

Don't be a slave to the grid — give your components the space they need and don't make any compromises until it's actually necessary.

Relative sizing doesn't scale

It's tempting to believe that every part of an interface should be sized relative to one another, and that if element A needs to shrink by 25% on smaller screens, that element B should shrink by 25%, too.

For example, say you're designing an article at a large screen size. If your body copy is 18px and your headlines are 45px, it's tempting to encode that relationship by defining your headline size as `2.5em`; 2.5 times the current font size.

Team communication optimized for deep work.

Meet Sarah, the founder of CoffeeKit and waffle lover. CoffeeKit helps boutique coffee shops manage their business operations.

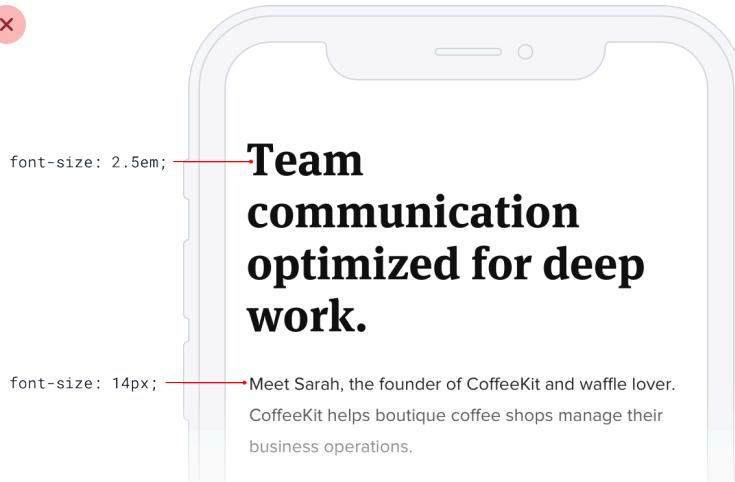
The year is 2018 and coffee hipsterdom is spreading like a stream of espresso exiting a portafilter at 9 bars of pressure.

Mustache wax and fedora hats are in short supply.

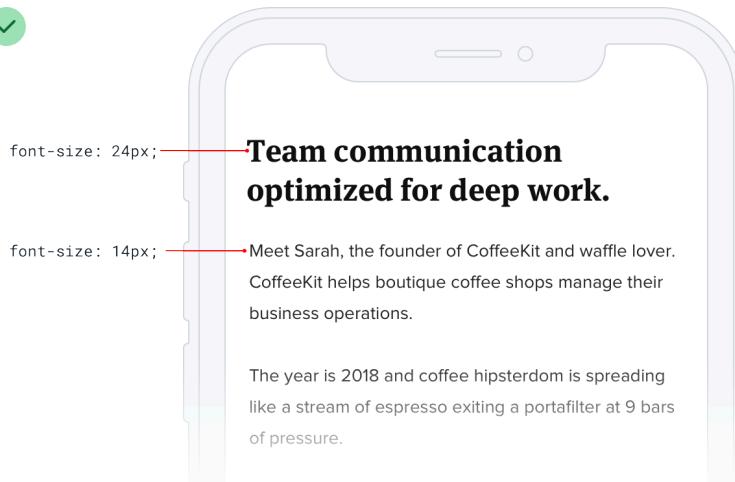
If you were to be dropped into any American urban city from outer space, you'd think you were in Portland, Oregon.

There's nothing inherently wrong with using relative units like `em`, but don't be fooled into believing that relationships defined this way can remain static — `2.5em` might be the perfect headline size on desktop but there's no guarantee that it'll be the right size on smaller screens.

Say you reduce the size of your body copy to 14px on small screens to keep the line length in check. Keeping your headlines at 2.5em means a rendered font size of 35px — way too big for a small screen!



A better headline size for small screens might be somewhere between 20px and 24px:



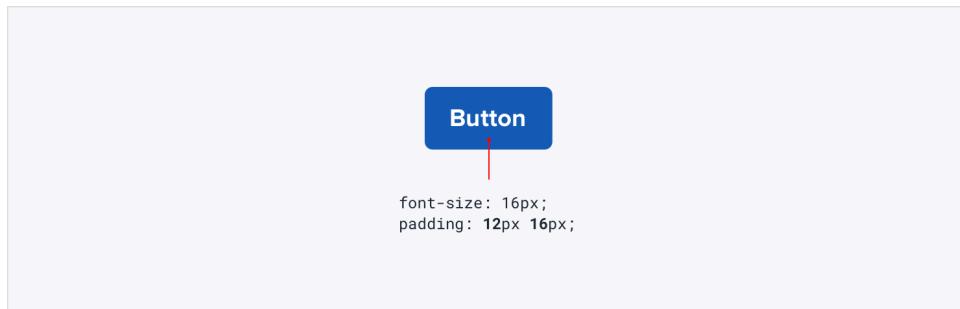
That's only 1.5-1.7x the size of the 14px body copy — a totally different relationship than what made sense on desktop screens. That means there isn't any real relationship at all, and that there's no real benefit in trying to define the headline size relative to the body copy size.

As a general rule, elements that are large on large screens need to shrink faster than elements that are already fairly small — the difference between small elements and large elements should be less extreme at small screen sizes.

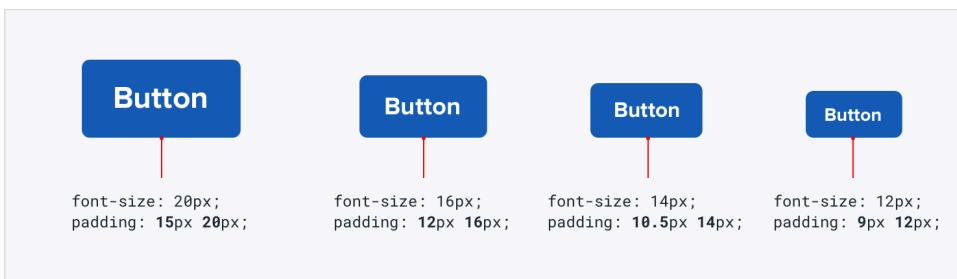
Relationships within elements

The idea that things should scale independently doesn't just apply to sizing elements at different screen sizes; it applies to the properties of a single component, too.

Say you've designed a button. It's got a 16px font size, 16px of horizontal padding, and 12px of vertical padding:

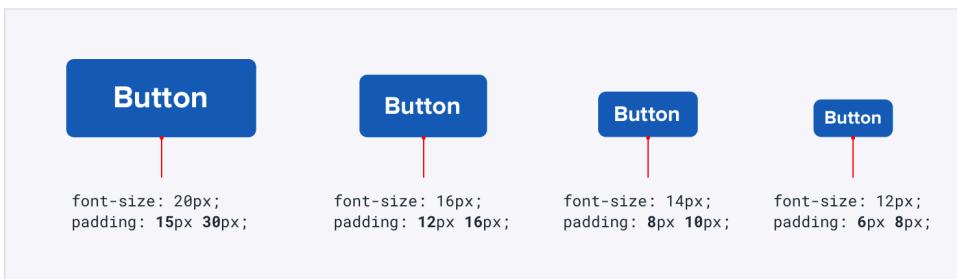


Much like the previous example, it's tempting to think that the padding should be defined in terms of the current font size. That way if you want a larger or smaller button, you only need to change the font size and the padding will update automatically, right?



This works — the buttons *do* scale up or down and preserve the same proportions. But is that what we really want?

Compare that to these buttons, where the padding gets more generous at larger sizes and disproportionately tighter at smaller sizes:

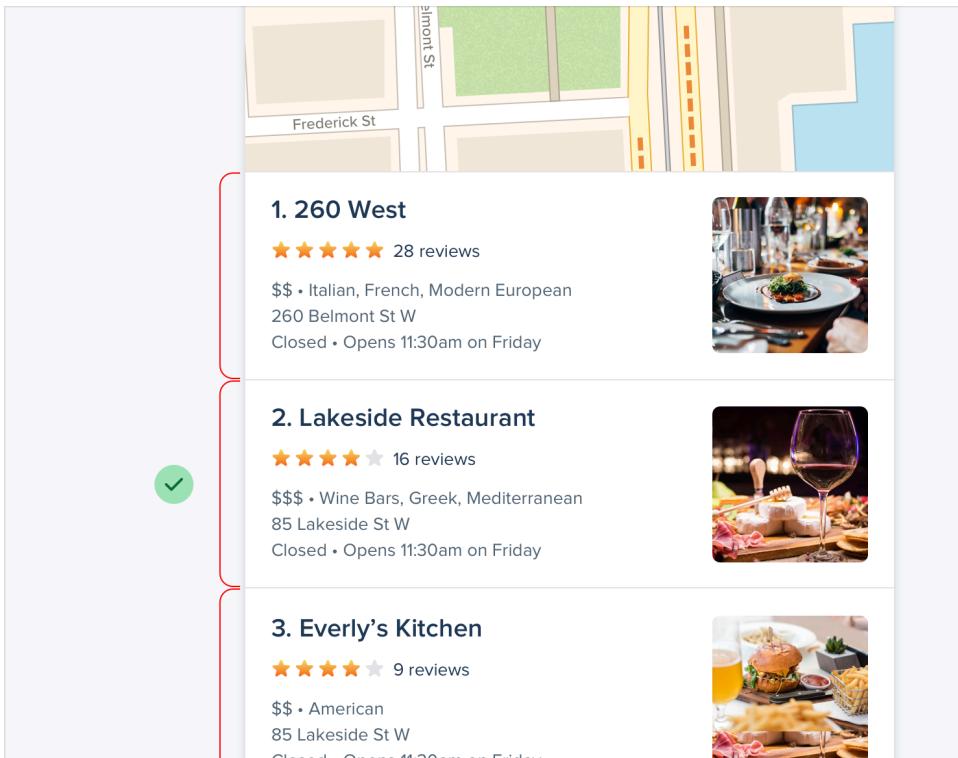


Here the large button actually *feels* like a larger button, and the small buttons actually feel like smaller buttons, not like we simply adjusted the zoom.

Let go of the idea that everything needs to scale proportionately — giving yourself the freedom to fine-tune things independently makes it a hell of a lot easier to design for multiple contexts.

Avoid ambiguous spacing

When groups of elements are explicitly separated — usually by a border or background color — it's obvious which elements belong to which group.



1. 260 West

★★★★★ 28 reviews

\$\$ • Italian, French, Modern European

260 Belmont St W

Closed • Opens 11:30am on Friday



2. Lakeside Restaurant

★★★★★ 16 reviews

\$\$\$ • Wine Bars, Greek, Mediterranean

85 Lakeside St W

Closed • Opens 11:30am on Friday



3. Everly's Kitchen

★★★★★ 9 reviews

\$\$ • American

85 Lakeside St W

Closed • Opens 11:30am on Friday



But when there isn't a visible separator, it's not always so obvious.

Say you're designing a form with stacked labels and inputs. If the margin below the label is the same as the margin below the input, the elements in the form group won't feel obviously "connected".

 **2 BILLING ADDRESS** Are you purchasing for someone else?

Full name

20px  Street address

20px  City

Province/Territory



At best the user has to work harder to interpret the UI, and at worst it means accidentally putting the wrong data in the wrong field.

The fix is to increase the space between each form group so it's clear which label belongs to which input:

 **2 BILLING ADDRESS** Are you purchasing for someone else?

Full name

20px  Street address

10px  City

Province/Territory

 Postal code

This same problem shows up in article design when there's not enough space above section headings:



designers, developers, marketers, and support folks. CoffeeKit uses Level to communicate.

24px

Monday morning has arrived.

24px

After eating a waffle, Sarah starts her workday by checking her Level Inbox. First up, Joe Gibraltar posted a direct question for



designers, developers, marketers, and support folks. CoffeeKit uses Level to communicate.

36px

Monday morning has arrived.

12px

After eating a waffle, Sarah starts her workday by checking her Level Inbox. First up, Joe Gibraltar posted a direct question for

...and in bulleted lists, when the space between bullets matches the line-height of a single bullet:



JOIN OUR GROWING COMMUNITY

- Connect with accomplished founders in your local area and have online conversations with others in the community.
- Partake in exclusive questioning sessions with investors, executives and high-profile leaders.
- Pitch your concepts and get insights from other founders.

24px

24px



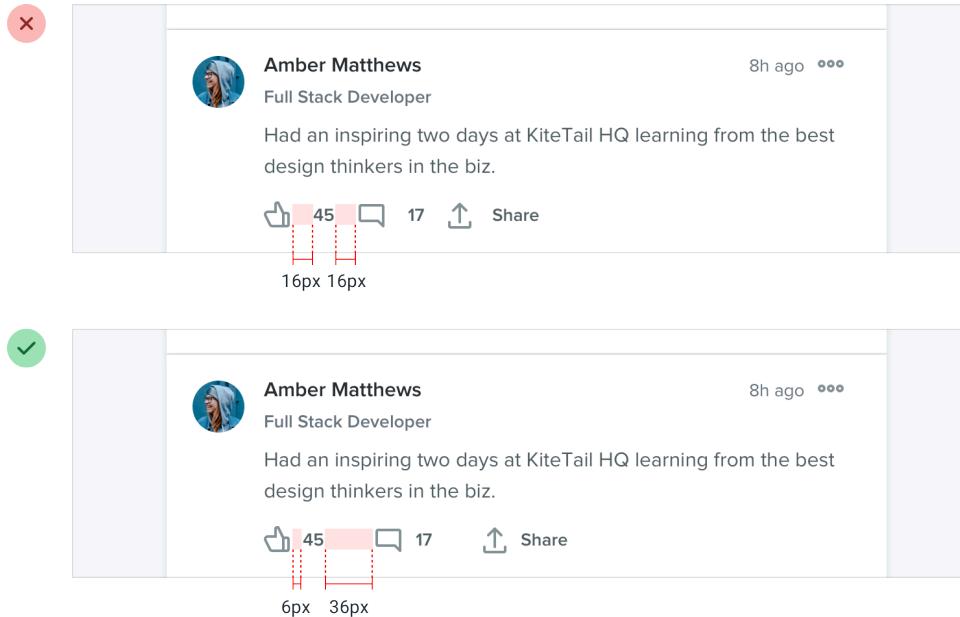
JOIN OUR GROWING COMMUNITY

- Connect with accomplished founders in your local area and have online conversations with others in the community.
- Partake in exclusive questioning sessions with investors, executives and high-profile leaders.
- Pitch your concepts and get insights from other founders.

24px

36px

It's not just vertical spacing that you have to worry about either; it's easy to make this mistake with components that are laid out horizontally, too:

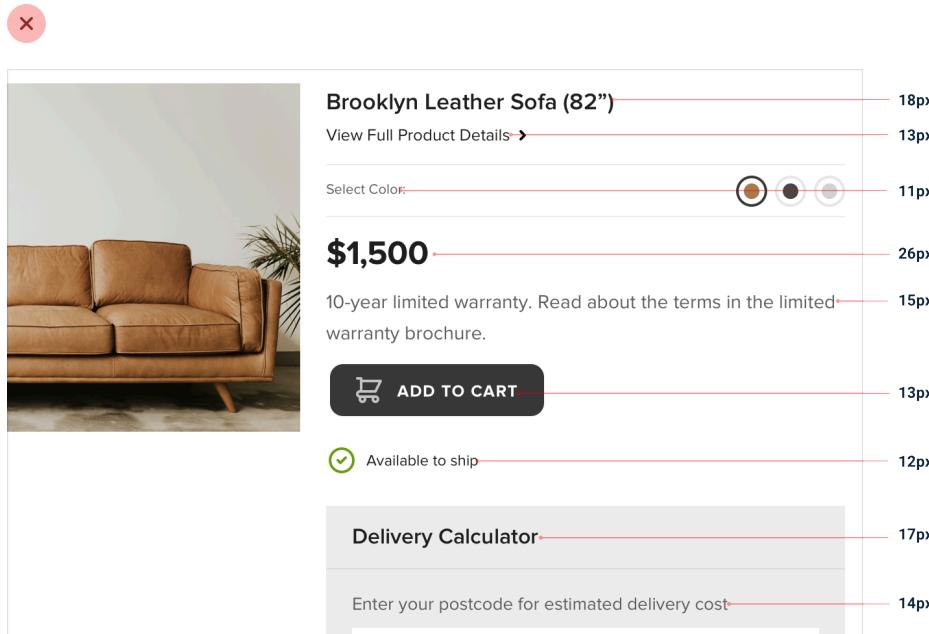


Whenever you're relying on spacing to connect a group of elements, always make sure there's more space *around* the group than there is within it — interfaces that are hard to understand always look worse.

Designing Text

Establish a type scale

Most interfaces use way too many font sizes. Unless a team has a rigid design system in place, it's not uncommon to find that every pixel value from 10px to 24px has been used in the UI *somewhere*.



Choosing font sizes without a system is a bad idea for two reasons:

1. It leads to annoying inconsistencies in your designs.
2. It slows down your workflow.

So how do you define a type system?

Choosing a scale

Just like with spacing and sizing, a linear scale won't work. Smaller jumps between font sizes are useful at the bottom of the scale, but you don't want to waste time deciding between 46px and 48px for a large headline.

Modular scales

One approach is to calculate your type scale using a *ratio*, like 4:5 (a "major third"), 2:3 (a "perfect fifth"), or perhaps the "golden ratio", 1:1.618. This is often called a "modular scale".

You start with a sensible base value (*16px is common since it's the default font size for most browsers*), apply your ratio to get the next value, then apply your ratio to *that* value to get the next value, and so on and so forth:

?

0.618em (9.88875px)

The quick brown fox jumps over the lazy dog

1em (16px)

The quick brown fox jumps over the lazy dog

1.618em (25.888px)

The quick brown fox jumps over the lazy dog

2.618em (41.8868px)

The quick brown fox jumps over th

4.236em (67.7728px)

The quick brown fox j

6.854em (109.656px)

The quick br

The mathematical purity of this approach is alluring, but in practice, it's not perfect for a couple of reasons.

1. You end up with fractional values.

Using a 16px base and 4:5 ratio, your scale will end up with lots of sizes that don't land right on the pixel, like 31.25px, 39.063px, 48.828px, etc. Browsers all handle subpixel rounding a little bit differently, so it's best to avoid fractional sizes if you can avoid it.

If you do want to use this approach, make sure you round the values yourself when defining the scale to avoid off-by-one pixel issues across browsers.

2. You usually need more sizes.

This approach can work well if you're defining a type scale for long form content like an article, but for interface design, the jumps you get using a modular scale are often a bit too limiting.

With a (*rounded*) 3:4 type scale, you get sizes like 12px, 16px, 21px, and 28px. While this might not seem too limiting on the surface, in practice you're going to wish you had a size between 12px and 16px, and another between 16px and 21px.

You could use a tighter ratio like 8:9, but at this point you're just trying to pick a scale that happens to match the sizes you already know you want.

Hand-crafted scales

For interface design, a more practical approach is to simply pick values by hand. You don't have to worry about subpixel rounding errors this way, and you have total control over which sizes exist instead of outsourcing that job to some mathematical formula.

Here's an example of a scale that works well for most projects and aligns nicely with the spacing and sizing scale recommended in "Establishing a spacing and sizing system":



12px The quick brown fox jumps over the lazy dog

14px The quick brown fox jumps over the lazy dog

16px The quick brown fox jumps over the lazy dog

18px The quick brown fox jumps over the lazy dog

20px The quick brown fox jumps over the lazy dog

24px The quick brown fox jumps over the lazy dog

30px The quick brown fox jumps over the lazy dog

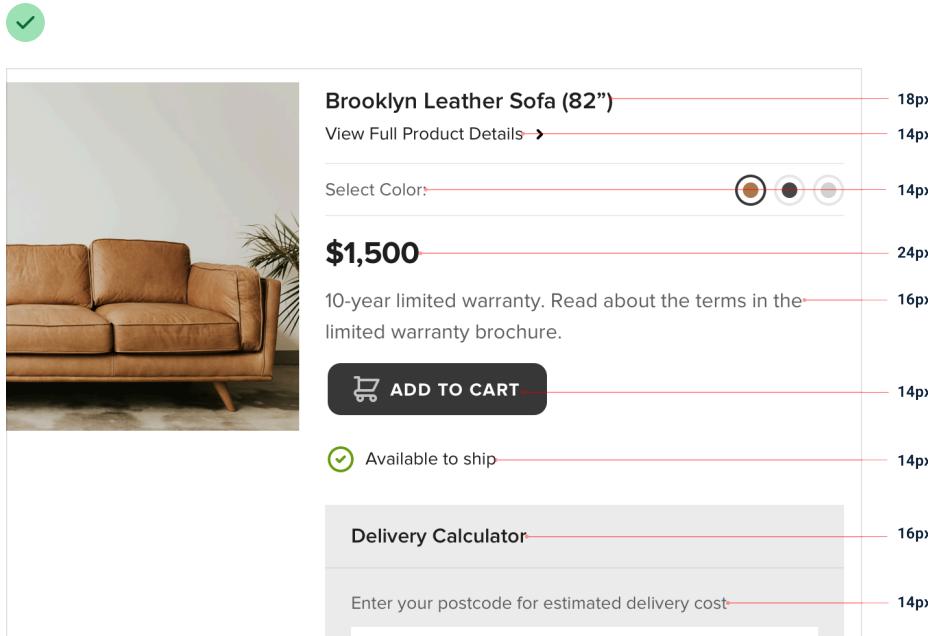
36px The quick brown fox jumps over the lazy dog

48px The quick brown fox jumps over th

60px The quick brown fox jumps

72px The quick brown fox ju

It's constrained just enough to speed up your decision making, but isn't so limited as to make you feel like you're missing a useful size.



Avoid em units

When you're building a type scale, don't use *em* units to define your sizes.

Because *em* units are relative to the current font size, the computed font size of nested elements is often not actually a value in your scale.

For example, say you've defined an *em*-based type scale like this:

.75em The quick brown fox jumps over the lazy dog

.875em The quick brown fox jumps over the lazy dog

1em The quick brown fox jumps over the lazy dog

1.25em The quick brown fox jumps over the lazy dog

1.5em The quick brown fox jumps over the lazy dog

If you give an element a font size of 1.25em (*20px by default*), inside of that element 1em is now equal to 20px. That means that if you give one of the *nested* elements a font size of .875em, the actual computed font size is 17.5px, not a value from your type scale!

Stick to *px* or *rem* units — it's the only way to guarantee you're actually sticking to the system.

Use good fonts

With thousands of different typefaces out there to choose from, separating the good from the bad can be an intimidating task.

Aa	Freight Sans Extra Bold	Aa	Harmonia Sans Regular	Aa	Work Sans Semibold	Aa	Proxima Nova Bold	Aa	I R
Aa	Aktiv Grotesk Regular	Aa	Avenir Next Demi Bold	Aa	Roboto Regular	Aa	Museo Sans 300	Aa	N R
Aa	Akagi Pro Regular	Aa	Din Pro Regular	Aa	Meta Serif Book	Aa	Encode Sans Black	Aa	Ir R
Aa	Futura Bold	Aa	Franklin Gothic Book	Aa	Freight Text Regular	Aa	Jubilat Light Italic	Aa	E R

Developing an eye for all of the details that make a good typeface can take years. You probably don't have years, so here are a few tricks you can use to start picking out high quality typefaces right away.

Play it safe

For UI design, your safest bet is a fairly neutral sans-serif — think something like Helvetica.

If you really don't trust your own taste, one great option is to rely on the system font stack:

```
-apple-system, Segoe UI, Roboto, Noto Sans, Ubuntu, Cantarell, Helvetica Neue;
```

It might not be the most ambitious choice, but at least your users will already be used to seeing it.

Ignore typefaces with less than five weights

This isn't always true, but as a general rule, typefaces that come in a lot of different weights tend to be crafted with more care and attention to detail than typefaces with fewer weights.

Many font directories (*like Google Fonts*) will let you filter by "number of styles", which is a combination of the available weights as well as the italic variations of those weights.

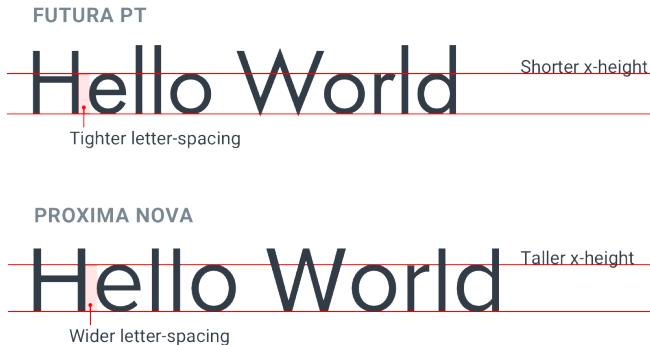
A great way to limit the number of options you have to choose from is to crank that up to 10+ (to account for italics):

The screenshot shows the Google Fonts website interface. At the top, there's a navigation bar with links for 'Directory', 'Featured', 'Articles', and 'About'. A search bar is on the right with a magnifying glass icon and the placeholder 'Search'. Below the navigation, there's a preview area with sample text: 'The quick brown fox jumps over the lazy dog'. To the left of this text is a dropdown menu set to 'Sans Serif' and a 'Classification' dropdown also set to 'Sans Serif'. On the right, there are several filtering options: 'Languages' (set to 'All Languages'), 'Number of styles' (set to '10+' with a slider), 'Thickness' (a slider with a checked checkbox), 'Slant' (a slider with a checked checkbox), and 'Width' (a slider with a checked checkbox). At the bottom left, there's a section for the font 'Acumin Pro' with '90 Styles', a 'Activate' toggle switch, and icons for heart, code, and another activate switch. The main preview area shows the text 'The quick brown fox jumps over the lazy dog' in a sans-serif font.

On Google Fonts specifically, that cuts out 85% of the available options, leaving you with less than 50 sans-serifs to choose from.

Optimize for legibility

When someone designs a font family, they are usually designing it for a specific purpose. Fonts meant for headlines usually have tighter letter-spacing and shorter lowercase letters (*a shorter x-height*), while fonts meant for smaller sizes have wider letter-spacing and taller lowercase letters.



Keep this in mind and avoid using condensed typefaces with short x-heights for your main UI text.

Trust the wisdom of the crowd

If a font is popular, it's probably a good font. Most font directories will let you sort by popularity, so this can be a great way to limit your choices.

This is especially useful when you're trying to pick out something other than a neutral UI typeface. Picking a nice serif with some personality for example can be tough.

Sort by Popularity

Font	Styles	Sample Text
EB Garamond	10 Styles	The quick brown fox jumps over the lazy dog
Alegreya	10 Styles	The quick brown fox jumps over the lazy dog
Cormorant Garamond	10 Styles	The quick brown fox jumps over the lazy dog

Leveraging the collective decision-making power of thousands of other people can make it a lot easier.

Steal from people who care

Inspect some of your favorite sites and see what typefaces they are using.

Styles	Computed	Event Listeners
.title { font-family: "Graphik", -apple- system, Segoe UI, Roboto, Noto Sans, Ubuntu, Cantarell, Helvetica Neue; }		
html, body, div, span, applet, object, iframe, h1, h2, h3, h4, h5, h6, p, blockquote, pre, a, abbr, acronym, address, big, cite, code, del, dfn, em, font, img, ins, kbd, q, s, samp, small, strike, strong, sub, sup, tt, var, b, u, i, center, dl, dt, dd, ol, ul, li, fieldset, form		

There are a lot of great design teams out there full of people with *really* strong opinions about typography, and they'll often choose great fonts that you might have never found using some of the safer approaches outlined above.

Developing your intuition

Once you start paying closer attention to the typography on well-designed sites, it's not long before you feel pretty comfortable labeling a typeface as awesome or terrible.

You're gonna be a type snob soon enough, but the advice outlined above will help get you by in the meantime.

Keep your line length in check

When styling paragraphs, it's easy to make the mistake of fitting the text to your layout instead of trying to create the best reading experience.

Usually this means lines that are too long, making text harder to read.



— ~120 characters per line —

Call me Ishmael. Some years ago—never mind how long precisely—having little or no money in my purse, and nothing particular to interest me on shore, I thought I would sail about a little and see the watery part of the world. It is a way I have of driving off the spleen and regulating the circulation. Whenever I find myself growing grim about the mouth; whenever it is a damp, drizzly November in my soul; whenever I find myself involuntarily pausing before coffin warehouses, and bringing up the rear of every funeral I meet; and especially whenever my hypos get such an upper hand of me, that it requires a strong moral principle to prevent me from deliberately stepping into the street, and methodically knocking people's hats off—then, I account it high time to get to sea as soon as I can. This is my substitute for pistol and ball. With a philosophical flourish Cato throws himself upon his sword; I quietly take to the

For the best reading experience, make your paragraphs wide enough to fit between 45 and 75 characters per line. The easiest way to do this on the web is using *em* units, which are relative to the current font size. A width of 20-35em will get you in the right ballpark.

45 - 55 characters per line

There was no possibility of taking a walk that day. We had been wandering, indeed, in the leafless shrubbery an hour in the morning; but since dinner (Mrs. Reed, when there was no company, dined early) the cold winter wind had brought with it clouds so sombre, and a rain so penetrating, that further out-door exercise was now out of the question.

55 - 65 characters per line

There was no possibility of taking a walk that day. We had been wandering, indeed, in the leafless shrubbery an hour in the morning; but since dinner (Mrs. Reed, when there was no company, dined early) the cold winter wind had brought with it clouds so sombre, and a rain so penetrating, that further out-door exercise was now out of the question.

65 - 75 characters per line

There was no possibility of taking a walk that day. We had been wandering, indeed, in the leafless shrubbery an hour in the morning; but since dinner (Mrs. Reed, when there was no company, dined early) the cold winter wind had brought with it clouds so sombre, and a rain so penetrating, that further out-door exercise was now out of the question.

Going a bit wider than 75 characters per line can sometimes work too, but be aware that you're entering risky territory — stick to the 45-75 range if you want to play it safe.

Dealing with wider content

If you're mixing paragraph text with images or other large components, you should still limit the paragraph width even if the overall content area needs to be wider to accommodate the other elements.



The one-size-fits-all platform

Need a simple website or an online store? Build a beautiful website without touching code. We'll provide you with all of the tools and insights to grow your online business.



Beautiful templates

Who said you need to hire a professional



Sell online

Whether you're getting started or already



Grow your Audience

Start a social media campaign or set up a

max-width: none;

✓

The one-size-fits-all platform

Need a simple website or an online store? Build a beautiful website without touching code. We'll provide you with all of the tools and insights to grow your online business.

 Beautiful templates Who said you need to hire a professional	 Sell online Whether you're getting started or already	 Grow your Audience Start a social media campaign or set up a
---	--	---

max-width: 34em;

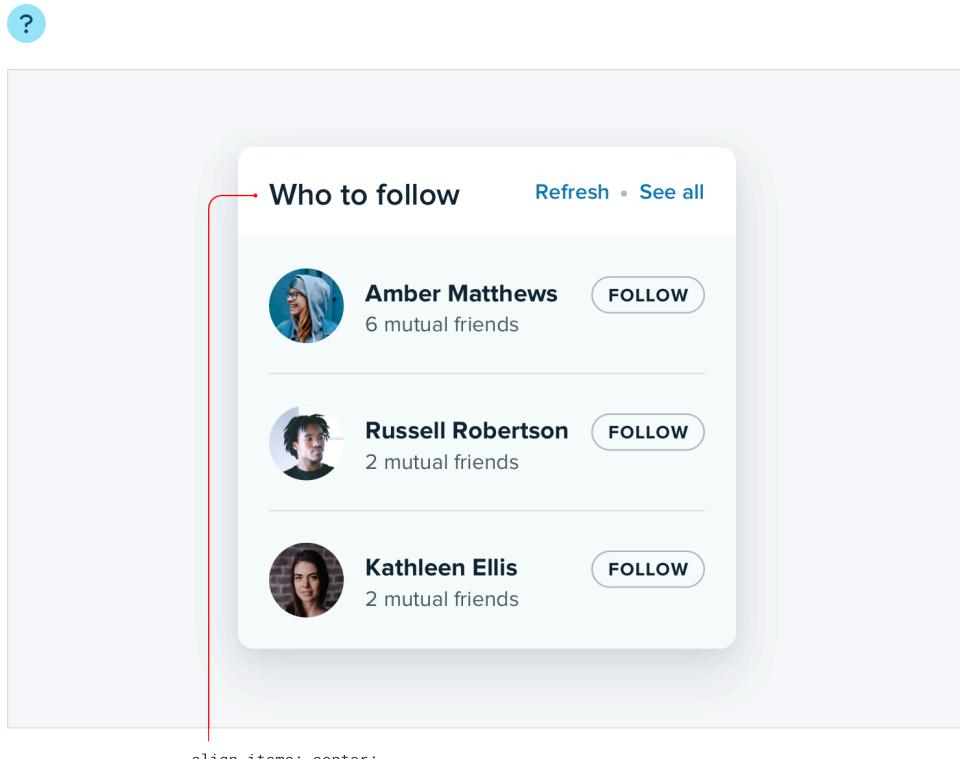
It might seem counterintuitive at first to use different widths in the same content area, but the result almost always looks more polished.

Baseline, not center

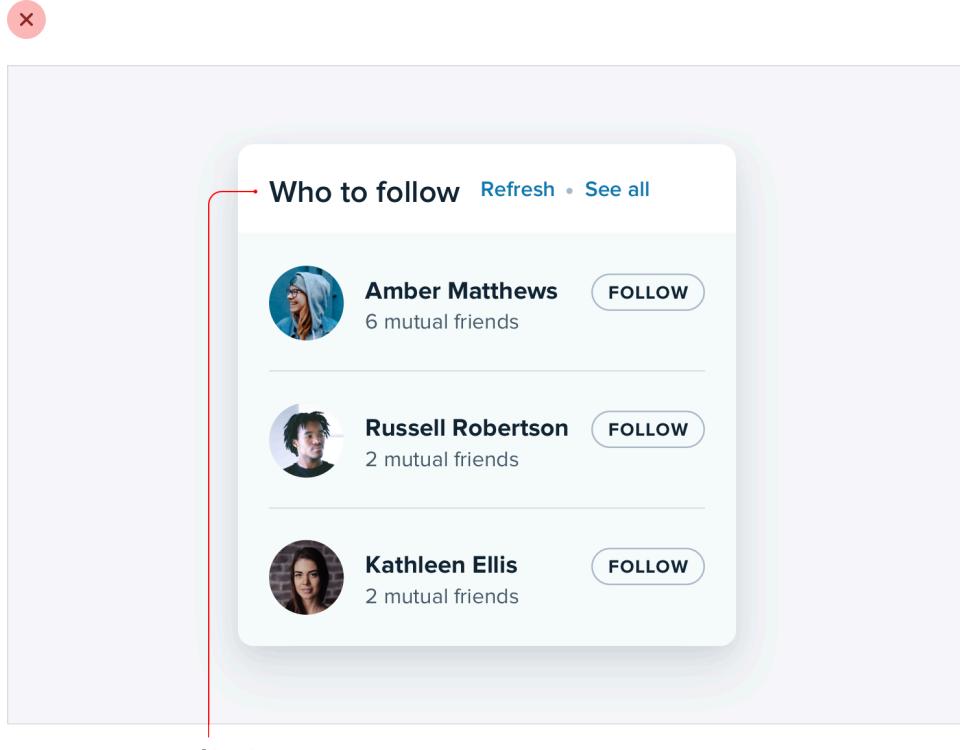
There are a lot of situations where it makes sense to use multiple font sizes to create hierarchy on a single line.

For example, maybe you're designing a card that has a large title in the top left and a smaller list of actions in the top right.

When you're mixing font sizes like this, your instinct might be to vertically center the text for balance:



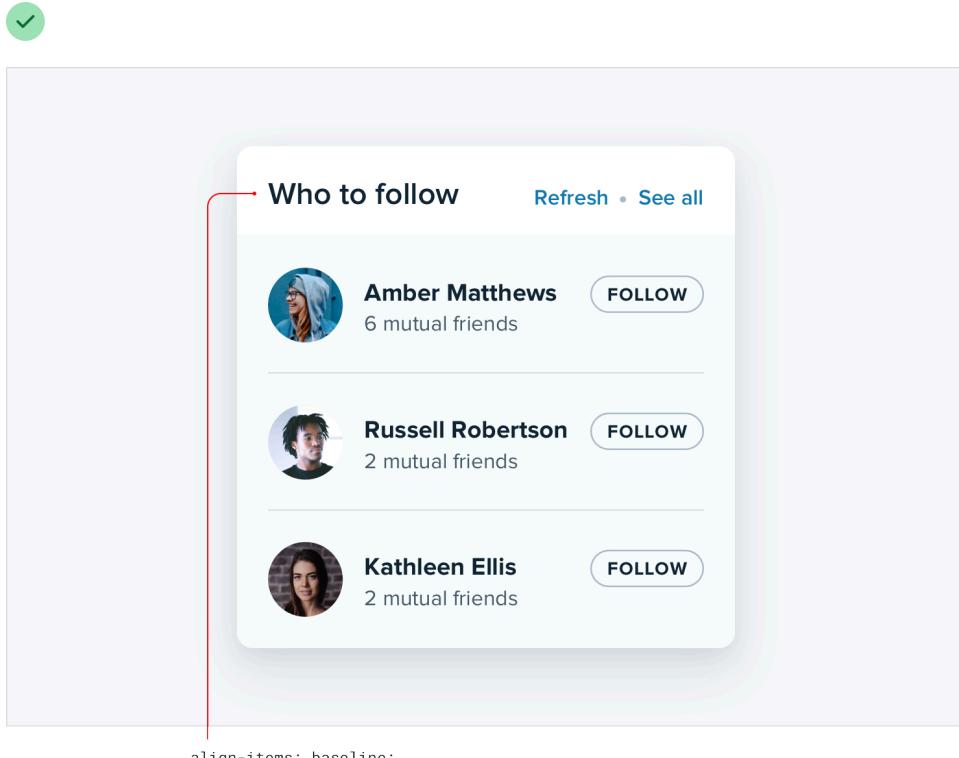
When there's a decent amount of space between the different font sizes it often won't look bad enough to catch your attention, but when the text is close together the awkward alignment becomes more obvious:



A better approach is to align mixed font sizes by their *baseline*, which is the imaginary line that letters rest on:



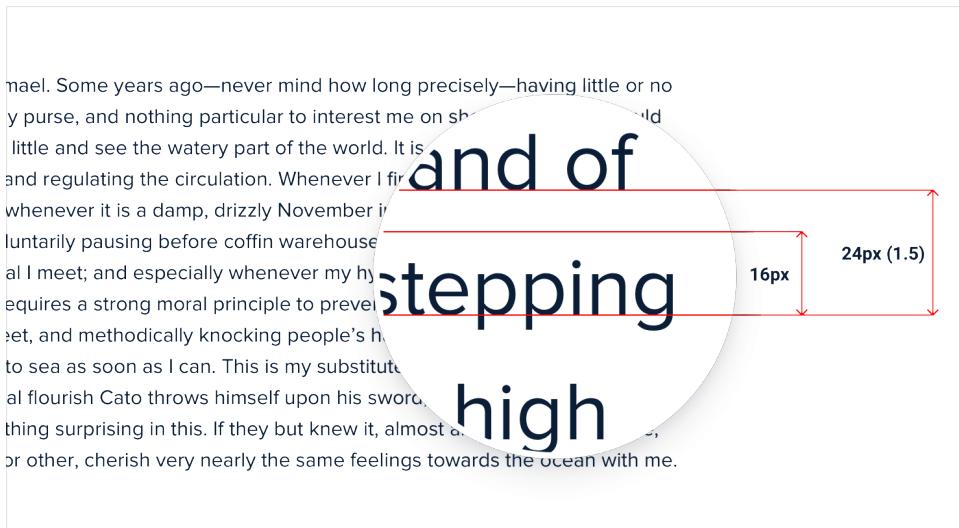
When you align mixed font sizes by their baseline, you're taking advantage of an alignment reference that your eyes already perceive.



The result is a simpler, cleaner look than what you get when you center two pieces of text and offset their baselines.

Line-height is proportional

You might have heard the advice that a line-height of about 1.5 is a good starting point from a readability perspective.



While that's not necessarily untrue, choosing the right line-height for your text is a bit more complicated than just using the same value across the board in all situations.

Accounting for line length

The reason we add space between lines of text is to make it easy for the reader to find the next line when the text wraps. Have you ever accidentally read the same line of text twice, or accidentally skipped a line? The line-height was probably too short.

When lines of text are spaced too tightly, it's easy to finish reading a line of text at the right edge of a page then jump your eyes all the way back to the left edge only to be unsure which line is next.



Call me Ishmael. Some years ago—never mind how long precisely—having little or no money in my purse, and nothing particular to interest me on shore, I thought I would sail about a little and see the watery part of the world. It is a way I have of driving off the spleen and regulating the circulation. Whenever I find myself growing grim about the mouth; whenever it is a damp, drizzly November in my soul; whenever I find myself involuntarily pausing before coffin warehouses, and bringing up the rear of every funeral I meet; and especially whenever my hypos get such an upper hand of me, that it requires a strong moral principle to prevent me from deliberately stepping

This problem is magnified when lines of text are long. The further your eyes have to jump horizontally to read the next line, the easier it is to lose your place.

That means that your line-height and paragraph width should be proportional — narrow content can use a shorter line-height like 1.5, but wide content might need a line-height as tall as 2.



There was no possibility of taking a walk that day. We had been wandering, indeed, in the leafless shrubbery an hour in the morning; but since dinner (Mrs. Reed, when there was no company, dined early) the cold winter

1.5

There was no possibility of taking a walk that day. We had been wandering, indeed, in the leafless shrubbery an hour in the morning; but since dinner (Mrs. Reed, when there was no company, dined early) the cold winter wind had brought with it clouds so sombre, and a rain so penetrating, that further out-door exercise was now out of the question.

2

Accounting for font size

Line length isn't the only factor in choosing the right line-height — font size has a big impact as well.

When text is small, extra line spacing is important because it makes it a lot easier for your eyes to find the next line when the text wraps.



Call me Ishmael. Some years ago—never mind how long precisely—having little or no money in my purse, and nothing particular to interest me on shore, I thought I would sail about a little and see the watery part of the world. It is a way I have of driving off the spleen and regulating the circulation. Whenever I find myself growing grim about the mouth; whenever it is a damp, drizzly November in my soul; whenever I find myself involuntarily pausing before coffin warehouses, and bringing up the rear of every funeral I meet; and especially whenever my hypos get such an upper hand of me, that it requires a strong moral principle to prevent me from deliberately stepping into the street, and methodically knocking people's hats off—then, I account it high time to get to sea as soon as I can. This is my substitute for pistol and ball. With a philosophical flourish Cato throws himself upon his sword; I quietly take to the chin. There is nothing surprising in this. If they but knew it, almost all men in

1.25 ↓



Call me Ishmael. Some years ago—never mind how long precisely—having little or no money in my purse, and nothing particular to interest me on shore, I thought I would sail about a little and see the watery part of the world. It is a way I have of driving off the spleen and regulating the circulation. Whenever I find myself growing grim about the mouth; whenever it is a damp, drizzly November in my soul; whenever I find myself involuntarily pausing before coffin warehouses, and bringing up the rear of every funeral I meet; and especially whenever my hypos get such an upper hand of me, that it requires a strong moral principle to prevent me from deliberately stepping into the street, and methodically knocking people's hats off—then, I account it high time to get to

1.75 ↑

But as text gets larger, your eyes don't need as much help. This means that for large headline text you might not need any extra line spacing, and a line-height of 1 is perfectly fine.



Team communication optimized for deep work.

1.5

Meet Sarah, the founder of CoffeeKit and waffle lover. CoffeeKit



Team communication optimized for deep work.

1

Meet Sarah, the founder of CoffeeKit and waffle lover. CoffeeKit

helps boutique coffee shops manage their business operations

Line-height and font size are *inversely* proportional — use a taller line-height for small text and a shorter line-height for large text.

Not every link needs a color

When you're including a link in a block of otherwise non-link text, it's important to make sure that the link stands out and looks clickable.

Nothing quite compares to greenfield software development. The canvas is blank and you finally have the opportunity to do it "[The Right Way](#)" from the ground up. If you've been building web apps for a while, you've undoubtedly found yourself working with technologies that you'd never use again, given the luxury of a blank canvas. And if you follow the open source world, there's probably a brand-spanking-new [boutique framework](#) you've been itching to take for a spin.

With all the excitement of a blank canvas comes an equal amount of anxiety. You know you are one ill-advised choice away from being stuck with the "imperfect" tech stack. You have a hunch about what you want to use, but being the dutiful

But when you're designing an interface where almost everything is a link, using a treatment designed to make links "pop" in paragraph text can be really overbearing.



[Violinist stunsarians with an electrifying...](#)

Mike Price
1.9M views • 8 months ago



[Artisan glass blowing artist makes incredible creations](#)

Museum of Glass
756K views • 2 years ago



[How to make a perfect loaf of bread from scratch](#)

Bread Masters
3M views • 4 months ago



[Man makes not missin...](#)

Free Throw
104K views

Instead, emphasize most links in a more subtle way, like by just using a heavier font weight or darker color.



 5:45	 3:30	 8:21	
ed violinist stuns rians with an electrifying...	Artisan glass blowing artist makes incredible creations	How to make a perfect loaf of bread from scratch	Man make not missin
anie Price ws • 8 months ago	Museum of Glass 756K views • 2 years ago	Bread Masters 3M views • 4 months ago	Free Throw 104K views

Some links might not even need to be emphasized by default at all. If you've got links in your interface that are really ancillary and not part of the main path a user takes through the application, consider adding an underline or changing the color *only on hover*.



 5:45	 3:30	 8:21	
ed violinist stuns rians with an electrifying...	Artisan glass blowing artist makes incredible creations	How to make a perfect loaf of bread from scratch	Man make not missin
anie Price ws • 8 months ago	Museum of Glass 756K views • 2 years ago	Bread Masters 3M views • 4 months ago	Free Throw 104K views

They'll still be discoverable to any users who think to try, but won't compete for attention with more important actions on the page.

Align with readability in mind

In general, text should be aligned to match the direction of the language it's written in. For English (and most other languages), that means that the vast majority of text should be left-aligned.

Amanda Wagner
Human Resources Specialist
Join us on November 27th for our 1st Dribbble meetup in Waterloo! Calling all designers, UI/UXers, illustrators and other creatives from the KW and surrounding area together to talk shop, mingle, and walk away with some great swag!

8h ago

23 17 Share

text-align: left;

Other alignment options do have their place though, you just need to use them effectively.

Don't center long form text

Center-alignment can look great for headlines or short, independent blocks of text.

Beautiful templates
Our templates are all you need to stand out from the rest of the competition.

Sell online
We provide you with everything you need to run a successful business online.

Grow your Audience
Our marketing tools enable you to engage directly with your audience.

But if something is longer than two or three lines, it will almost always look better left-aligned.







The one-size-fits-all solution
Build a beautiful website without touching code. We'll provide you with all of the tools and insights to grow your online business. Our templates are all you need to stand out from the rest of the competition.



The one-size-fits-all solution
Build a beautiful website without touching code. We'll provide you with all of the tools and insights to grow your online business. Our templates are all you need to stand out from the rest of the competition.

If you've got a few blocks of text you want to center but one of them is a bit too long, the easiest fix is to rewrite the content and make it shorter:









Beautiful templates
Our templates are all you need to stand out from the rest of the competition.



Sell online
Whether you're getting started or already well established, we provide you with everything you need to run a successful business online.



Grow your A
Our marketing tools engage directly with



Beautiful templates
Our templates are all you need to stand out from the rest of the competition.



Sell online
We provide you with everything you need to run a successful business online.



Grow your A
Our marketing tools engage directly with

Not only will it fix the alignment issue, it will make your design feel more consistent, too.

Right-align numbers

If you're designing a table that includes numbers, right-align them.

SYMBOL	PRICE	CHG	MKT CAP
APPL	\$174.72	-1.97	1.007T
GOOGL	\$1,068.73	+17.91	874.4B
AMZN	\$1,699.19	+30.79	830.9B
MSFT	\$1,699.19	+0.67	838.2B

SYMBOL	PRICE	CHG	MKT CAP
APPL	\$174.72	-1.97	1.007T
GOOGL	\$1,068.73	+17.91	874.4B
AMZN	\$1,699.19	+30.79	830.9B
MSFT	\$1,699.19	+0.67	838.2B

When the decimal in a list of numbers is always in the same place, they're a lot easier to compare at a glance.

Hyphenate justified text

Justified text looks great in print and can work well on the web when you're going for a more formal look, but without special care, it can create a lot of awkward gaps between words:

I recently set out to implement user registration for a project I'm working on in Elixir/Phoenix. It wasn't long before I encountered a challenge that I have stumbled upon with every other ORM library: accepting a collection of form inputs and saving it across multiple (related) records in the database.

Awkward gaps between words
hyphens: none;

To avoid this, whenever you justify text, you should also enable hyphenation:



I recently set out to implement user registration for a project I'm working on in Elixir/Phoenix. It wasn't long before I encountered a challenge that I have stumbled upon with every other ORM library: accepting a collection of form inputs and saving it across multiple (related) records in the database.

Gaps between words reduced

hyphens: auto;

Justified text works best in situations where you're trying to mimic a print look, perhaps for an online magazine or newspaper. Even then, left aligned text works great too, so it's really just a matter of preference.

Use letter-spacing effectively

When styling text, a lot of effort is put into getting the weight, color, and line-height just right, but it's easy to forget that letter-spacing can be tweaked, too.

Tight letter-spacing

letter-spacing: -0.05em;

There was no possibility of taking a walk that day. We had been wandering, indeed, in the leafless shrubbery an hour in the morning; but since dinner (Mrs. Reed, when there was no company, dined early) the cold winter wind had brought with it clouds so sombre, and a rain so penetrating, that further out-door exercise was now out of the question.

Normal letter-spacing

letter-spacing: 0;

There was no possibility of taking a walk that day. We had been wandering, indeed, in the leafless shrubbery an hour in the morning; but since dinner (Mrs. Reed, when there was no company, dined early) the cold winter wind had brought with it clouds so sombre, and a rain so penetrating, that further out-door exercise was now out of the question.

Wide letter-spacing

letter-spacing: 0.05em;

There was no possibility of taking a walk that day. We had been wandering, indeed, in the leafless shrubbery an hour in the morning; but since dinner (Mrs. Reed, when there was no company, dined early) the cold winter wind had brought with it clouds so sombre, and a rain so penetrating, that further out-door exercise was now out of the question.

As a general rule, you should trust the typeface designer and leave letter-spacing alone. That said, there are a couple of common situations where adjusting it can improve your designs.

Tightening headlines

When someone designs a font family, they design it with a purpose in mind.

A family like Open Sans is designed to be highly legible even at small sizes, so the built-in letter-spacing is a lot wider than a family like Oswald which is designed for headlines.

Open Sans

Team Communication Made Easy

Oswald

Team Communication Made Easy

If you want to use a family with wider letter-spacing for headlines or titles, it can often make sense to decrease the letter-spacing to mimic the condensed look of a purpose-built headline family:

Open Sans

Team Communication Made Easy

Open Sans

Team Communication Made Easy — letter-spacing: -0.05em;

Avoid trying to make this work the other way around though — headline fonts rarely work well at small sizes even if you increase the letter spacing.

Improving all-caps legibility

The letter-spacing in most font families is optimized for normal “sentence case” text — a capital letter followed by mostly lowercase letters.

Lowercase letters have a lot of variety visually. Letters like *n*, *v*, and *e* fit entirely within a typeface’s x-height, other letters like *y*, *g*, and *p* have descenders that poke out below the baseline, and letters like *b*, *f*, and *t* have ascenders that extend above.



All-caps text on the other hand isn’t so diverse. Since every letter is the same height, using the default letter-spacing often leads to text that is harder to read because there are fewer distinguishing characteristics between letters.

For that reason, it often makes sense to increase the letter-spacing of all-caps text to improve readability:

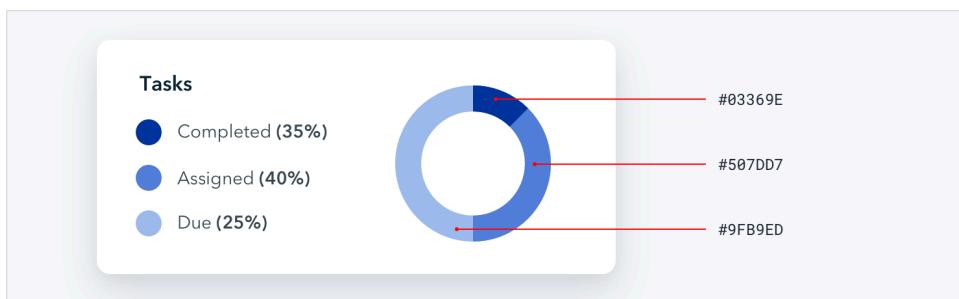


Working with Color

Ditch hex for HSL

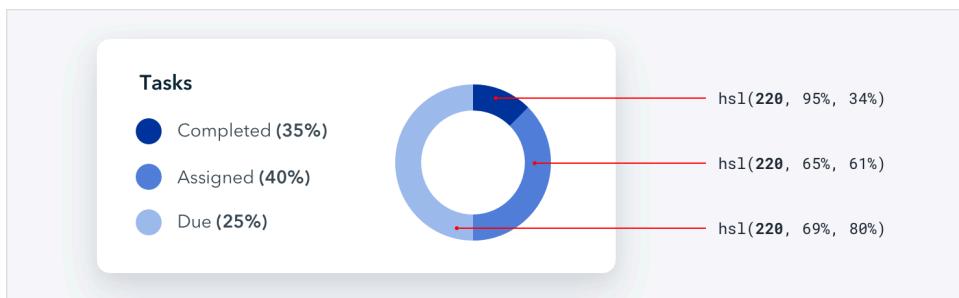
Hex and RGB are the most common formats for representing color on the web, but they're not the most useful.

Using hex or RGB, colors that have a lot in common visually look nothing alike in code.

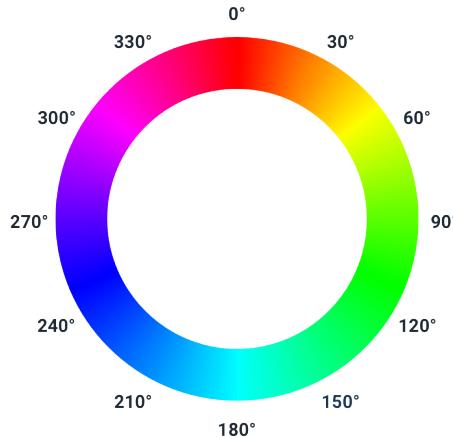


HSL fixes this by representing colors using attributes the human-eye intuitively perceives: *hue*, *saturation*, and *lightness*.

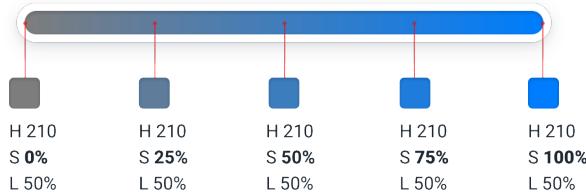
Hue is a color's position on the color wheel — it's the attribute of a color that lets us identify two colors as "blue" even if they aren't identical.



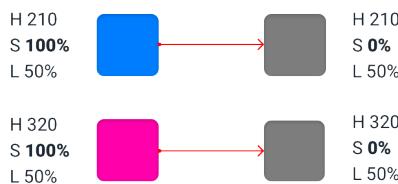
Hue is measured in degrees, where 0° is red, 120° is green, and 240° is blue.



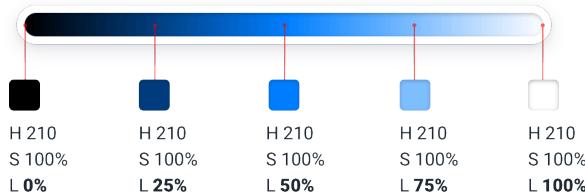
Saturation is how colorful or vivid a color looks. 0% saturation is grey (no color), and 100% saturation is vibrant and intense.



Without saturation, hue is irrelevant — rotating the hue when saturation is 0% doesn't actually change the color at all.



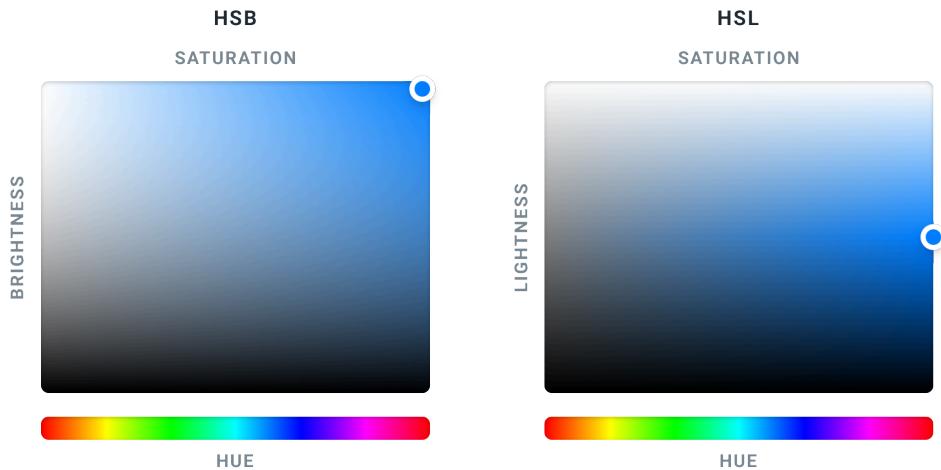
Lightness is just what it sounds like — it measures how close a color is to black or to white. 0% lightness is pure black, 100% lightness is pure white, and 50% lightness is a pure color at the given hue.



HSL vs. HSB

Don't confuse HSL for HSB — *lightness* in HSL is not the same than *brightness* in HSB.

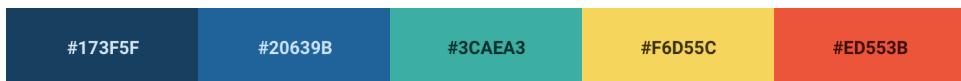
In HSB, 0% brightness is always black, but 100% brightness is only white when the saturation is 0%. When saturation is 100%, 100% brightness in HSB is the same as 100% saturation and 50% lightness in HSL.



HSB is more common than HSL in design software, but browsers only understand HSL, so if you're designing for the web, HSL should be your weapon of choice.

You need more colors than you think

Ever used one of those color palette generators where you pick a starting color, tweak some options, and are then bestowed the five perfect colors you should use to build your website?

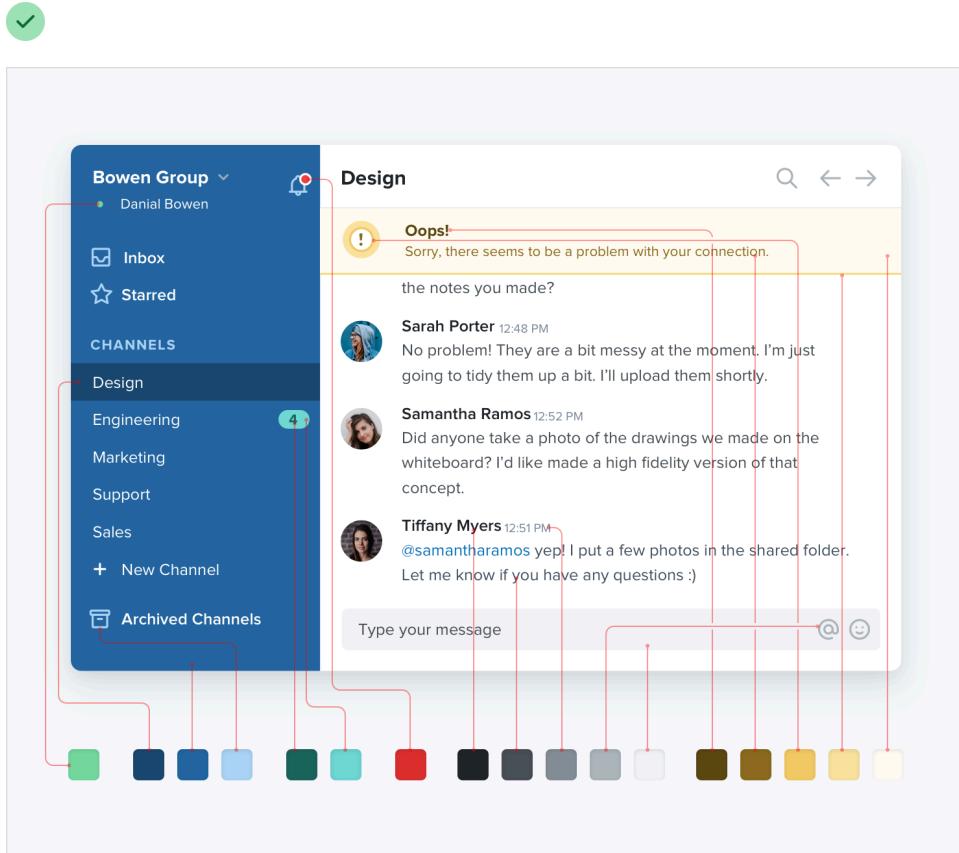


This calculated approach to picking the perfect color scheme is extremely seductive, but it's not very useful unless you want your site to look like this:

The screenshot shows a messaging application interface. On the left, there's a dark blue sidebar with the title "Bowen Group" and a dropdown arrow. Below it are sections for "Inbox" (with a red notification dot), "Starred", and "CHANNELS". Under "CHANNELS", the "Design" channel is selected, highlighted with a dark blue background. Other channels listed are "Engineering", "Marketing", "Support", "Sales", and "+ New Channel". At the bottom of the sidebar is "Archived Channels". The main area is titled "Design" and shows a message from "Sarah Porter" at 12:48 PM: "Sorry, there seems to be a problem with your connection." Below that is a message from "Samantha Ramos" at 12:52 PM: "the notes you made?". A message from "Tiffany Myers" at 12:51 PM follows: "Sarah Porter 12:48 PM No problem! They are a bit messy at the moment. I'm just going to tidy them up a bit. I'll upload them shortly." and "Samantha Ramos 12:52 PM Did anyone take a photo of the drawings we made on the whiteboard? I'd like made a high fidelity version of that concept." At the bottom of the main area is a message input field with placeholder text "Type your message" and a blue send button with icons for @ and 😊.

What you actually need

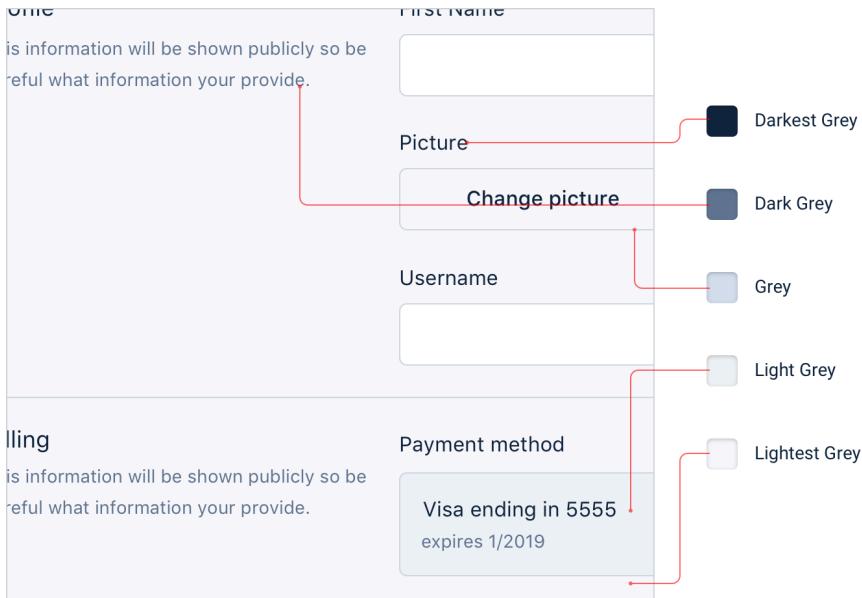
You can't build anything with five hex codes. To build something real, you need a much more comprehensive set of colors to choose from.



You can break a good color palette down into three categories.

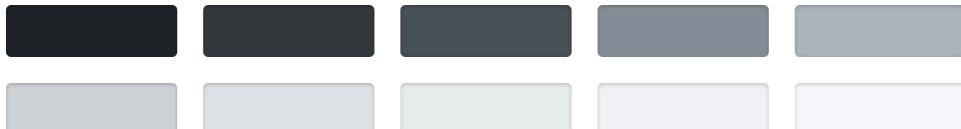
Greys

Text, backgrounds, panels, form controls — almost everything in an interface is grey.



You'll need more greys than you think, too — three or four shades might sound like plenty but it won't be long before you wish you had something a little darker than shade #2 but a little lighter than shade #3.

In practice, you want 8-10 shades to choose from (more on this in "*Define your shades up front*"). Not so many that you waste time deciding between shade #77 and shade #78, but enough to make sure you don't have to compromise too much.

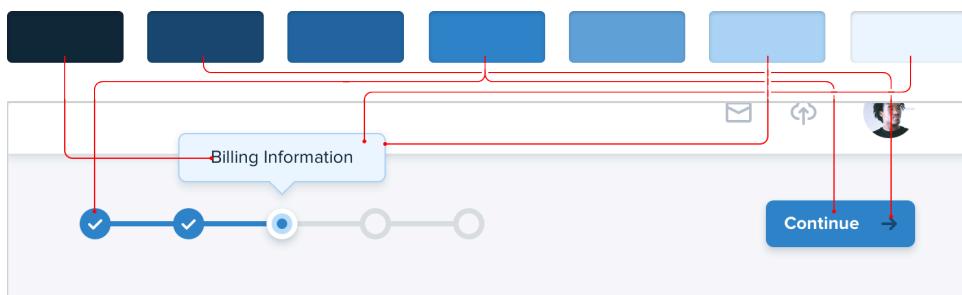


True black tends to look pretty unnatural, so start with a really dark grey and work your way up to white in steady increments.

Primary color(s)

Most sites need one, *maybe* two colors that are used for primary actions, active navigation elements, etc. These are the colors that determine the overall look of a site — the ones that make you think of Facebook as “blue”.

Just like with greys, you need a variety (5-10) of lighter and darker shades to choose from.

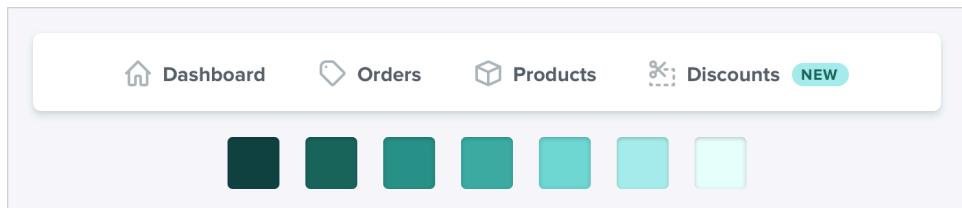


Ultra-light shades can be useful as a tinted background for things like alerts, while darker shades work great for text.

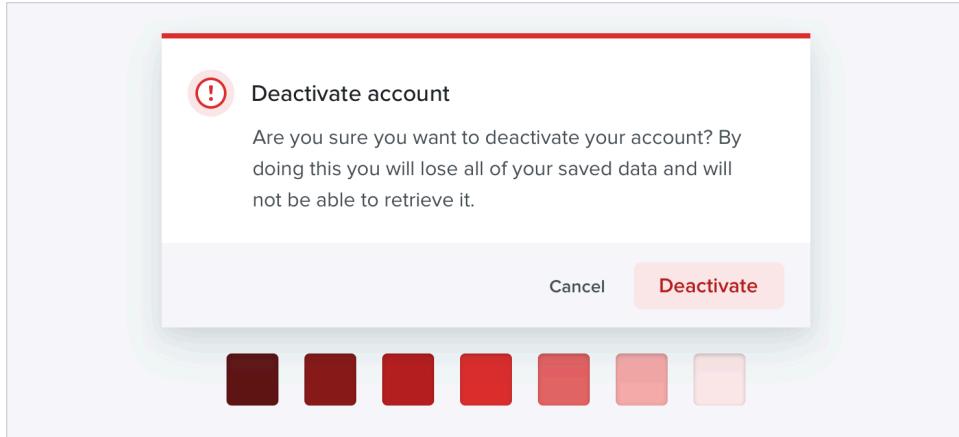
Accent colors

On top of primary colors, every site needs a few accent colors for communicating different things to the user.

For example, you might want to use an eye-grabbing color like yellow, pink, or teal to highlight a new feature:



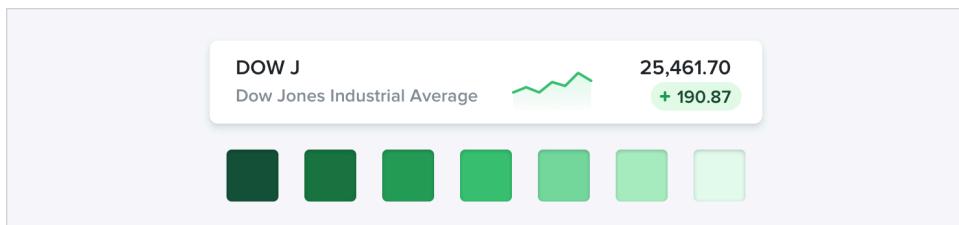
You might also need colors to emphasize different semantic states, like red for confirming a destructive action:



...yellow for a warning message:



...or green to highlight a positive trend:



You'll want multiple shades for these colors too, even though they should be used pretty sparingly throughout the UI.

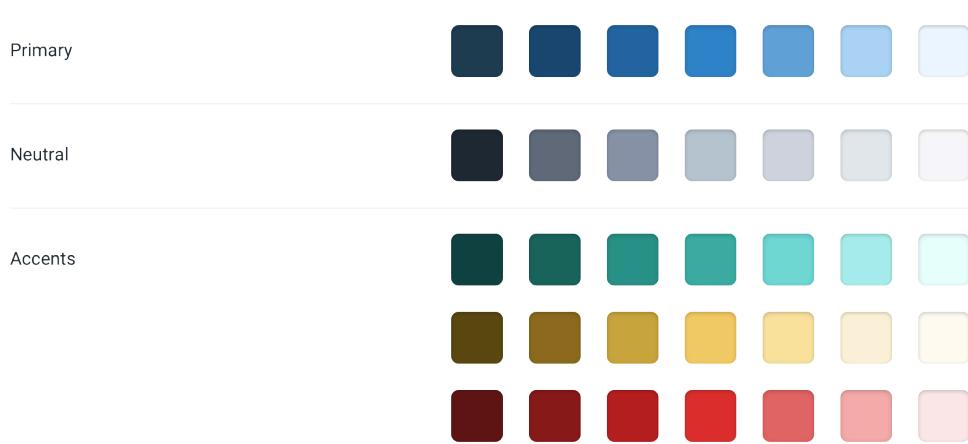
If you're building something where you need to use color to distinguish or categorize similar elements (like lines on graphs, events in a calendar, or tags on a project), you might need even more accent colors.

All in, it's not uncommon to need as many as *ten* different colors with 5-10 shades each for a complex UI.

Define your shades up front

When you need to create a lighter or darker variation of a color in your palette, don't get clever using CSS preprocessor functions like "lighten" or "darken" to create shades on the fly. That's how you end up with 35 *slightly* different blues that all look the same.

Instead, define a fixed set of shades up front that you can choose from as you work.



So how do you put together a palette like this anyways?

Choose the base color first

Start by picking a *base* color for the scale you want to create — the color in the middle that your lighter and darker shades are based on.

There's no real scientific way to do this, but for primary and accent colors, a good rule of thumb is to pick a shade that would work well as a button background.



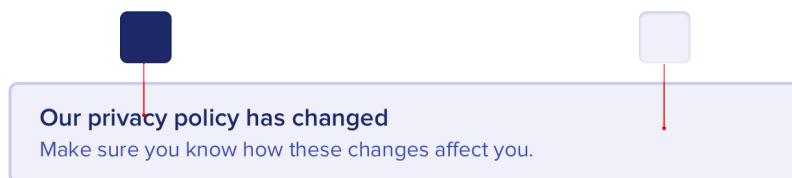
It's important to note that there are no real rules here like "start at 50% lightness" or anything — every color behaves a bit differently, so you'll have to rely on your eyes for this one.

Finding the edges

Next, pick your darkest shade and your lightest shade. There's no real science to this either, but it helps to think about where they will be used and choose them using that context.

The darkest shade of a color is usually reserved for text, while the lightest shade might be used to tint the background of an element.

A simple alert component is a good example that combines both of these use cases, so it can be a great place to pick these colors.



Start with a color that matches the hue of your base color, and adjust the saturation and lightness until you're satisfied.

Filling in the gaps

Once you've got your base, darkest, and lightest shades, you just need to fill in the gaps in between them.

For most projects, you'll need at least 5 shades per color, and probably closer to 10 if you don't want to feel too constrained.

Nine is a great number because it's easy to divide and makes filling in the gaps a little more straightforward. Let's call our darkest shade 900, our base shade 500, and our lightest shade 100.



Start by picking shades 700 and 300, the ones right in the middle of the gaps. You want these shades to feel like the perfect compromise between the shades on either side.



This creates four more holes in the scale (800, 600, 400, and 200), which you can fill using the same approach.



You should end up with a pretty balanced set of colors that provide just enough options to accommodate your design ideas without feeling limiting.

What about greys?

With greys the base color isn't as important, but otherwise the process is the same. Start at the edges and fill in the gaps until you have what you need.



Pick your darkest grey by choosing a color for the darkest text in your project, and your lightest grey by choosing something that works well for a subtle off-white background.

It's not a science

As tempting as it is, you can't rely purely on math to craft the perfect color palette.

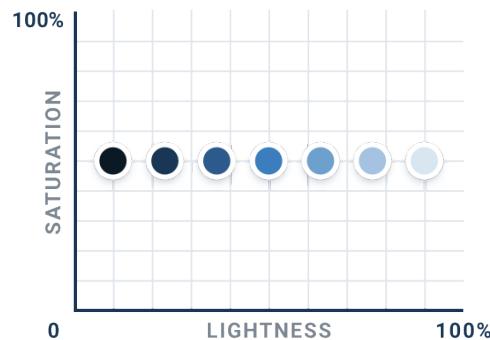
A systematic approach like the one described above is great to get you started, but don't be afraid to make little tweaks if you need to.

Once you actually start using your colors in your designs, it's almost inevitable that you'll want to tweak the saturation on a shade, or make a couple of shades lighter or darker. Trust your eyes, not the numbers.

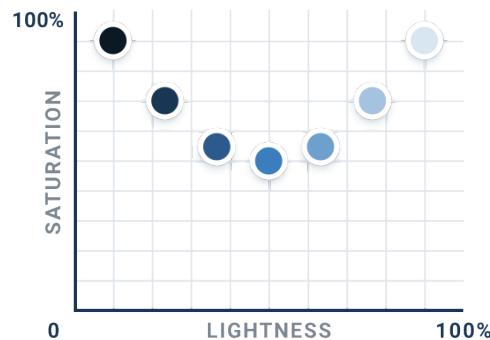
Just try to avoid adding *new* shades too often if you can avoid it. If you're not diligent about limiting your palette, you might as well have no color system at all.

Don't let lightness kill your saturation

In the HSL color space, as a color gets closer to 0% or 100% lightness, the impact of saturation is weakened — the same saturation value at 50% lightness looks more colorful than it does at 90% lightness.



That means that if you don't want the lighter and darker shades of a given color to look washed out, you need to increase the saturation as the lightness gets further away from 50%.



It's subtle but little details like this add up, especially when a color is being applied to a large section of a UI.

But what if your base color is already heavily saturated? How do you increase the saturation if it's already at 100%?

Use perceived brightness to your advantage

Which of these two colors do you think is lighter?



The yellow, right? Well it turns out both colors actually have the exact same "lightness" in terms of HSL:

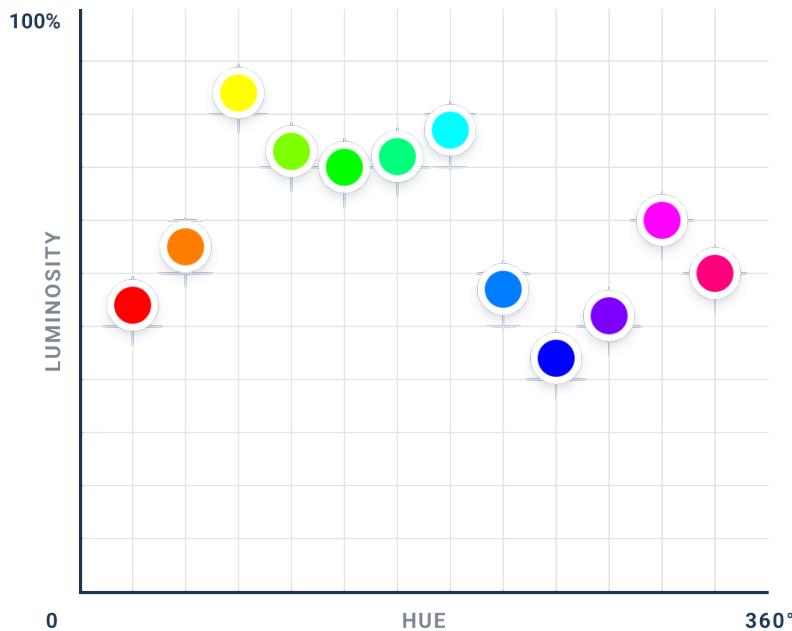


So why do we see the yellow as lighter? Well it turns out that every hue has an inherent perceived brightness due to how the human eye perceives color.

You can calculate the perceived brightness of a color by plugging its RGB components into this formula:

$$\sqrt{0.299 r^2 + 0.587 g^2 + 0.114 b^2}$$

Taking samples of different hues with 100% saturation and 50% lightness, we can get a good sense of the perceived brightness of different colors around the color wheel:



As expected, yellow has a higher perceived brightness than blue. But what's interesting here is that perceived brightness doesn't simply change linearly from the darkest hue to the lightest hue — instead, there are three separate local minimums (red, green, and blue) and three local maximums (yellow, cyan, and magenta).

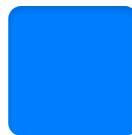
Changing brightness by rotating hue

On the surface, this is certainly an interesting thing to understand about color. But things get really interesting when you realize how you can use this knowledge in your designs.

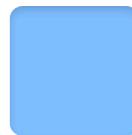
Normally when you want to change how light a color looks, you adjust the *lightness* component:



H 210
S 100%
L 25%



H 210
S 100%
L 50%



H 210
S 100%
L 75%

While this does work to lighten or darken a color, you often lose some of the color's *intensity* — the color also looks closer to white or to black, not just lighter or darker.



Since different hues have a different perceived brightness, another way you can change the brightness of a color is *by rotating its hue*.

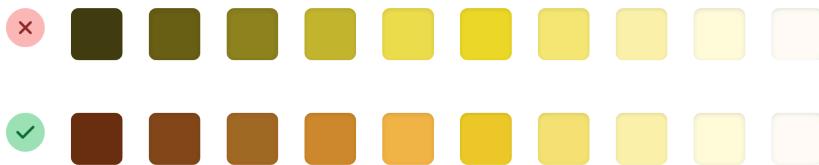
To make a color lighter, rotate the hue towards the nearest bright hue — 60°, 180°, or 300°.



To make a color darker, rotate the hue towards the nearest dark hue — 0°, 120°, or 240°.



This can be really useful when trying to create a palette for a light color like yellow. By gradually rotating the hue towards more of an orange as you decrease the lightness, the darker shades will feel warm and rich instead of dull and brown:



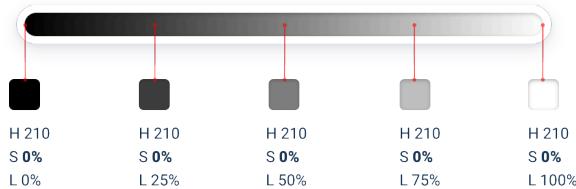
You can of course combine these approaches too, getting some of the brightness by adjusting the hue and some from adjusting the lightness.



While this is a great way to change a color's brightness without affecting its intensity, it works best in small doses. Don't rotate the hue more than 20-30° or it will look like a totally different color instead of just lighter or darker.

Greys don't have to be grey

By definition, true grey has a saturation of 0% — it doesn't have any actual color in it at all.



But in practice, a lot of the colors that we *think* of as grey are actually saturated quite heavily:

Recently Updated

- Carl Phillips
US\$249.00
Due in 5 days
View Invoice >
- Tyler Burns
US\$479.00
Due yesterday
View I

All Invoices

CLIENT	ISSUED DATE
Maria Schmidt	

Red arrows point from the grayscale squares in the app interface to their corresponding HSL color codes:
- Dark blue square: hsl(212, 56%, 16%)
- Medium blue square: hsl(212, 18%, 35%)
- Light gray square: hsl(210, 32%, 93%)
- Very light gray square: hsl(210, 38%, 97%)

This saturation is what makes some greys feel cool and other greys feel warm.

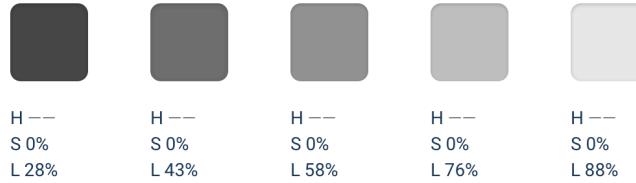
Color temperature

If you've ever purchased light bulbs before, you've had to make the decision between "warm white" bulbs that give off a yellow-ish light, and "cool white" bulbs that give off a blue-ish light.

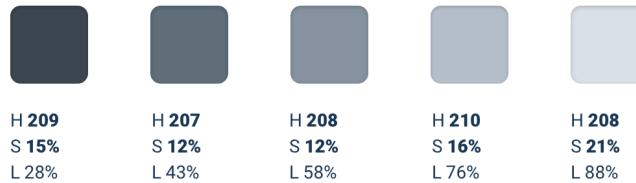
Saturating greys in a user interface works in a similar same way.

If you want your greys to feel cool, saturate them with a bit of blue:

NEUTRAL GREYS

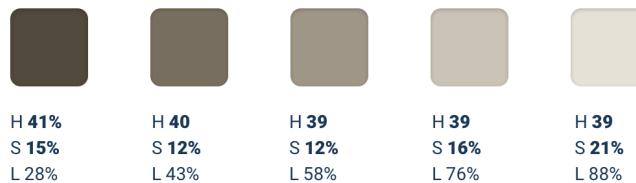


COOL GREYS



To give your greys a warmer feel, saturate them with a bit of yellow or orange:

WARM GREYS



To maintain a consistent temperature, don't forget to increase the saturation for the lighter and darker shades. If you don't, those shades will look a bit washed out compared to the greys that are closer to 50% lightness.

How much you want to saturate your greys is completely up to you — add just a little if you only want to tip the temperature slightly, or crank it up if you want the interface to lean strongly in one direction or the other.

Accessible doesn't have to mean ugly

To make sure your designs are accessible, the Web Content Accessibility Guidelines (WCAG) recommend that normal text (*under ~18px*) has a contrast ratio of at least 4.5:1, and that larger text has a contrast ratio of at least 3:1.

Normal Text

EXAMPLE	COLOR	CONTRAST	GRADE
The five boxing wizards jump quickly.	hsl(0, 0%, 54%)	3.45:1	Fail
The five boxing wizards jump quickly.	hsl(0, 0%, 42%)	5.41:1	AA
The five boxing wizards jump quickly.	hsl(0, 0%, 33%)	7.57:1	AAA

Large Text

EXAMPLE	COLOR	CONTRAST	GRADE
The five boxing wizards jump...	hsl(0, 0%, 59%)	2.96:1	Fail
The five boxing wizards jump...	hsl(0, 0%, 54%)	3.45:1	AA
The five boxing wizards jump...	hsl(0, 0%, 42%)	5.41:1	AAA

For typical *dark-text-on-a-light-background* situations, meeting this recommendation is pretty easy, but it gets a lot trickier when you start working with color.

Flipping the contrast

When using white text on a colored background, you'd be surprised how dark the color often needs to be to meet that 4.5:1 contrast ratio.



NAME	STATUS	POLICY	
Molly Sanders VP of Sales	Approved	\$20,000 All-inclus	2.25 (Fail)
Michael Roberts Advisory Board	Awaiting Approval	\$5,000 Basic Pol	1.56 (Fail)
Devin Childs Marketing Manager	Declined	\$5,000 Basic Pol	3.14 (Fail)

This can create hierarchy issues when those elements aren't supposed to be the focus of the page — dark colored backgrounds will really grab the user's attention.



NAME	STATUS	POLICY	
Molly Sanders VP of Sales	Approved	\$20,000 All-inclus	5.97 (AA)
Michael Roberts Advisory Board	Awaiting Approval	\$5,000 Basic Pol	5.07 (AA)
Devin Childs Marketing Manager	Declined	\$5,000 Basic Pol	5.34 (AA)

You can solve this problem by *flipping the contrast*. Instead of using light text on a dark colored background, use dark colored text on a light colored background:

	NAME	STATUS	POLICY
	Molly Sanders VP of Sales	Approved	\$20,000 All-inclus
	Michael Roberts Advisory Board	Awaiting Approval	\$5,000 Basic Pol
	Devin Childs Marketing Manager	Declined	\$5,000 Basic Pol

The color is still there to help support the text, but it's way less in-your-face and doesn't interfere as much with other actions on the page.

Rotating the hue

Even harder than white text on a colored background is *colored text on a colored background*. You'll run into this situation if you're ever trying to pick a color for some secondary text inside a dark-colored panel.

If you start by taking the background color and simply adjusting the lightness and saturation, you'll find that it's hard to meet the recommended contrast ratio without getting very close to pure white.



Here every step of the way

Think of us as your very own help desk, with free, around-the-clock support. Contact us anytime – 24 hours a day, 7 days a week.

REACH OUT NOW

Background
hsl(240, 34%, 34%)

Text
hsl(240, 44%, 89%)

8.37:1 **AAA**

You don't want the primary text and the secondary text to look the same, so what else can you do?

Well since some colors are brighter than others, one way to increase the contrast without getting closer to white is to *rotate the hue* towards a brighter color, like cyan, magenta, or yellow.



Here every step of the way

Think of us as your very own help desk, with free, around-the-clock support. Contact us anytime – 24 hours a day, 7 days a week.

REACH OUT NOW

Background
hsl(240, 34%, 34%)

Text
hsl(188, 100%, 85%)

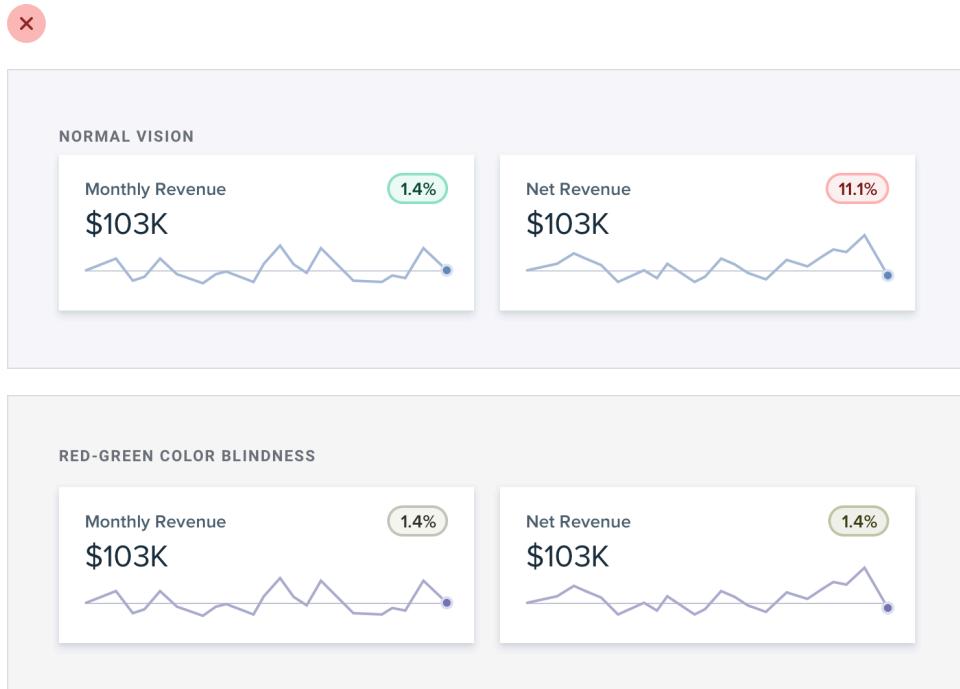
8.71:1 **AAA**

This can make it a lot easier to make the text accessible while still keeping it colorful.

Don't rely on color alone

Color can be a fantastic way to enhance information and make it easier to understand, but be careful not to *rely* on it, or users with color blindness will have a hard time interpreting your UI.

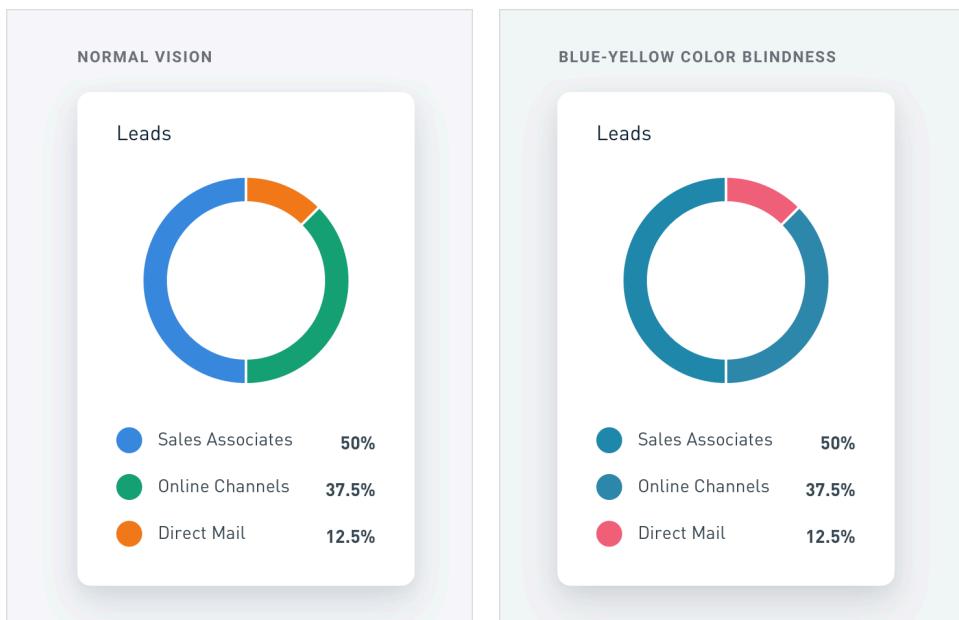
Take these metric cards for example. With this design, someone who is red-green colorblind can't easily tell if a metric has gotten better or worse:



An easy fix for this is to also communicate that information in some other way, like by adding icons to indicate if the change is positive or negative.

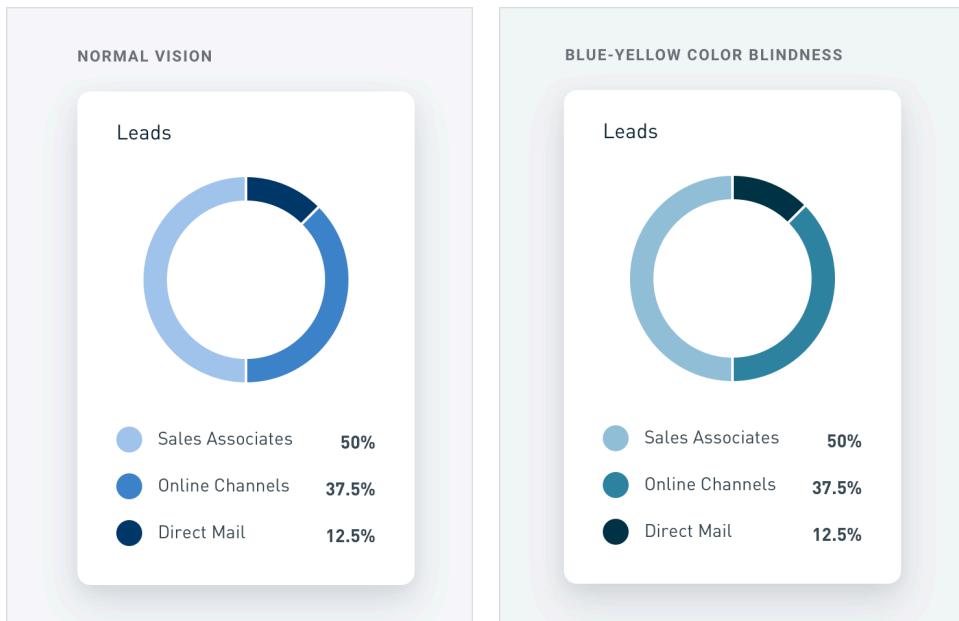


What about something like a graph, where each trend line has a different color?



In situations like this, try relying on contrast instead of using completely different colors. It's much easier for someone who's colorblind to tell the

difference between light and dark than it is for them to tell the difference between two distinct colors.



Always use color to support something that your design is already saying; never use it as the only means of communication.

Creating Depth

Emulate a light source

Have you ever noticed how some elements in an interface feel like they're raised off of the page, while others feel like they are inset into the background?

The image shows a user interface for a real estate search. On the left, there's a dark-themed sidebar with dropdown menus for 'Transaction Type' (set to 'For Sale') and 'Property Type' (set to 'Residential'). Below these are a price range slider and a list of features: 'Balcony', 'Central Air', and 'Pool'. A red bracket on the left side of the image groups the 'Property Type' dropdown, the price range slider, and the feature list, indicating they appear 'inset' from the main content area. On the right, there's a listing card for a 'DETACHED HOUSE • 5Y OLD' located at '528 Douglas Street' with a price of '\$680,000'. This card has a white background and a thin shadow, making it look 'raised' above the main interface. A red bracket on the right side of the image groups the listing card and the second image of a house, indicating they appear 'raised'.

Appears inset

Appears raised

Creating this effect might look complicated at first, but it actually only requires you to understand one fundamental rule.

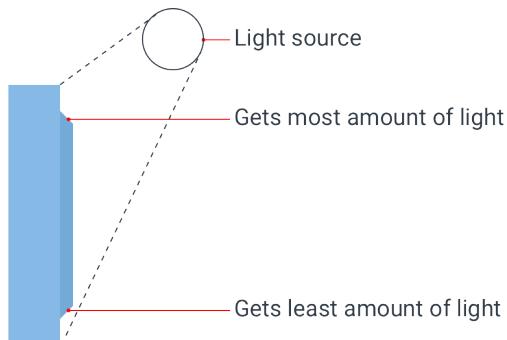
Light comes from above

Take a look at the panelling on this door:



Even though you're just looking at a flat image, it's still pretty obvious that the panels on the door are raised. Why is that?

Notice how the top edge of the panel is lighter? That's because it's angled towards the sky and receives more light. Similarly, the bottom edge is darker because it's angled away from the sky, receiving *less* light.

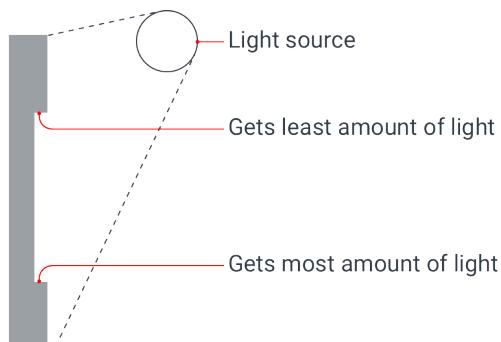


The only way those edges could possibly be oriented that way is if the panel itself is raised, so that's how our brains perceive it.

Now take a look at the panelling on this cabinet:



In this case it's clear that the panels are *inset* because there's a shadow at the top indicating that the lip above is blocking the light, and the bottom edge is lighter, indicating that it's angled upward.



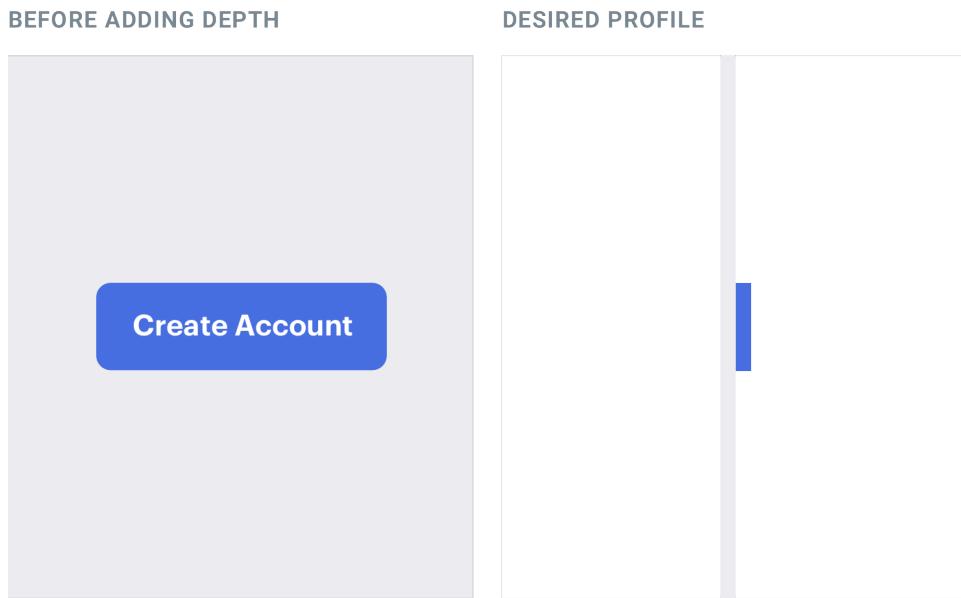
To create this same sense of depth in your designs, all you need to do is mimic the way light affects things in the real world.

Simulating light in a user interface

If you want an element to appear raised or inset, first figure out what *profile* you want that element to have, then mimic how a light source would interact with that shape.

Raised elements

For example, say you had a button and you wanted it to feel raised off of the page, with perfectly flat edges on the top and bottom:

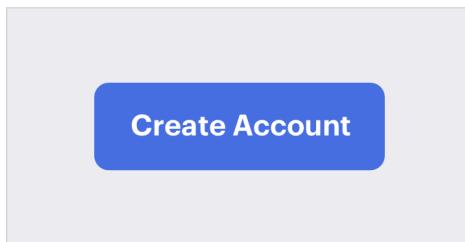


Because the top and bottom edges are both flat, it would be impossible to see both of them at the same time. People generally look slightly downward towards their screens, so for the most natural look, reveal a little bit of the top edge and hide the bottom edge.

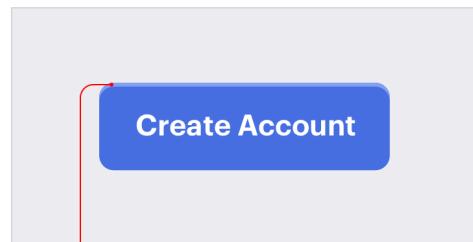
Since the top edge is facing upward, make it slightly lighter than the face of

the button, usually using a top border or an inset box shadow with a slight vertical offset:

BEFORE



AFTER



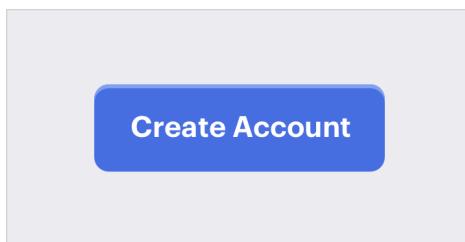
```
box-shadow: inset 0 1px 0 hsl(224, 84%, 74%);
```

Choose the lighter color by hand instead of using a semi-transparent white for best results — simply overlaying white can suck the saturation out of the underlying color.

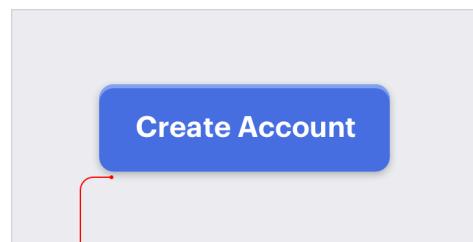
Next, you need to account for the fact that a raised element will block some of the light from reaching the area below the element.

Do this by adding a small dark box shadow with a slight vertical offset (you *only want the shadow to appear below the element*):

BEFORE



AFTER



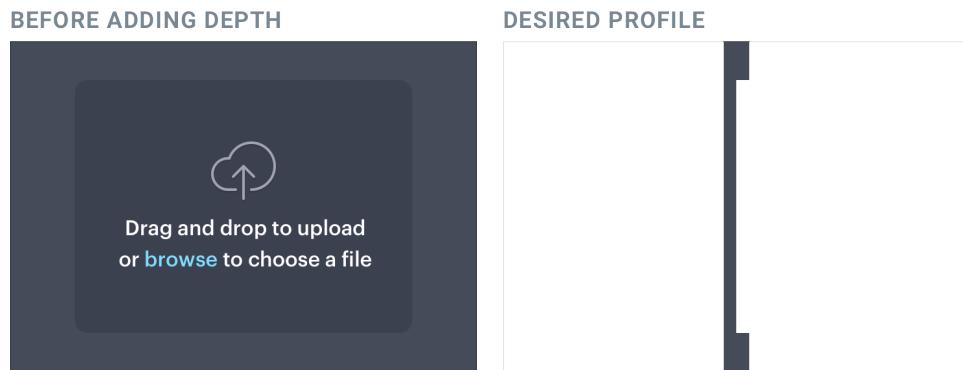
```
box-shadow: 0 1px 3px hsla(0, 0%, 0%, .2);
```

Don't get carried away with the blur radius, a couple of pixels is plenty. These

sorts of shadows should have pretty sharp edges — take a look at the shadow cast by the bottom of a wall outlet or window frame for a real-world example.

Inset elements

Say you're designing a "well" component that should feel like it's recessed into the page.



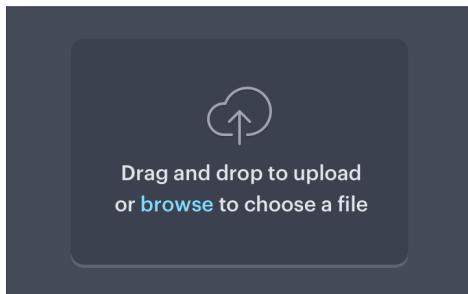
Looking slightly downward, only the bottom lip would be visible. Since it's facing towards the sky, give that edge a slightly lighter color using a bottom border or inset shadow with a negative vertical offset:



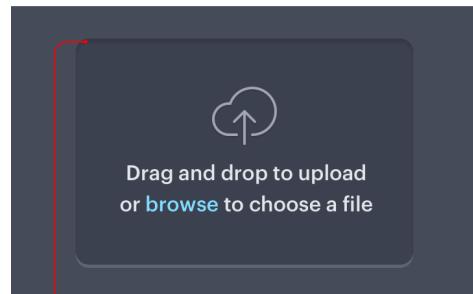
```
box-shadow: 0 -2px 0 hsla(0, 0%, 100%, .15);
```

The area above the well should block some of the light from reaching the very top of the well, so add a small dark inset box shadow with a slight positive vertical offset to make sure it doesn't poke through at the bottom:

BEFORE



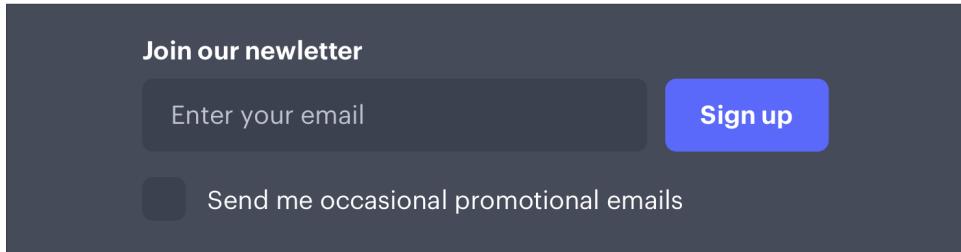
AFTER



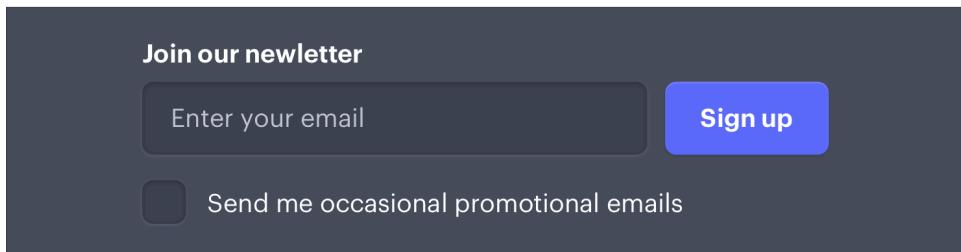
```
box-shadow: inset 0 2px 2px hsla(0, 0%, 0%, 0.1);
```

This same treatment works for any element that may need to appear inset, for example text inputs and checkboxes:

BEFORE



AFTER



Don't get carried away

Once you understand how to simulate light in an interface, it can be tempting to tinker away for hours, tweaking and tweaking to see how closely you can mimic the real world.

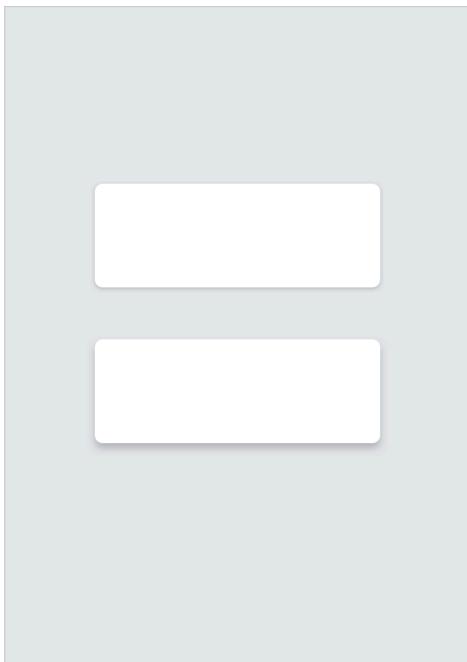
While this can be a fun exercise, in practice it can lead to interfaces that are busy and unclear. Borrowing some visual cues from the real world is a great way to add a bit of depth, but there's no need to try and make things look photo-realistic.

Use shadows to convey elevation

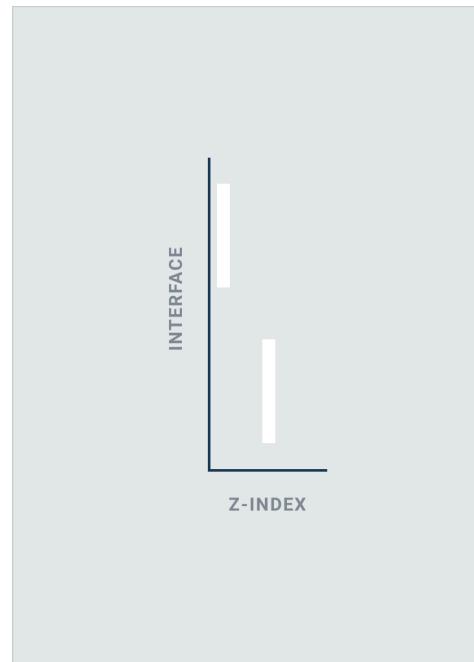
Shadows can be more than just a flashy effect — used thoughtfully, they let you position elements on a virtual z-axis to create a meaningful sense of depth.

Small shadows with a tight blur radius make an element feel only slightly raised off of the background, while larger shadows with a higher blur radius make an element feel much closer to the user:

FRONT VIEW

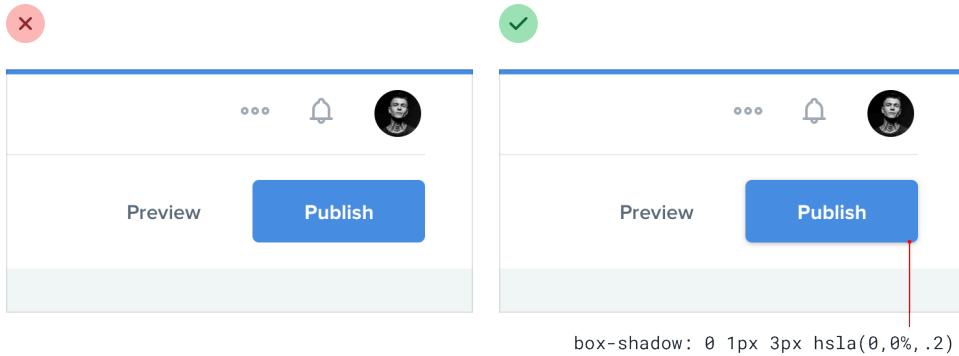


PROFILE VIEW

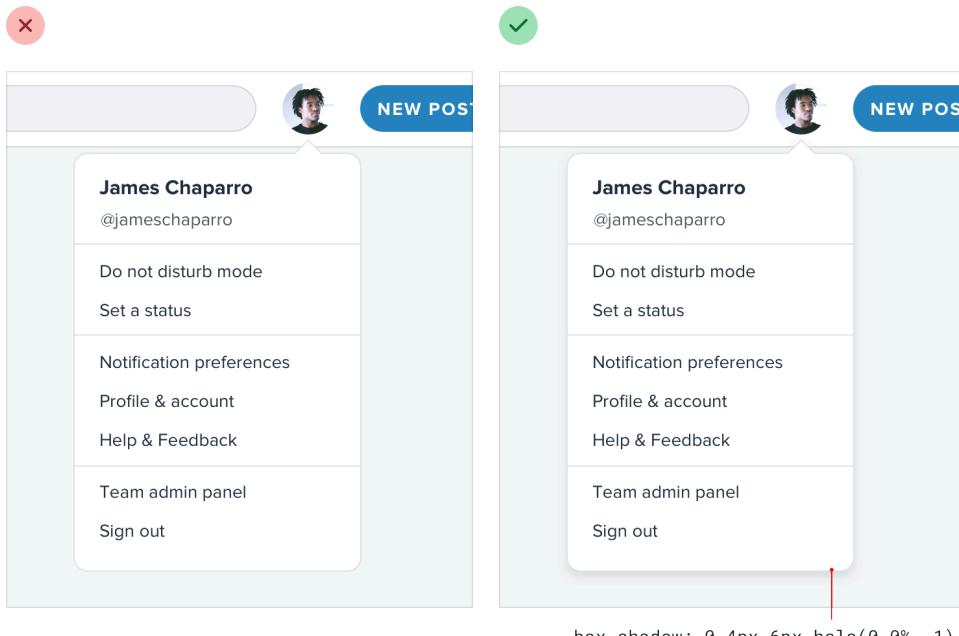


The closer something feels to the user, the more it will attract their focus.

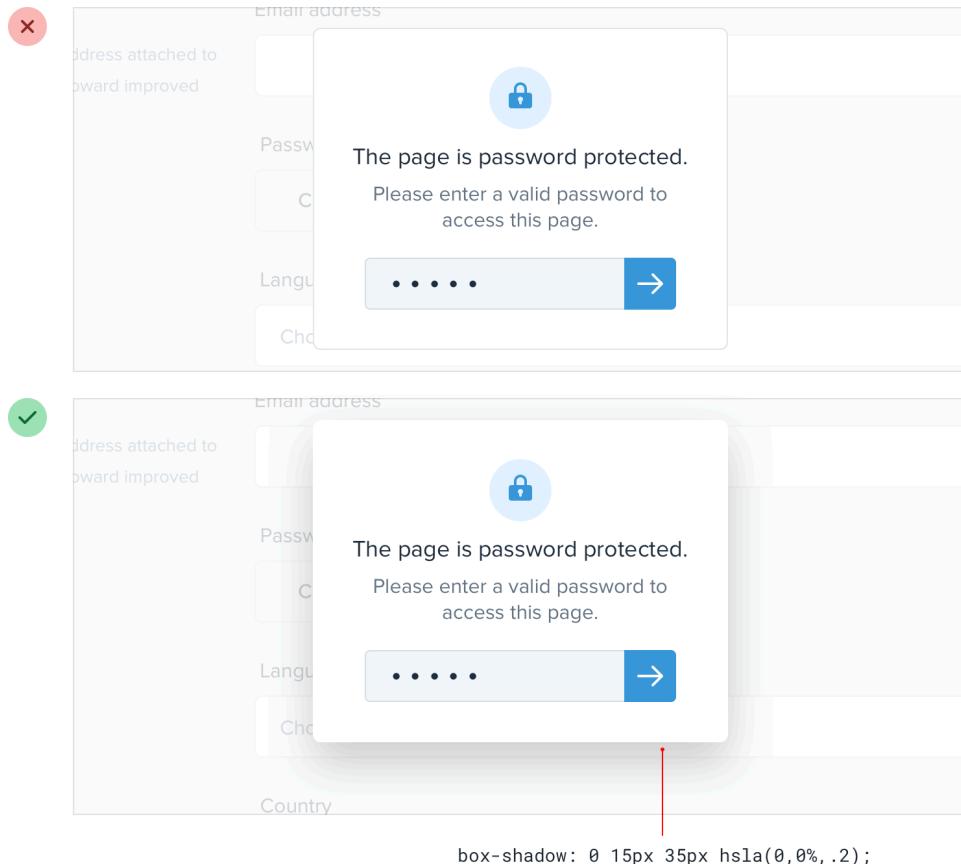
You might use a smaller shadow for something like a button, where you want the user to notice it but don't want it to dominate the page:



Medium shadows are useful for things like dropdowns; elements that need to sit a bit further above the rest of the UI:



Large shadows are great for modal dialogs, where you really want to capture the user's attention:

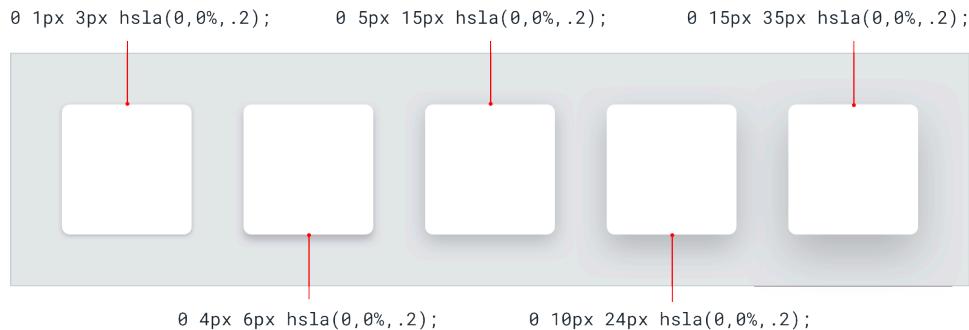


Establishing an elevation system

Just like with color, typography, spacing, and sizing, defining a fixed set of shadows will speed up your workflow and help maintain consistency in your designs.

You don't need a ton of different shadows — five options is usually plenty.

Start by defining your smallest shadow and your largest shadow, then fill in the middle with shadows that increase in size pretty linearly:



Combining shadows with interaction

Shadows aren't only useful for positioning elements on the z-axis statically; they're a great way to provide visual cues to the user as they interact with elements, too.

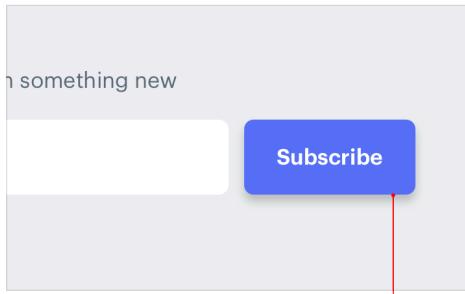
For example, say you had a list of items where the user could click and drag each item to sort them. Adding a shadow to an item when a user clicks it makes it feel like it pops forward above the other items in the list, and makes it clear to the user that they can drag it:

A screenshot of a mobile application interface showing a list of songs. At the top is a search bar labeled "Filter". Below it is a table with three columns: "TITLE", "ARTIST", and "ADDED". The table has five rows of data. The second row, which contains the song "Kerala" by Bonobo added "2 min ago", has a blue selection bar underneath it, and the text is slightly darker, indicating it is selected. The other four rows are standard table rows.

Filter		
TITLE	ARTIST	ADDED
+ Kerala	Bonobo	2 min ago
: + Blush	Leon Vynehall	4 days ago
+ Bad Kingdom	Moderat	6 days ago
+ Fewer Locks	Affelaye	6 days ago

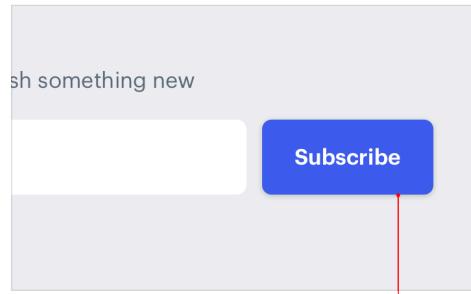
Similarly, you can make a button feel like it's being pressed into the page when a user clicks it by switching to a smaller shadow, or perhaps removing the shadow altogether:

NORMAL



`box-shadow: 0 4px 6px hsla(0,0%,.2);`

ON CLICK

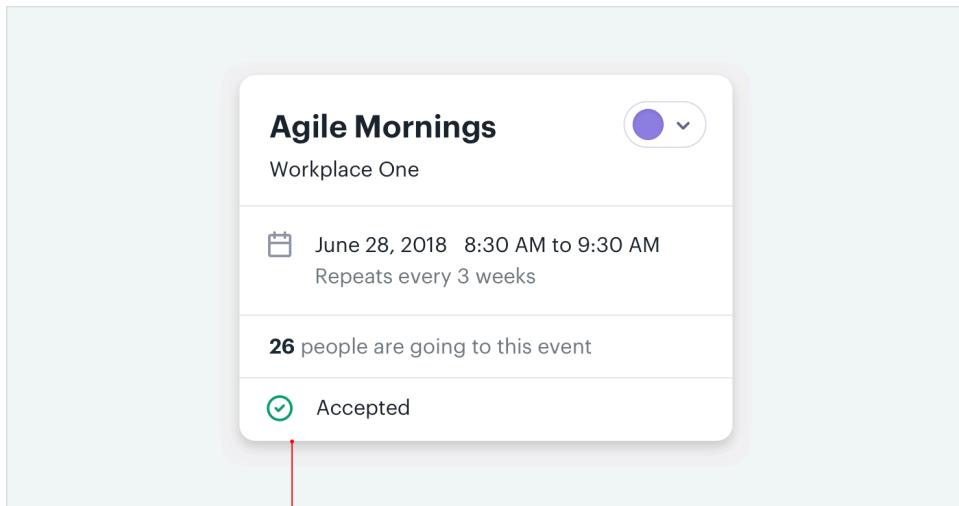


`box-shadow: 0 1px 3px hsla(0,0%,.2);`

Using shadows in a meaningful way like this is a great way to hack the process of choosing what sort of shadow an element should have. Don't think about the shadow itself, think about where you want the element to sit on the z-axis and assign it a shadow accordingly.

Shadows can have two parts

Ever inspected a really nice shadow on a site and noticed they were actually using two shadows?

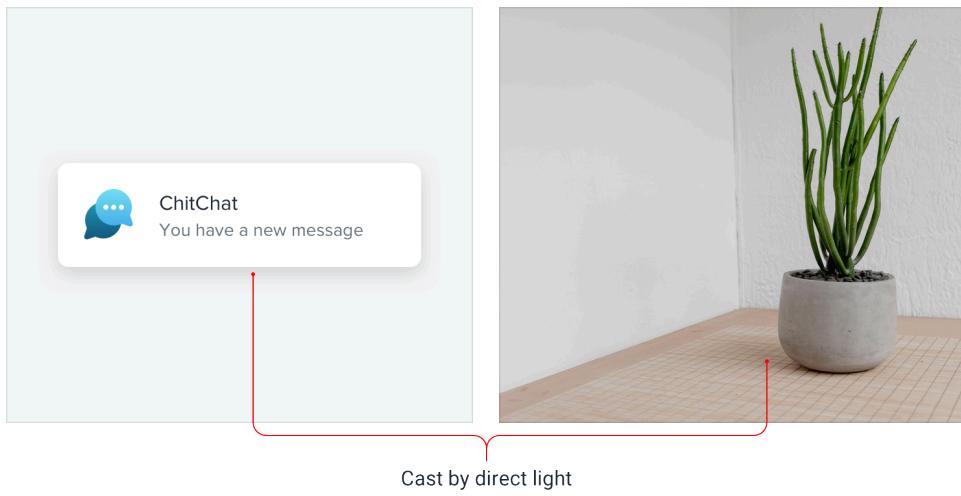


```
box-shadow: 0 4px 6px hsla(0,0%,.7), 0 5px 15px hsla(0,0%,.1);
```

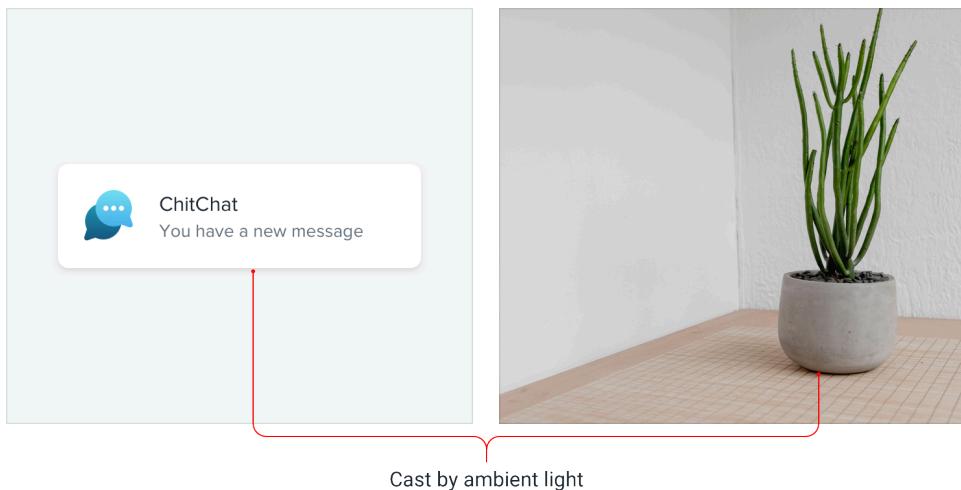
There's a method to this madness, and it's actually pretty simple and makes a lot of sense.

When you see someone combining two shadows, they're not just experimenting randomly until things look nice, they're using each shadow to do a specific job.

The first shadow is larger and softer, with a considerable vertical offset and large blur radius. It simulates the shadow cast behind an object by a direct light source.

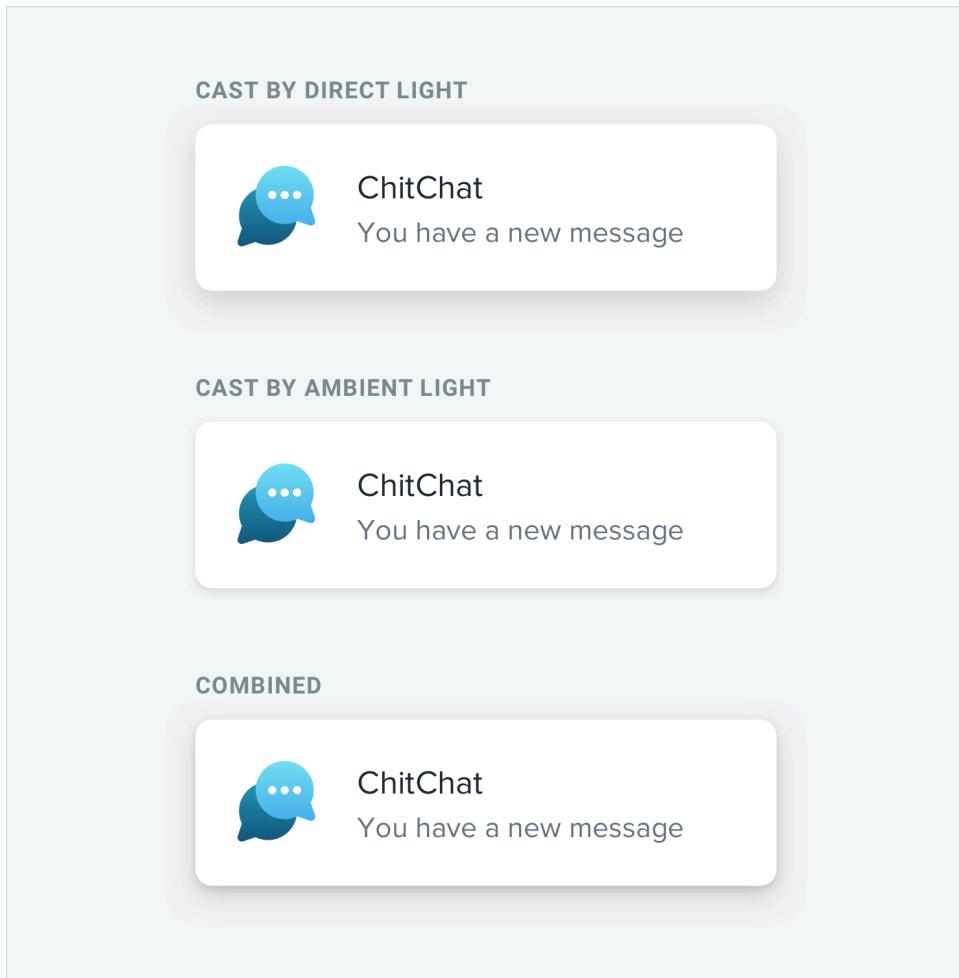


The second shadow is tighter and darker, with less of a vertical offset and a smaller blur radius. It simulates the shadowed area *underneath* an object where even ambient light has a hard time reaching.



Using two shadows like this gives you a lot more control than you'd get with a single shadow — you can keep the larger shadow nice and subtle while still

making the shadow closer the element's edges nice and defined.

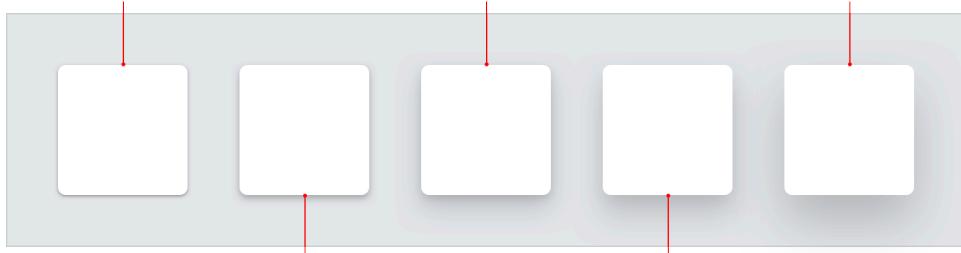


Accounting for elevation

As an object gets further away from a surface, the small, dark shadow created by a lack of ambient light slowly disappears (*go ahead, try it out with something on your desk*).

So if you're going to use this two-shadow technique in your own projects, make sure you make that shadow more subtle for shadows that represent a higher elevation.

```
0 1px 3px hsla(0,0%, .12);      0 10px 20px hsla(0,0%, .15);  
0 1px 2px hsla(0,0%, .24);      0 3px 6px hsla(0,0%, .10);      0 20px 40px hsla(0,0%, .2);
```



```
0 3px 6px hsla(0,0%, .15);      0 15px 25px hsla(0,0%, .15);  
0 2px 4px hsla(0,0%, .12);      0 5px 10px hsla(0,0%, .5);
```

It should be quite distinct for your lowest elevation, and almost (or completely) invisible at your highest elevation.

Even flat designs can have depth

When most people talk about “flat design”, they mean designing without shadows, gradients, or any other effects that try to mimic how light interacts with things in the real-world.

The image shows a mobile-style social media application interface. At the top right is a search bar with a magnifying glass icon and the word "Search". Next to it is a user profile icon of a woman and a blue "New Post" button with a pencil icon. The main content area displays a feed of posts:

- A post by Spencer Mendez (@spencermendez) from 8h ago. The post text is partially visible: "You been up to?". Below the text are icons for video and images. At the bottom of the post are interaction counts (17 likes, 1 share) and a "Share" button.
- A second post by Spencer Mendez (@spencermendez) from 9h ago. The post text is partially visible: "Spent 27th for our 1st Dribbble meetup in Waterloo! Calling all...". Below the text are interaction counts (17 likes, 1 share) and a "Share" button.
- A third post by Spencer Mendez (@spencermendez) from 9h ago. The post text is partially visible: "days at KiteTail HQ learning from the best design...". Below the text are interaction counts (17 likes, 1 share) and a "Share" button.

To the right of the feed is a sidebar:

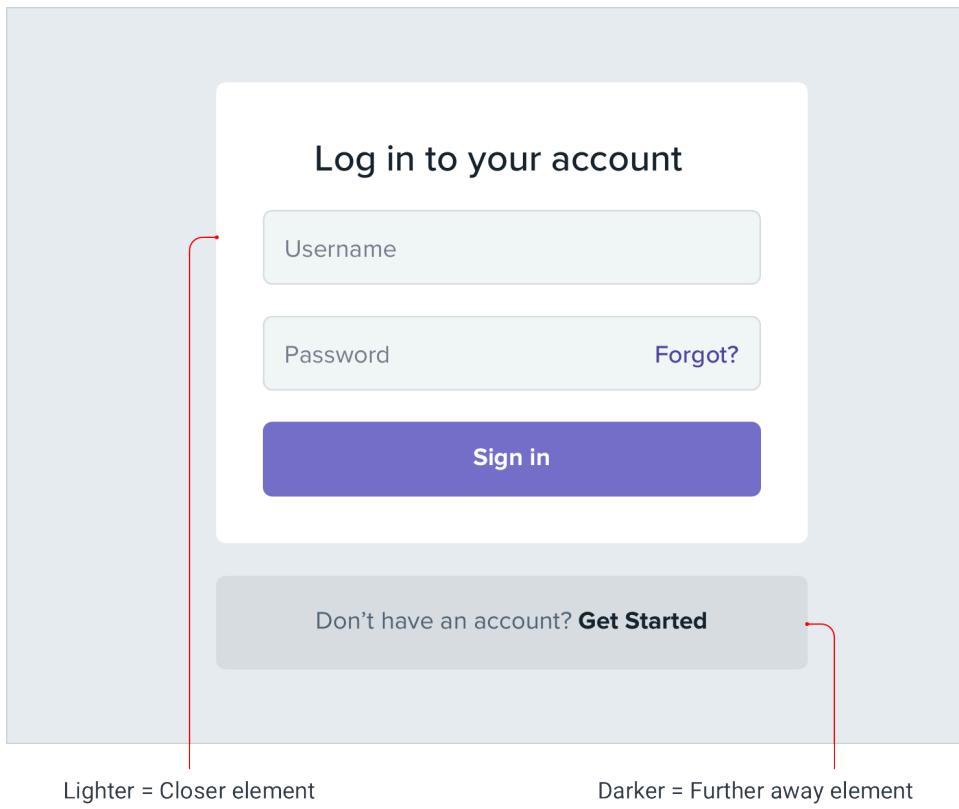
- Who to follow**: A section titled "Who to follow" with three suggestions: Spencer Mendez (@spencermendez), Clark Castillo (@ccastillo), and Carol Barnett (@carol_barnett). Each suggestion includes a small profile picture and a plus sign icon to follow.
- Trending News**: A section titled "Trending News" with three items: "#FirstSnow" (13.7K posts), "Thanksgiving Weekend" (17.8K posts), "#OscarBuzz" (10.3K posts), and "Black Friday" (10.2K posts). Each item includes a small thumbnail image and a plus sign icon.

But the most effective flat designs still convey depth, they just do it in a different way.

Creating depth with color

In general (*especially with shades of the same color*), lighter objects feel closer to us and darker objects feel further away.

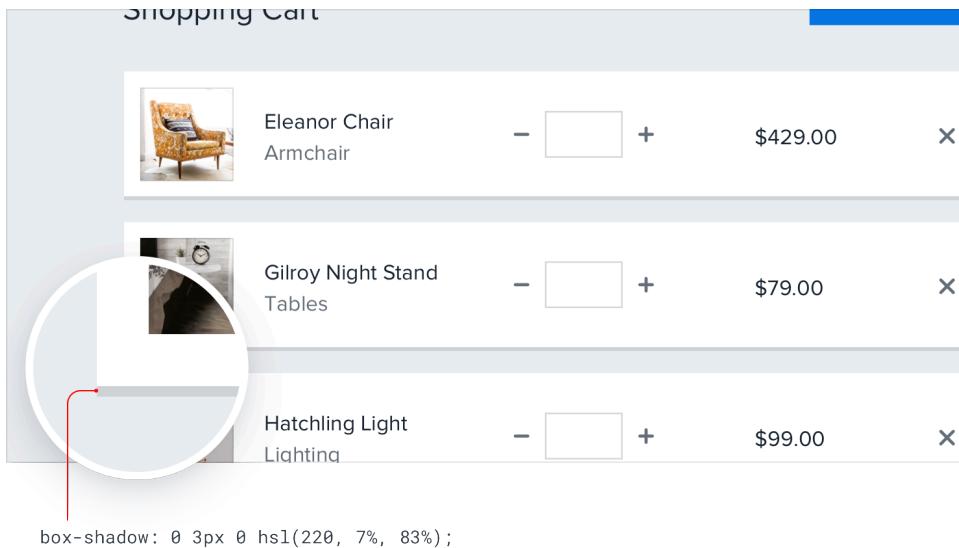
Make an element lighter than the background color to make it feel like it's raised off of the page, or darker than the background color if you want it to feel inset like a well:



This is just as applicable to non-flat designs, too — color is just another tool in your toolbelt for conveying distance.

Using solid shadows

Another way to communicate depth in a flat design is to use short, vertically offset shadows with no blur radius at all.

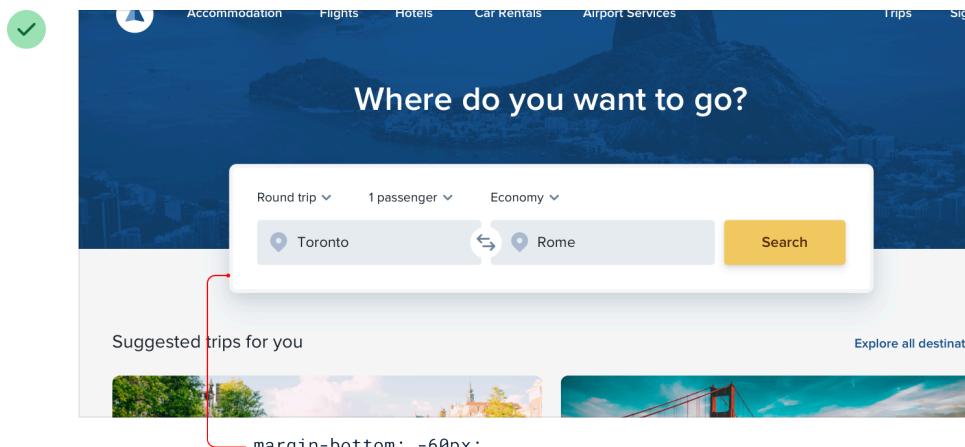
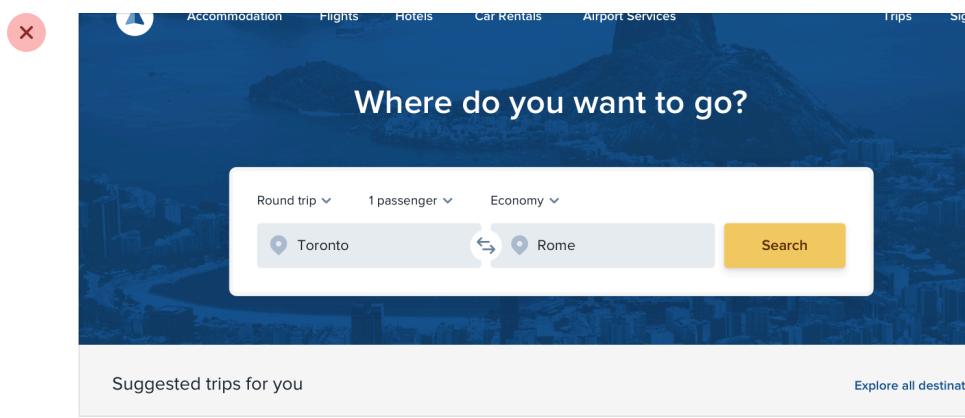


It's a great way to make a card or button stand off the page a little bit without sacrificing that flat aesthetic.

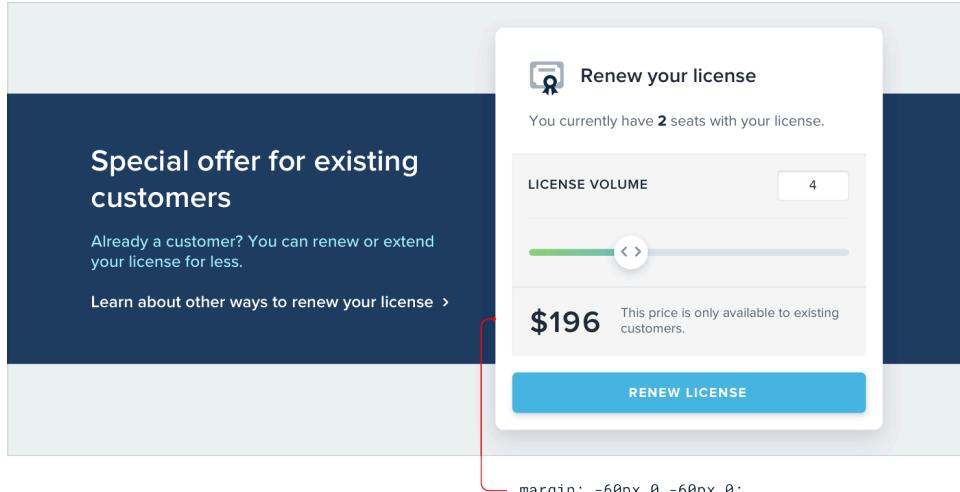
Overlap elements to create layers

One of the most effective ways to create depth is to overlap different elements to make it feel like a design has multiple *layers*.

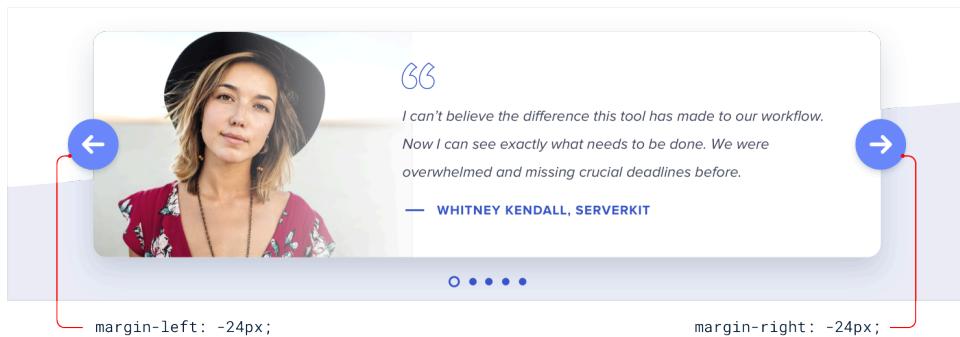
For example, instead of containing a card entirely within another element, offset it so it crosses the transition between two different backgrounds:



You could also make an element taller than its parent, so it overlaps on both sides:



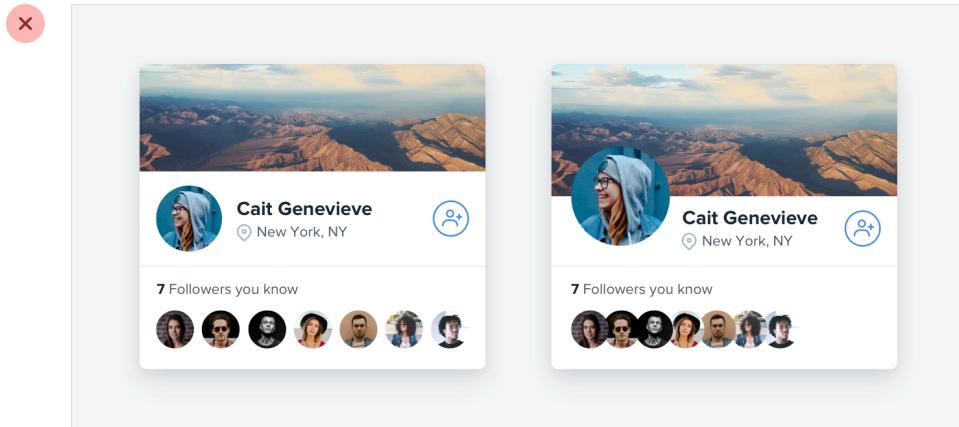
Overlapping elements can add depth to smaller components too, for example the controls on this carousel:



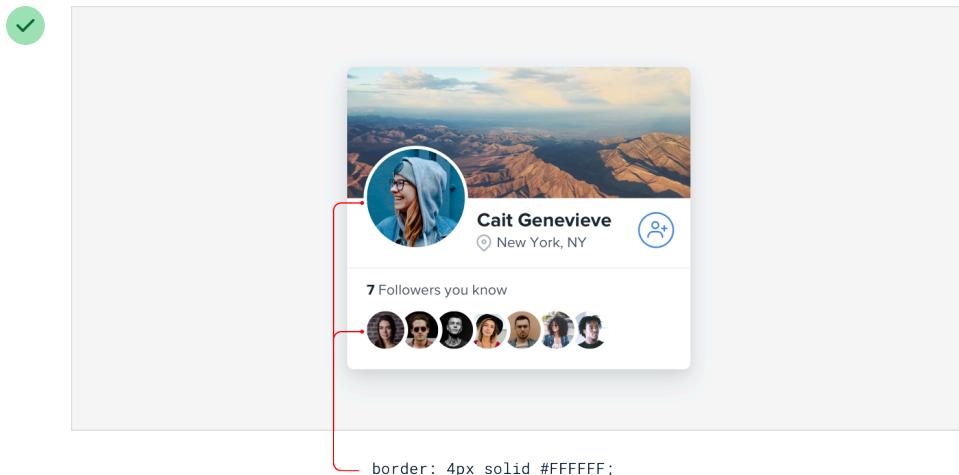
Overlapping images

This technique can work great with images as well, but without special

consideration it's easy for overlapping images to clash.



A simple trick for avoiding this is to give the images an “invisible border” — one that matches the background color — so there’s always a bit of a gap between images:

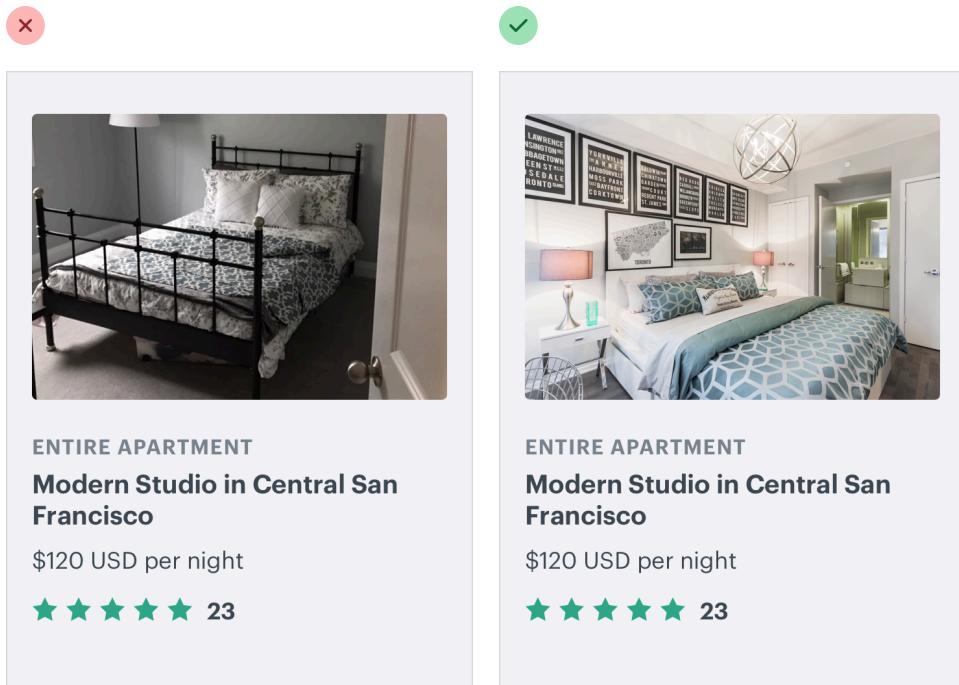


You'll still create the appearance of layers but with none of the ugly clashing.

Working with Images

Use good photos

Bad photos will ruin a design, even if everything else about it looks great.



If your design needs photography and you're not a talented photographer, you've got two options:

1. **Hire a professional photographer.**

If you need very specific photos for your project, entrust a professional. Taking great photos isn't just about using an expensive camera, it's about lighting, composition, color — skills that take years to develop.

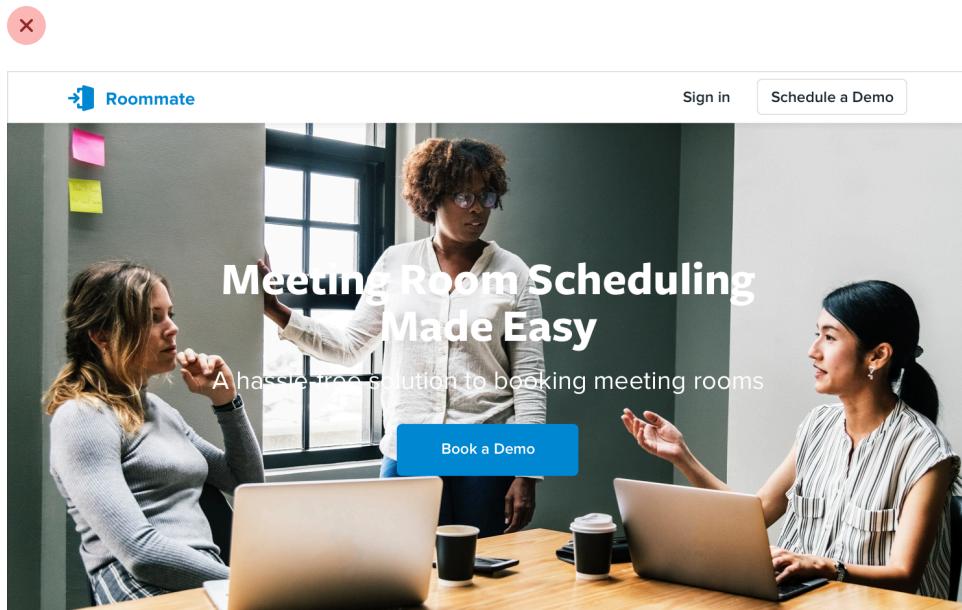
2. Use high quality stock photography.

If your needs are more generic, there are tons of great resources out there where you can purchase great stock photos. There are even sites like Unsplash that offer beautiful photography for free.

Whatever you do, don't design using placeholder images and expect to be able to take some photos with your smartphone and swap them in later. It never works.

Text needs consistent contrast

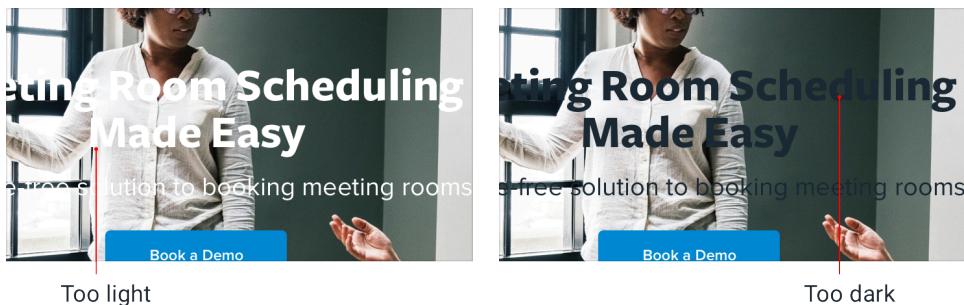
Ever tried to slap a headline on a big hero image, only to find that no matter what color you tried for the text, it was still hard to read?



That's because the problem isn't the text, it's the image.

The problem with background images

Photos can be very dynamic, with a lot of really light areas, and a lot of really dark areas. White text might look great in the dark areas, but it gets lost in the light areas. Dark text looks great in the light areas, but gets lost in the dark areas.



To solve this problem, you need to *reduce* the dynamics in the image to make the contrast between the text and the background more consistent.

Add an overlay

One way to increase the overall text contrast is to add a semi-transparent overlay to the background image.

The image shows a landing page with a semi-transparent black overlay. The overlay has a transparency value of hsla(0, 0%, 0%, .55). It covers the entire background image of three women in an office. Overlaid on the image is the text "Meeting Room Scheduling Made Easy" and "A hassle-free solution to booking meeting rooms". A blue button at the bottom is labeled "Book a Demo". A green circular icon with a checkmark is located in the top-left corner of the page.

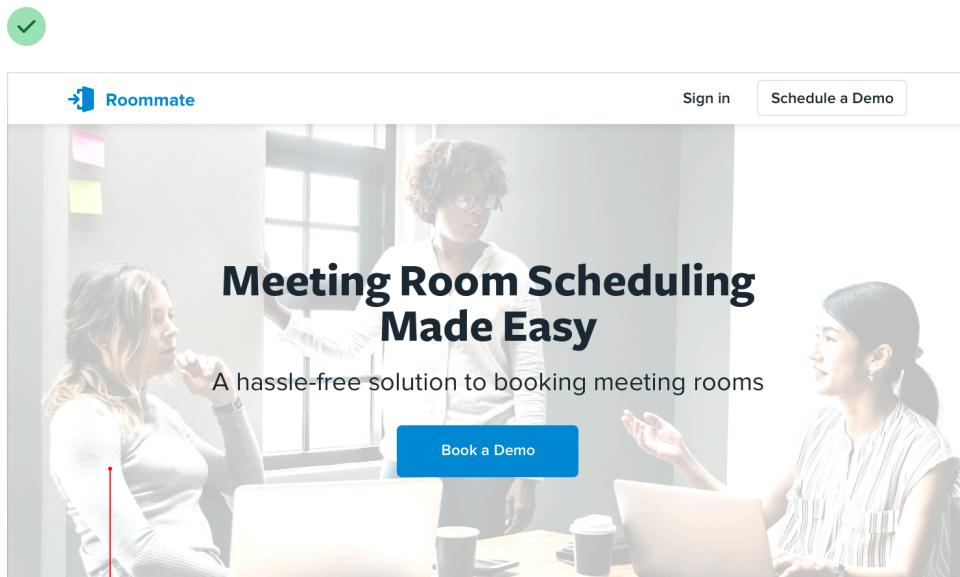
background-color: hsla(0, 0%, 0%, .55);

A black overlay will tone down the light areas and help light text stand out, while a white overlay will brighten up the dark areas and help dark text stand out.

Lower the image contrast

One of the compromises you make when using an overlay is that you're lightening or darkening the *whole* image, not just the problem areas.

If you want more control, another solution is to lower the contrast of the image itself:



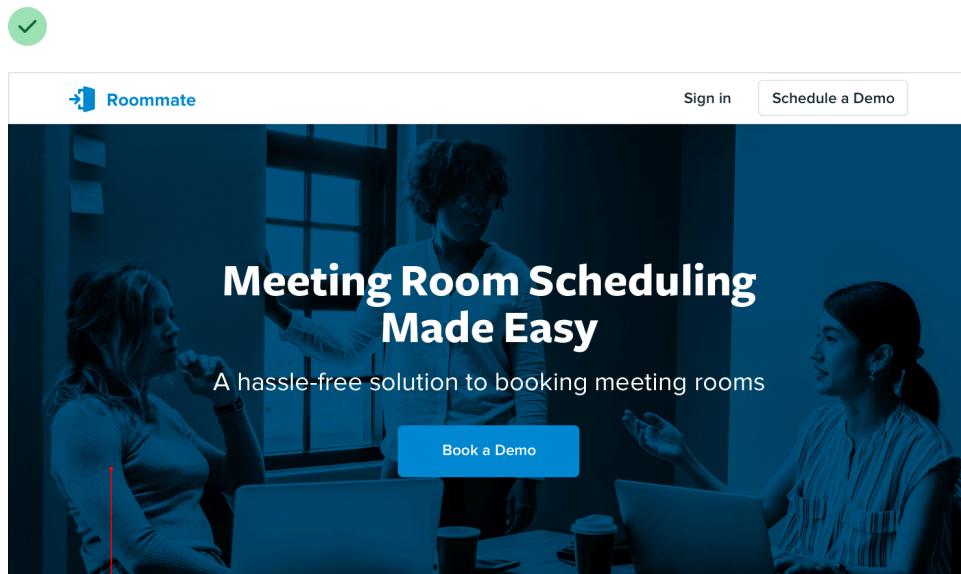
Lowering the contrast will change how light or dark the image feels overall, so make sure to adjust the brightness to compensate.

Colorize the image

Another way to help text stand out against an image is to colorize the image with a single color.

Some photo editing software includes this as a first-class feature, but if yours doesn't, you can create this effect in three steps:

1. **Lower the image contrast**, to balance things out a bit.
2. **Desaturate the image**, to remove any existing color.
3. **Add a solid fill**, using the “multiply” blend mode.



Color: #035581

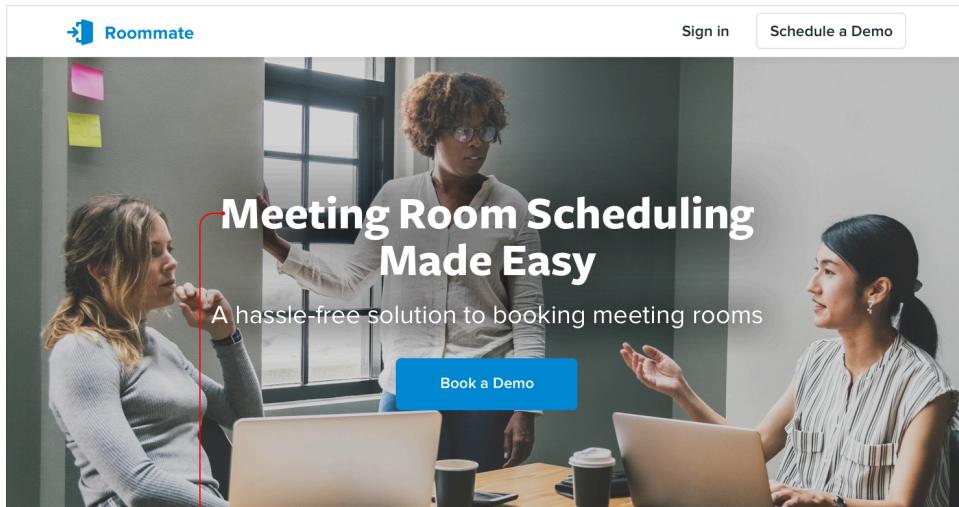
Blending mode: Multiply

This can also be a great way to make a background image pair more nicely with your existing brand colors.

Add a text shadow

If you want to preserve a bit more of the dynamics in a background image, a text shadow can be a great way to increase contrast only where you need it most.

You want it to look more like a subtle glow than an actual shadow, so use a large blur radius and don't add any kind of offset.



```
text-shadow: 0 0 50px hsla(0, 0%, 0%, .4);
```

It's still a good idea to reduce the overall image contrast, but combining that with a text shadow means you can reduce it a little less.

Everything has an intended size

Everyone knows that scaling bitmap images to larger than their original size is a bad idea — they immediately feel “fuzzy” and lose their definition.



But that's not the only way you can go wrong with scaling, even when you think you're playing it safe.

Don't scale up icons

If you're designing something that could use some large icons (*like maybe the “features” section of a landing page*), you might instinctively grab your favorite SVG icon set and bump up the size until they fit your needs.

They're vector images after all, so the quality isn't going to suffer if you increase the size, right?

A screenshot of a website featuring three service offerings. Each offering includes an icon, a title, and a brief description. The first offering is "Automatic rebalancing" with the subtitle "Adjust as the market changes". The second offering is "Dividend reinvesting" with the subtitle "Earn more as you grow". The third offering is "Advice from Advisors" with the subtitle "Advisors are available".

Automatic rebalancing Adjust as the market changes	Dividend reinvesting Earn more as you grow	Advice from Advisors are available
---	---	------------------------------------

While it's true that vector images won't degrade in quality when you increase their size, icons that were drawn at 16–24px are never going to look very professional when you blow them up to 3x or 4x their intended size. They lack detail, and always feel disproportionately "chunky".



Intended size (24px)



Scaled up (48px)



Drawn for large sizes (48px)

If small icons are all you've got, try enclosing them inside another shape and giving the shape a background color:



Scaled up (48px)



Larger but not scaled (48px)

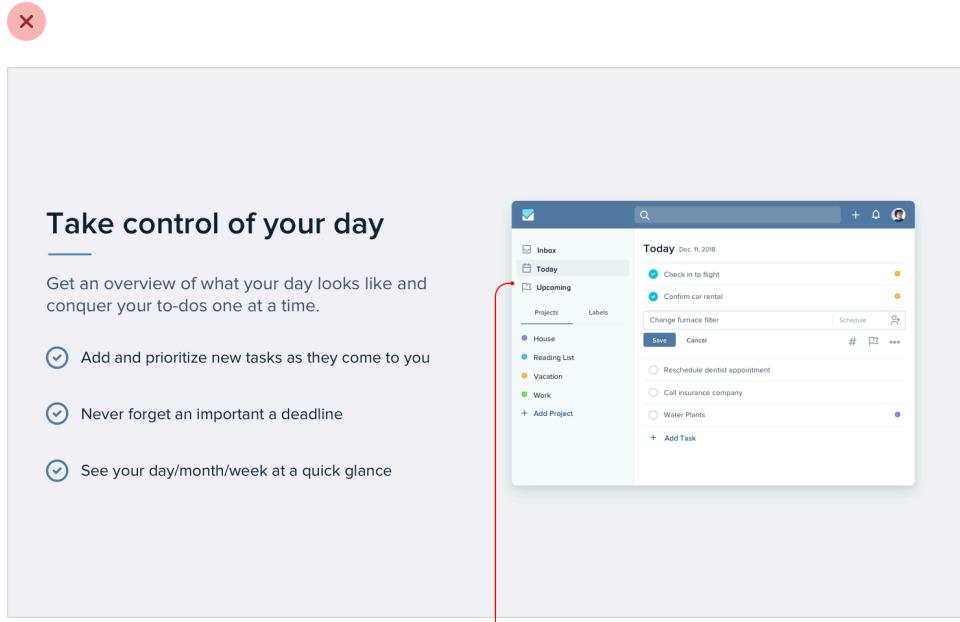
This lets you keep the actual icon closer to its intended size, while still filling the larger space.

A screenshot of a mobile application interface. At the top, there is a dark blue header bar with a green circular icon containing a white checkmark. Below the header, there are three horizontal cards, each featuring a small circular icon with a white outline and some descriptive text. The first card, on the left, has a green icon and the text "Automatic rebalancing" followed by "Adjust as the market changes". The second card, in the middle, has a purple icon and the text "Dividend reinvesting" followed by "Earn more as you grow". The third card, on the right, has an orange icon and the text "Advice from" followed by "Advisors ar".

Don't scale down screenshots

Say you want to include a screenshot of your app on that same features page.

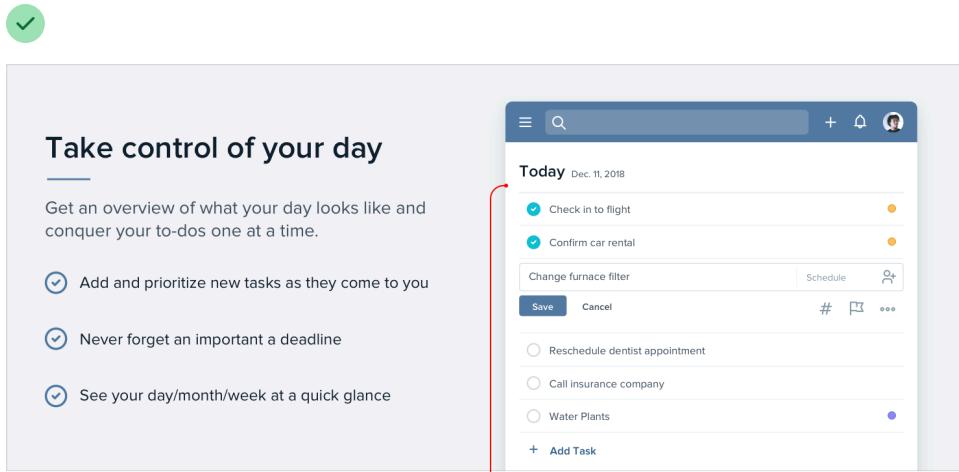
If you take a full-size screenshot and shrink it by 70% to make it fit, you'll end up with an image that's trying to cram way too much detail into far too little space.



The 16px font in your app becomes a 4px font in your screenshot, and visitors will be squinting with their eyeballs two inches from the screen, struggling to make out what all that text says.

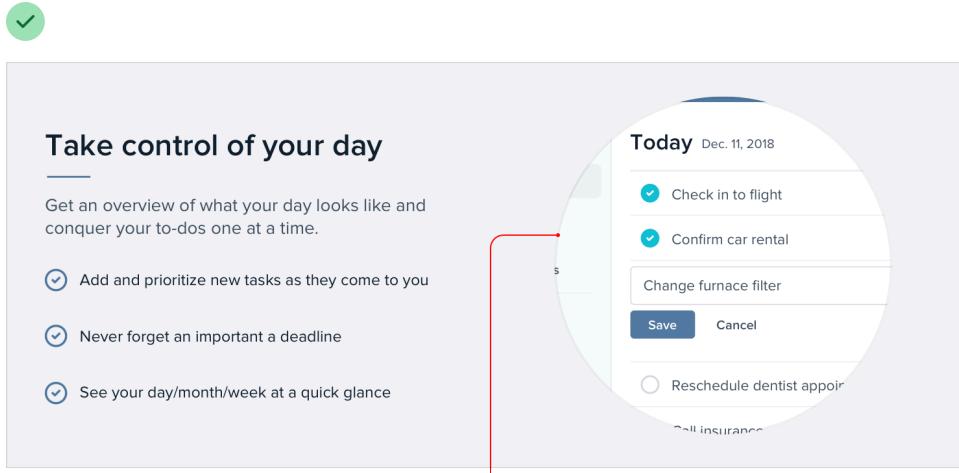
If you want to include a detailed screenshot in your design, take the screenshot at a smaller screen size (*like maybe your tablet layout*) and save a

lot of space for it so you don't have to shrink it as much:



Screenshot of smaller device layout

Or consider taking just a partial screenshot, so you can display it in less space without needing to scale it down:



Screenshot of specific area

If you really need to fit a whole-app screenshot in a tight space, try drawing a simplified version of the UI with details removed and small text replaced with simple lines:

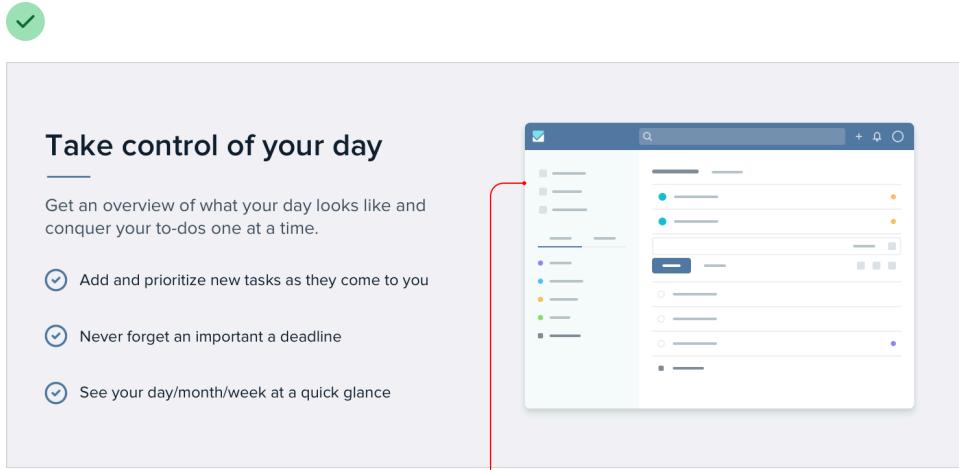


Illustration of simplified UI

It'll still communicate the big-picture design without tempting visitors to try and make out all of the details.

Don't scale down icons, either

Just as icons drawn to be used at 16px look chunky when you scale them up, icons intended to be used at larger sizes look choppy and fuzzy when you scale them down.

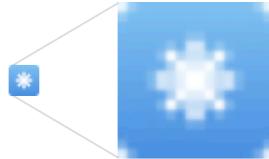
The most extreme example of this are favicons, those little icons you see next to the page title in a browser tab.

If you try to shrink a logo drawn at 128px down to favicon size, it all turns to

mush as the browser tries its best to render all of that detail in a tiny 16px square:



Original Icon



Icon Shrunk

A better approach is to redraw a super simplified version of the logo at the target size, so you control the compromises instead of leaving it up to the browser:



Original Icon



Simplified icon



Simplified icon shrunk