

User Interfaces

**100
PRACTICAL
UI/UX TIPS**

A collection of actionable
ideas to make your
interfaces easy to use

Contents

Intro	7
-------------	---

■ Typography

Place headings close to their paragraphs	8
Avoid text justification	9
Make text easy to scan	10
Make links look like links	11
Use proper line height	12
Don't make lines too lengthy	13
Use different techniques to make hierarchy	14
Do not over-emphasize	15
Keep an eye on contrast	16
Don't use pure black color	17
Handle text on images properly	18
Use modular type scale	19
Proportional vs tabular figures	20

■ Forms

Have enough space between inputs	21
Don't hide form tips	22
Show password rules right away	23
Avoid multicolumn layouts	24
Use labels, not placeholders	25
Be careful with dropdowns	26
Replace default file inputs	27
Autofocus the first input	28
Use reasonable input width	29
Don't erase credentials after unsuccessful login	30
Remember email in forgot password page	31
Dividing form into multiple steps	32

Use correct input type	33
Marking required/optional fields	34
Progressive disclosure	35
Collapsing checkboxes	36
■ Validation	
Autoscroll to the first error	37
Help users to fill forms without mistakes	38
Consider positive feedback as well	39
Put error messages in the right place	40
■ Landing pages	
Require fewer fields	41
Watch your text contrast on hero images	42
Don't have too much text	43
Wrong blocks separation	44
Illustrating your app	45
Text container width	46
False bottom effect	47
Visual hint for scrolling down below the fold	48
Small font size	49
■ Usability	
Place inputs where users expect them	50
Delaying the welcome email on mobile	51
Place labels above sliders	52
Verification code push notification/SMS	53
Don't put dangerous actions near frequent ones	54
False bottom effect (another example)	55
Put frequent options first	56
Avoid using carousels	57
Consider allowing undo instead of confirmation	58

Label your icons	59
Instantly check the verification code	60
Provide further instructions	61
Use a separate area for toggle password icon	62

■ Visuals

Animation speed	63
Normalize charts	64
Label country flags	65
Low detail vs high detail map	66
Buttons / tags confusion	67
Styling icons that have different size	68
Modal depth	69
Overlapping trick	70
Similarity law	71
Grouping piecharts sections	72
Styling images	73
Overusing primary color	74
Color saturation in dark mode	75
Harsh colored table borders	76
Watch your shadows	77
Icon consistency	78

■ Empty/loading states

Leverage empty states	79
Avoid vertical layout shifts	80
Avoid width changing when buttons are loading state	81
Wrong loader placement	82
Don't show loader right away	83
Use smart/progressive loaders	84

■ Focus and user attention

De-emphasize other elements	85
Highlighting input search trick	86
Don't put too much text in notifications	87
Don't show multiple hints at once	88
Use one primary button in dialogs	89
Use face and fingers to control attention	90

■ Navigation

Don't hide navigation links	91
Increase clickable area	92
Vertical vs Horizontal navigation	93
How to place more links in horizontal navigation	94
Breadcrumbs pattern	95
Show active link	96
Fluid vs fixed sidebar	97

■ Misc

Color blindness & icons	98
Use empty alt tag for decorative images	99
Aligning items using baseline	100
Show the number of votes along with rating	101
Instantly show how many goods were found	102
Filters indicator trick	103
How to make tables responsive	104
Group elements logically	105
Don't forget to put the post date	106
Remember that data can be large	107
Afterward	108

Intro

Hello, and thank you for purchasing this book!

My name is Victor.

For the whole year, I've been tweeting UI/UX tips, threads, curating resources. Some of them got positive comments. Some didn't.

In total, I made around 120 tips. Then I went through them, removed those that weren't that good, enhanced the rest and made a book out of them.

For most of the tips, I made illustrations and tried to find some additional links that prove every tip or provide further explanations.

Every page contains exactly one tip. I removed all the useless fluff. And I sincerely hope that you'll find this book helpful.

By the way, I'll dedicate the following year to YouTube, where I have so many plans:

- Make UI/UX reviews
- Make videos about UX Laws
- Cover those topics that cannot be represented as 1-2 tweets
- And much, much more

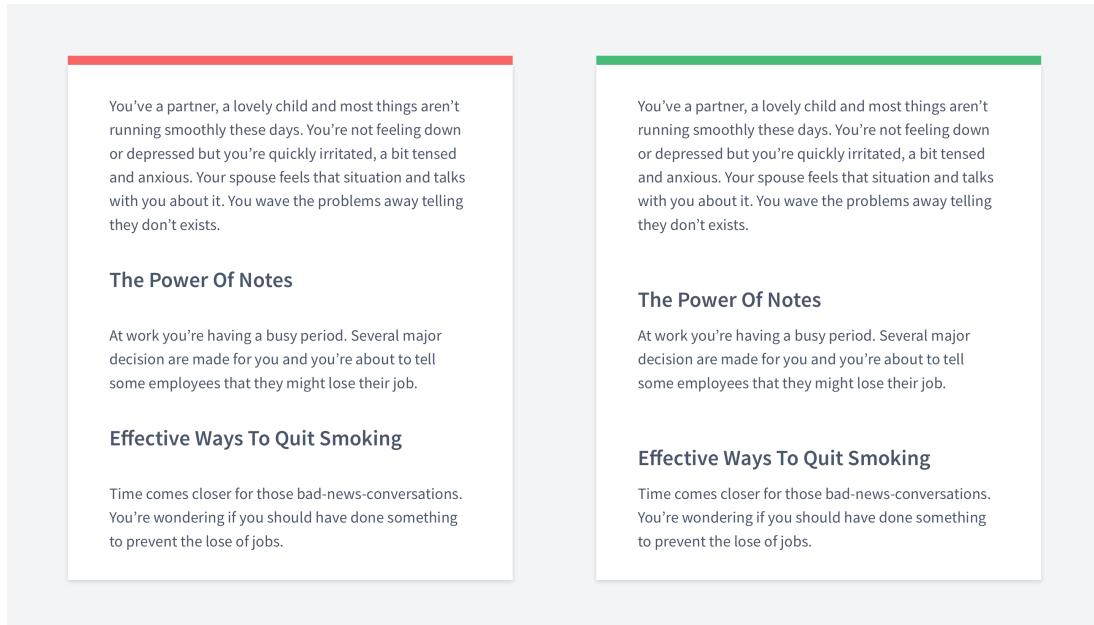
It's going to be an exciting journey! So don't forget to [check it out!](#)

Typography

Place headings close to their paragraphs

The [proximity principle](#) states that items close together are likely to be perceived as part of the same group — sharing similar functionality or traits.

This is one of the most used laws across user interfaces, in typography as well.



In the first picture, the headings have the same whitespace between the first and the second paragraph. In this case, you don't have clearly separated paragraphs, and it would be hard to say what heading belongs to what paragraph. At least it will take some time to figure it out.

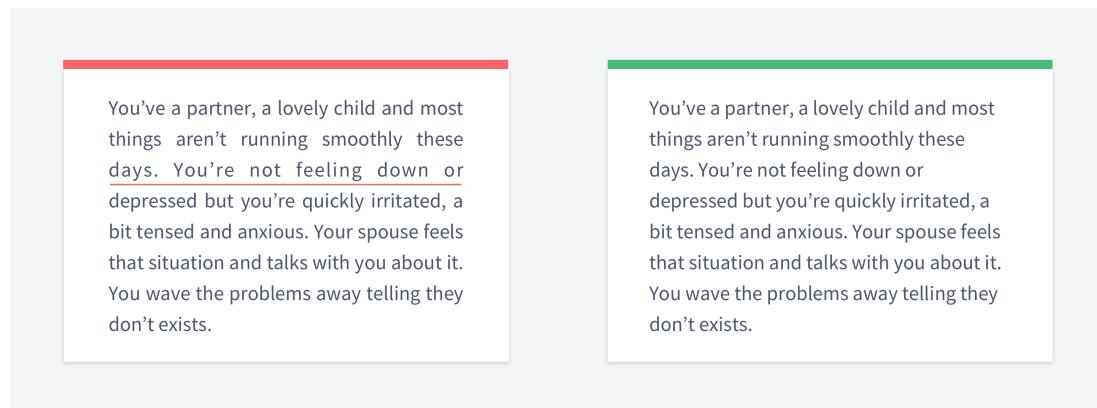
In the second picture, the headings are placed closer to the paragraphs to which they belong.

Avoid text justification

Justifying text is usually considered a bad practice.

The reason is that it tends to create [rivers](#) in text. Rivers are the white gaps in the text when there is too much space between words on consecutive text lines.

First of all, the text looks unbalanced and hard to read.



But what's might be more important is that it affects users with [dyslexia](#).

There are many studies and articles on the Internet related to this issue. For example, the British Dyslexia association [advises aligning](#) text to the left side.

Forms

Have enough space between inputs

The proximity law we talked about [earlier](#) can also be applied to form elements.

A common mistake is when input labels have the same spacing between inputs. Without additional effort, you cannot easily say to which input the label belongs.

The image shows two side-by-side form examples. Each example consists of a white rectangular box with a thin gray border. Inside each box, there is a horizontal red bar at the top. Below the red bar, there are four input fields arranged vertically. Each input field has a label to its left and a corresponding empty text box below it. In the first example (left), the vertical spacing between the labels and the input boxes is very small. In the second example (right), the vertical spacing is significantly larger, creating more breathing room between the labels and the input fields.

First Name	First Name
<input type="text"/>	<input type="text"/>
Last Name	Last Name
<input type="text"/>	<input type="text"/>
Email	Email
<input type="text"/>	<input type="text"/>
Password	Password
<input type="text"/>	<input type="text"/>

Remember: "*Whitespace is like air: it is necessary for design to breathe.*" – Wojciech Zieliński

Don't hide form tips

It looks like a good solution to hide hints under an icon because we can save some space. The UI will look cleaner.

But, If the information hidden under a tooltip is vital for completing the form, [don't hide it.](#)

The image displays two side-by-side wireframe mockups of a registration form. Both forms include three input fields: 'Name', 'Email', and 'Phone Number'. Each input field is accompanied by a small circular icon containing a question mark, which typically serves as a tooltip trigger. In the left mockup, the input fields have a red border, while the right mockup has a green border. Below each input field, there is descriptive text explaining its purpose: 'So that we know how to address to you' for Name, 'We will send the order receipt to this email' for Email, and 'We will notify you via SMS of the order status' for Phone Number. A blue 'Register' button is located at the bottom of each form.

The reason is that in order to complete the form, users will need to hover over the tooltips constantly. Instead, it would be much better to show the hint right away. For example, below inputs.

Validation

Autoscroll to the first error

Sometimes the size of a form can exceed the monitor viewport.

Now imagine the following situation. A user fills the form, hits the submit buttons, and nothing happens for them. However, the problem is that they've just made a mistake but couldn't see it because the incorrect fields are not in the viewport.

You have a couple of options.

The first one is to display validation errors next to the submit button. But this is considered a bad practice.

A great alternative is to scroll users up to the first invalid field. Remember that it's **crucial** that you have the scroll animation. Otherwise, if the screen suddenly changes without any animation, it'd confuse users.

Landing pages

Require fewer fields

The more fields you require, the more significant becomes the barrier for users to fill the form. Thus, [the conversion rate decreases](#). There is also [a good question on UX stackexchange](#) related to conversion rates.

You don't need the user name to let them use your app.

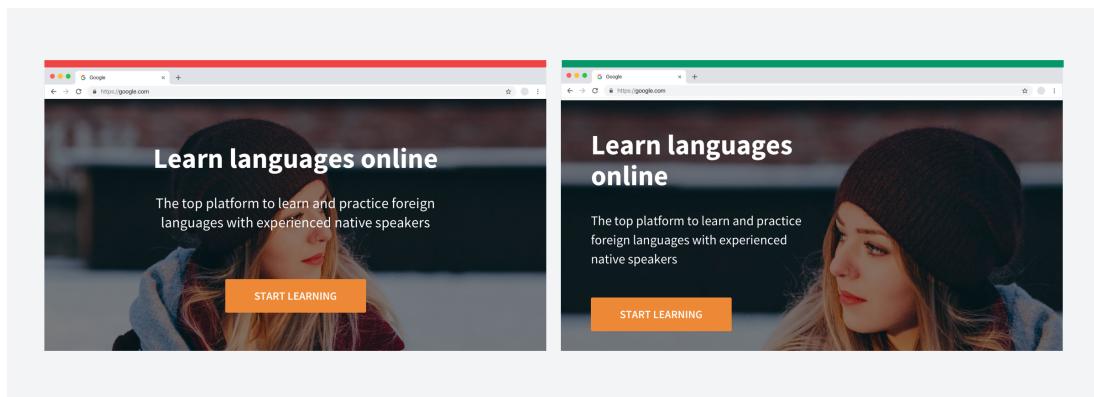
You don't need the password confirmation since they can restore it via the reset password form.

You probably don't even need to verify users' emails to let them try your app out. You can allow them to use the features that don't require email verification. And only ask it when they want to do something crucial that requires verified email.

Watch your text contrast on hero images

It often happens that you have a background image in the hero section. The image might be a photo of a person or something else. There are two common mistakes.

1. Placing text on an important part of the image. For example, right on a person's face. This is quite easy to fix. You should adjust the image so that the text doesn't cover any essential parts of the picture.
2. Having low contrast between the text and the image. You can fix this one in a few ways, but the basic idea is to add an overlay on the image to place the text on a dark background, which guarantees the proper contrast ratio.



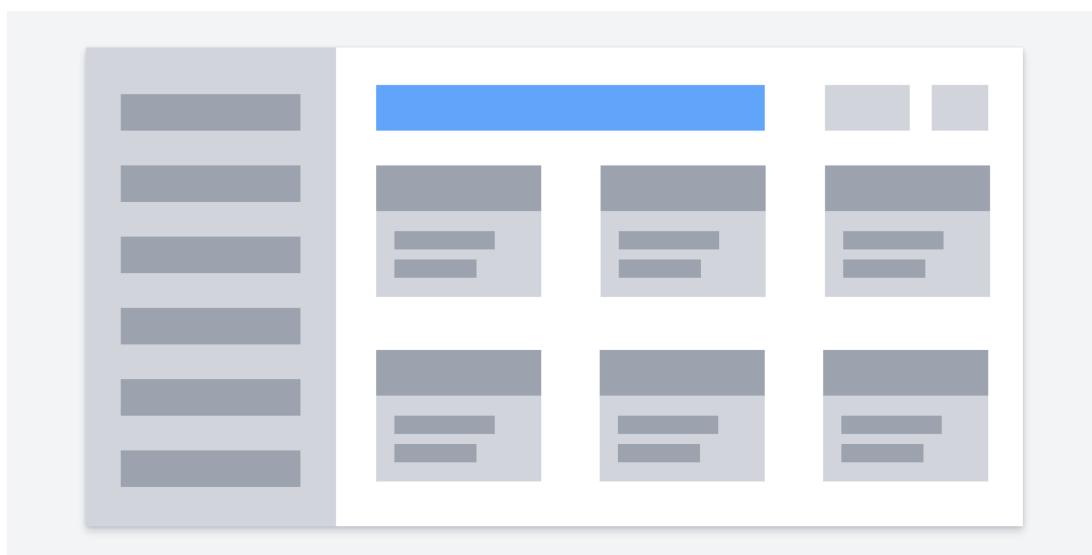
Usability

Place inputs where users expect them

There is a UX law called [Jakob's Law](#). Users spend most of their time on other sites. This means that users prefer your site to work the same way as all the other sites they already know.

Say you need to put a search bar somewhere on your site. The idea is that you should choose a place that is commonly used, thus, familiar to most users.

For example, for a search bar, it'd be best to place it somewhere at the top of the page.

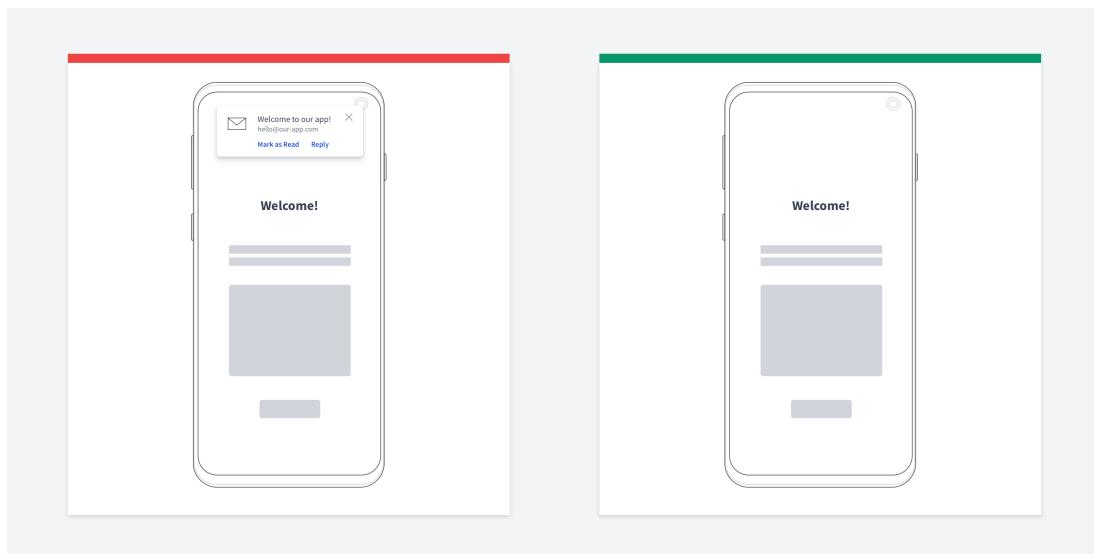


Delaying the welcome email on mobile

An opinionated trick to improve mobile UX.

If the user signed up via the mobile app, delay the "welcome email" delivery.

If it doesn't contain any important info, you don't need to bother users with yet another notification that they will swipe.

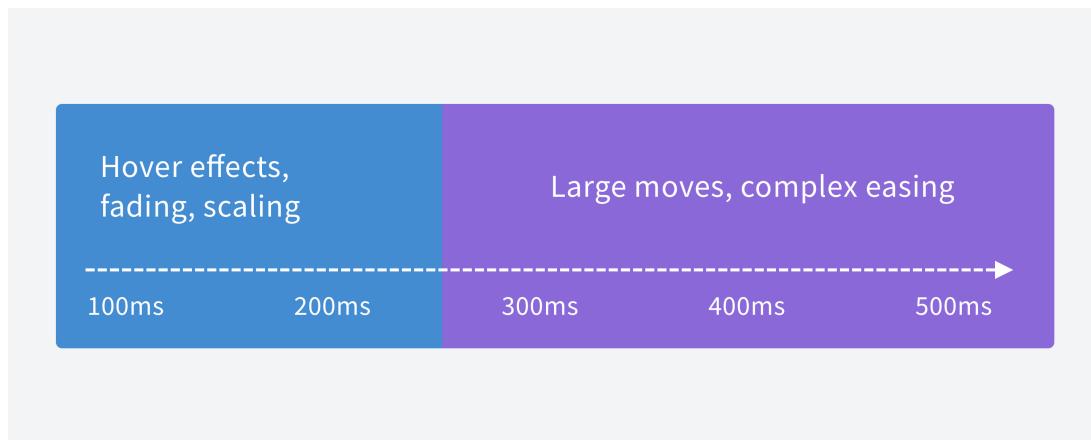


Visuals

Animation speed

Animation speed matters.

- less than 100ms - won't be perceived at all
- 100ms - 200ms - should be used for micro animations (e.g. hover effects)
- 200ms - 300ms - an intermediate step, might be used for micro animations too
- more than 300ms - complex animations



Here is a [detailed guide](#) for animation speed.

Normalize charts

Make the Y-axis maximum value slightly higher than the peak bar value.

If your maximum Y-axis value is 3-4 times higher than the peak bar value, you'll have a lot of free space. Plus, the bars will look zoomed-out.

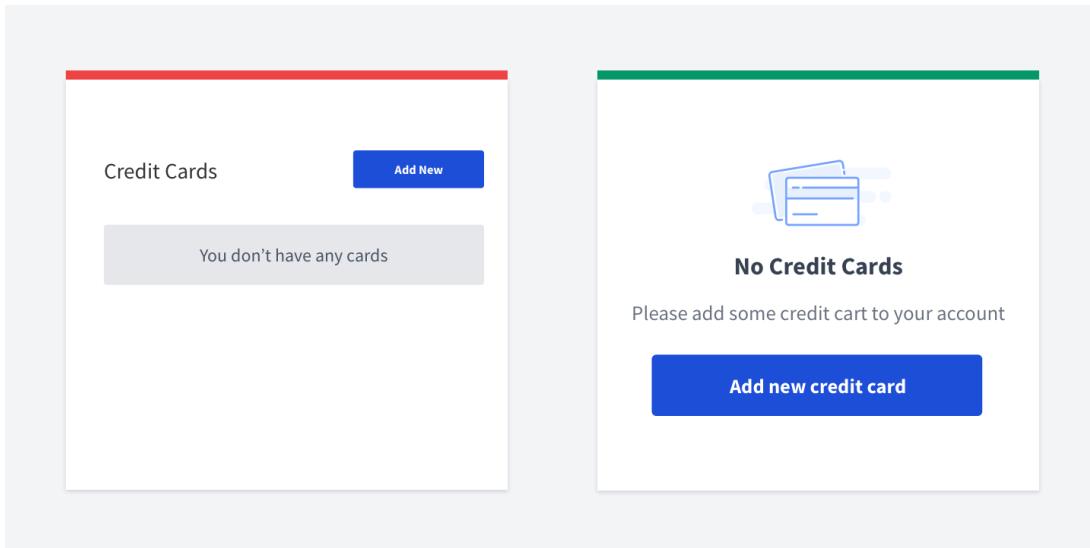


Empty/loading states

Leverage empty states

When you need to display a list of records, a typical pattern looks like on the left image.

But if there are no records, you can use available space for a more informative and engaging message.

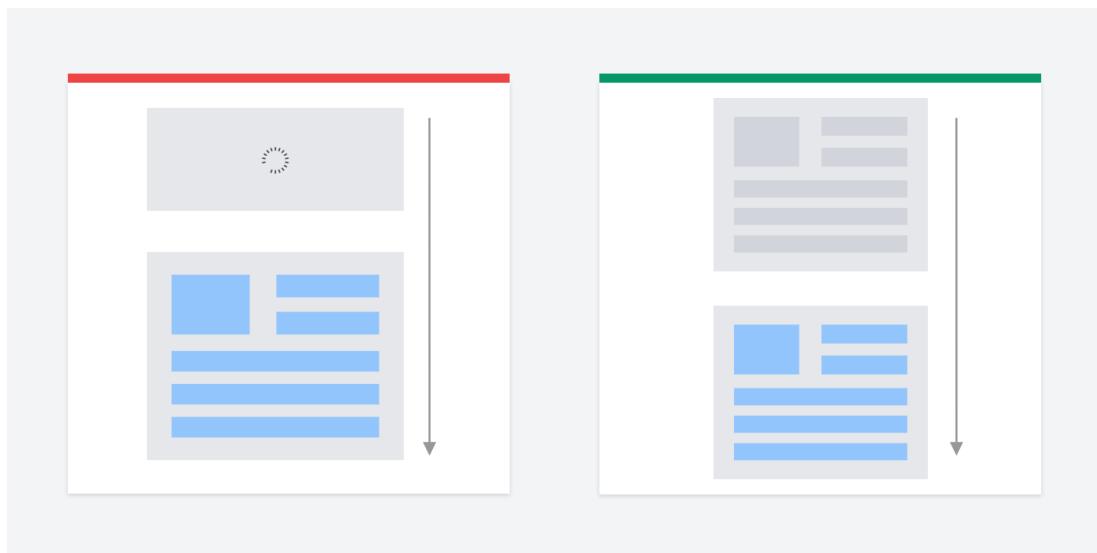


Avoid vertical layout shifts

If you use loaders when loading big content, you'll have a layout shift after the content is loaded.

While if you use skeletons, the height will stay the same.

It's best to use loaders in such things as buttons. At the same time, skeletons work well for layout/articles/large content.



Focus and user attention

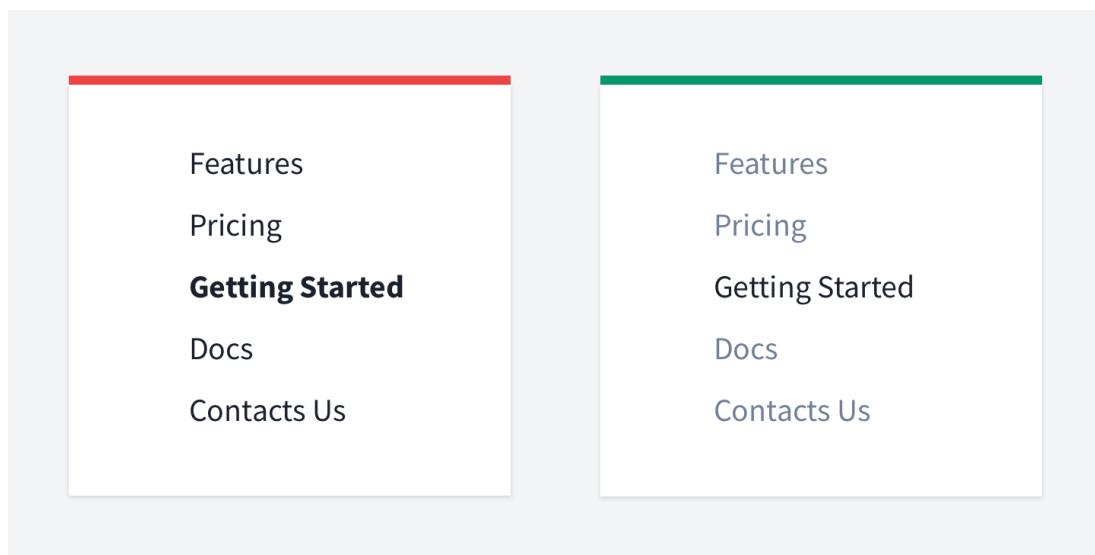
De-emphasize other elements

If you need to emphasize an element, for example, a link, you can do it in two ways.

One way is to simply make it bolder.

But it can make the layout shift because bold links take more place. But on top of that, it just doesn't look good.

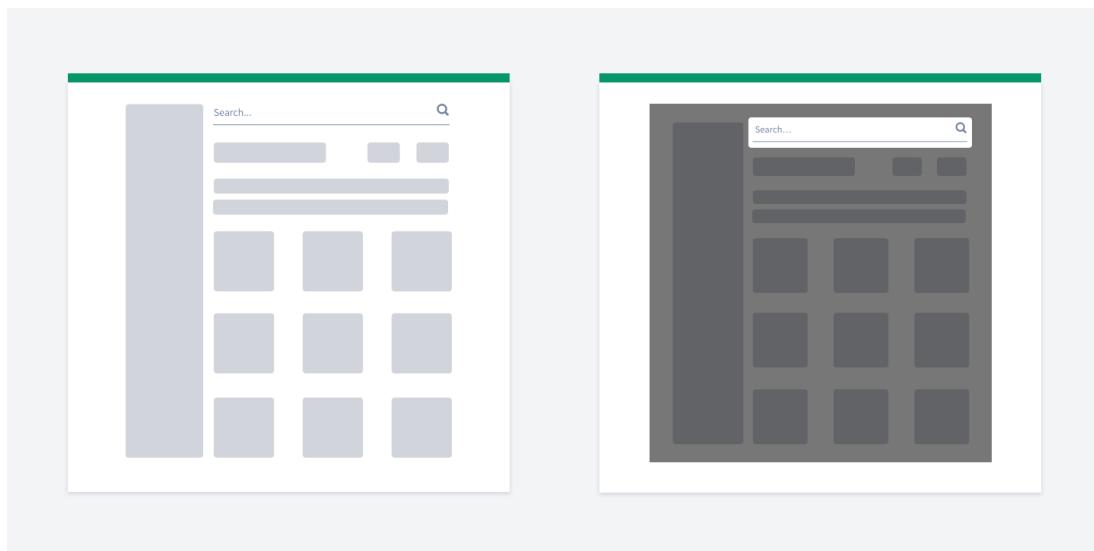
Consider the second way. De-emphasize other elements instead. It looks much cleaner.



Highlighting input search trick

A very simple yet a nice trick: when the user focuses on the search input, put a dark overlay on the rest part of the site.

It'll probably make the user more concentrated on the search task. At least I saw quite a few sites doing this.



Misc

Color blindness & icons

Do not rely only on colors when building UI.

[1 in 12 men has some degree of color blindness](#). Use text messages and icons along with colors.

Normal vision

Coin	Price	Market Cap	24h
BTC	\$37,816.39	\$710,033,690,297	4.3%
ETH	\$2,486.51	\$290,831,713,402	3.1%
BNB	\$321.93	\$54,127,899,904	1.9%

Green blind / Deutanopia

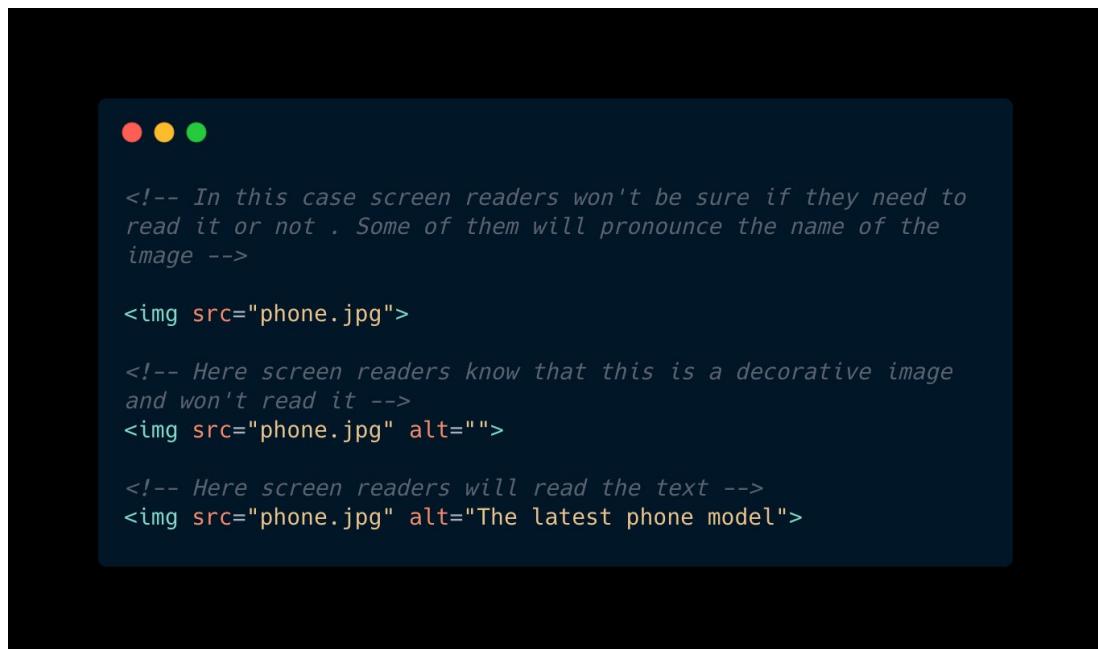
Coin	Price	Market Cap	24h
BTC	\$37,816.39	\$710,033,690,297	4.3%
ETH	\$2,486.51	\$290,831,713,402	3.1%
BNB	\$321.93	\$54,127,899,904	1.9%

Use empty alt tag for decorative images

Did you know that having an empty alt attribute in the `img` tag is not the same as not having it at all?

If your image serves decorative purposes, an [empty alt tag tells screen readers to ignore the image](#).

If you don't have any alt attribute at all, the screen reader won't be sure if it's decorative or not.



Afterward

There is one secret tip that I haven't told you about yet.

Every rule has an exception. In every situation, there are many conditions. Sometimes the business doesn't care about accessibility, sometimes your boss "knows" how to do things better, sometimes, your research was wrong.

It's alright to make mistakes.

The golden rule here is to iterate. Research, iterate, try and then try again.

That's it guys. Don't forget to follow me on [YouTube](#). In the new year, I plan to make many awesome videos. At least I'll try to.

Take care.