something in 20 seconds. But it all depends on how talkative you can get.

BE ACTIVE

Be active in places where other designers are. Dribbble, Twitter, Medium, Instagram - they all provide a place where you can find inspiration, talk to like-minded people and make your first "design friends".

Being a part of that community (a worldwide one) will also help you to grow. Even if you read one five-minute design article a day, you'll see that knowledge compound into automatic design decisions in your actual projects. If you can, share your work and your struggles, ideas and more on social media. That will open a world of opportunities for you, like appearing on podcasts, invitations to speak at online events, and even possibly job offers.

You can also do basic research on Twitter if you don't have any real projects just yet and need at least basic research examples for your case studies. The possibilities are endless, and if you truly want to be a successful designer, then get out there and talk to people. Even when you best design alone (like I do), design is a team sport and there's plenty to learn from everyone around you.

If you have the dedication, you can start a blog (or a Medium page), or a YouTube channel, but keep in mind that these things will take your already limited time, so they'll likely require you to commit even more time to your growth. That means there will be less time for other activities. That's why it's everyone's personal decision to take.

But if you can't even do 30 minutes a day for a month to grow your skills, your road to success will be much harder and longer, if not impossible.

Good luck!

**We wish you all the success in
the world! Thank you for being
with us!**

Michal + Diana.



33.

Summary

Summary

Congratulations!

You've made it to the end of the 2021 edition of "Designing User Interfaces."
Thank you!

You can also join our Slack Community by simply sending us an email.

All the updates until the end of 2021 were free for everyone who purchased the book. We are also working on our second, slightly more focused book (smaller), a video course and more. We'll offer considerable discounts to all of you who purchased the current edition of this book.

Don't forget to follow us and let's connect!

Our company website - www.hype4.com

Our company on LinkedIn - www.linkedin.com/company/hype4

Michal on YouTube - www.youtube.com/c/malewiczhype/

Michal on LinkedIn - www.linkedin.com/in/michalmalewicz/

Diana on LinkedIn - www.linkedin.com/in/dianamalewicz/

Michal's Twitter - <https://twitter.com/michalmalewicz>

Diana's Twitter - <https://twitter.com/dianemalewicz>

Michal's Medium - www.medium.com/@michalmalewicz

Diana's Medium - www.medium.com/@dianamalewicz

HYPE4 Academy on Instagram - www/instagram.com/hype4academy/

THANK YOU!

Big thanks to all of our friends and family! We know how hard it had been for you to listen about box-models and button alignment over dinner!

Thanks to the core team of HYPE4 for all your support during this ordeal:
Albert Pawłowski, Szymon Adamiak, Albert Walicki !

Special thanks to all our friends who attended our DGS events in Sopot, Poland. You shared so many designs problems and issues with us that we knew what to focus on first and to our **SquarePlanet** community.

Thanks for all the help with marketing tips, promotion, first reads to :
Patryk Kościelniak, Kacper Wiśniewski, Piotr Canowiecki, Brian Egan, Silvia Bormüller, Fabricio Teixeira and the entire **UX Collective!**

Book Website (www.designingui.com):

Design: **Diana Malewicz**

Development: **Albert Walicki / HYPE4.com**

Photos from: **Unsplash.com**



This book and all the visual examples were all designed in **Sketch** (www.sketch.com) which is our favorite UI design tool (and it works for ebook design too!)

OUR OTHER BOOKS



Good design is one thing, but a bad developer can ruin even the best projects. This is why we're creating this special book for Front-End Developers that will help them get to the next level, earn more and do a way better job!

You can sign up for discounts and free downloads at:

<https://frontendunicorn.com>



Aside from my own startup experiences ranging from 1999 until today, I also interviewed dozens of founders who managed to make it, often against all odds, and distilled their insight into easy to follow examples of what to do and, most importantly - what NOT to do in the startup world.

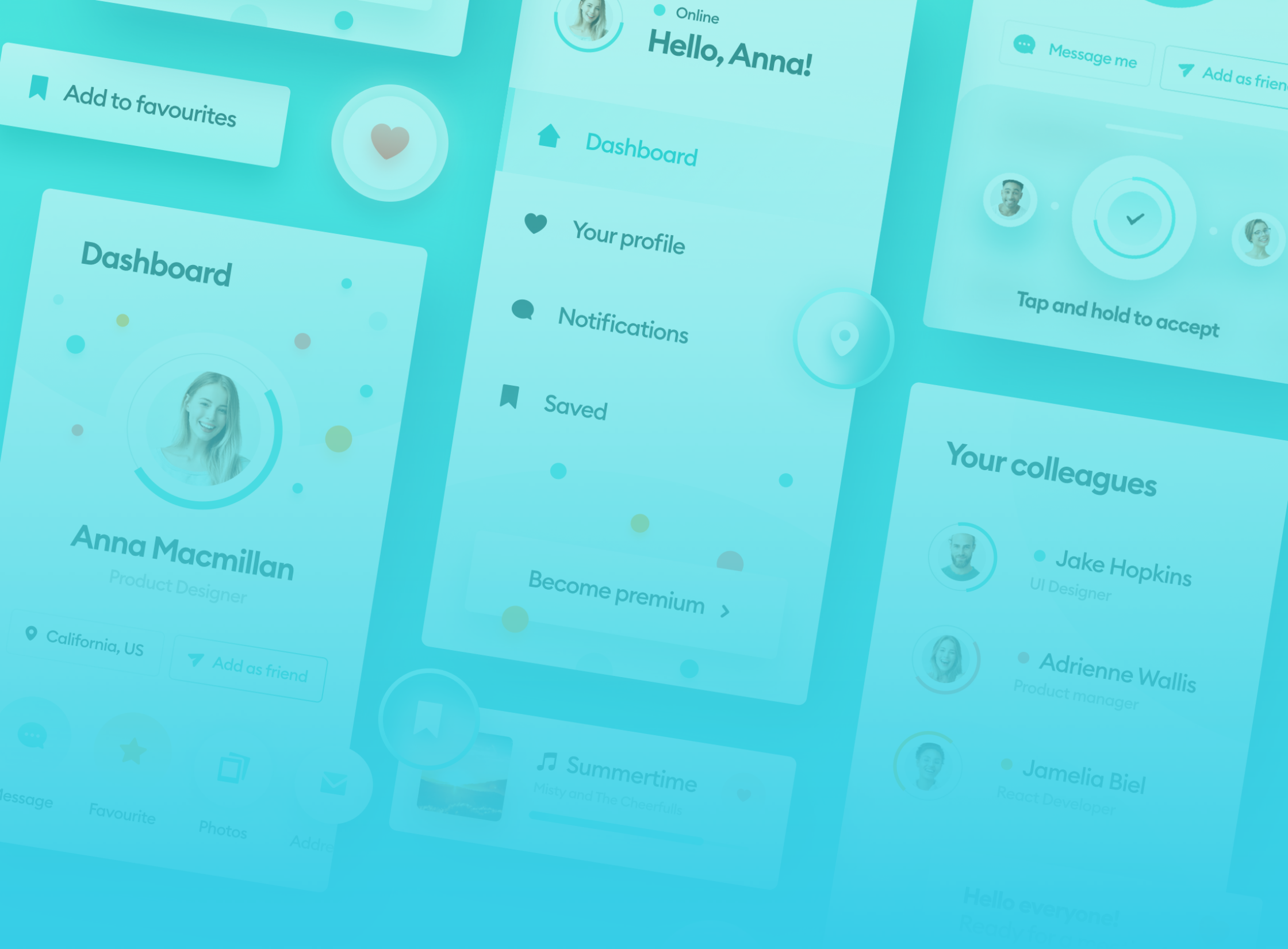
You can sign up for discounts and free downloads at:

<https://thatstartupbook.com>

Updated on March 15th.

Thanks for your feedback and comments. Here's a list of people who contributed to this version in some way.

**Albert Pawłowski,
Kinga Maciąg,
Katarzyna Kędzielska,
Marcin Kwiatkowski,
Paulina Dąbrowska,
Kamil Markiewka,
Rosie Way,
Kévin Vennitti,
Luis Cislaghi,
Axel Beyer.**



www.hype4.academy
www.designingUI.com