

[< Go to the original](#)

Selection controls—UI component series

Making choices never was more satisfying. A detailed look at checkboxes, radio buttons, and toggles design.

**Taras Bakusevych**[Follow](#) UX Collective a11y-light ~6 min read · June 24, 2020 (Updated: November 11, 2022) · Free: Yes

The word "toggle" is a reference to a switch with a short handle that alternates between two states each time it is activated. You encounter it every time you "switch" on the lights.

As for "Radio Buttons" the word comes from the car radios that as common practice had a set of buttons under the dial that could mechanically store station presets, so the user switch between stations faster. Pressing one of these buttons would cause it to stay down until another was pressed.

Origins



Checkboxes — are used when there are one or many independent options and users may select any number of choices, including none, one, or several.

Radio buttons — are used when there is a list of two or more options that are mutually exclusive and users must select only one of them.

Toggle switches — are used when there are two mutually exclusive options and always have a default value. Toggles selection takes effect immediately.

Selection controls

Types

Checkboxes	Radio buttons	Toggle-Switch
Colors	Shapes	Textures
Choice chips		Multi-select chips

by Taras Bakushevych

Multi-select chips — are a compact alternative to checkboxes. Allow users to select multiple options and are primarily used for filtering on mobile.

Selection controls have been used in user interfaces for a long time, so we have formed strong expectations on how they should behave. Here is a simple cheat sheet that you can follow to choose the right type for the right situation.

Selection controls					
How to know when use what control?					
	Radio Buttons	Checkboxes	Single checkbox	Toggle Switch	
How many selections users can make?	1 	0-All 	2 (yes/no) 	2 (on/off) 	
How would you describe the choices?	Mutually exclusive 	Independent of each other 	Mutually exclusive 	Mutually exclusive 	
Is there a default option?	Yes 	No 	Yes 	Yes 	
When selection will take effect?	After users submit 	After users submit 	After users submit 	Immediately 	

by Taras Bakunavych

Inspiration was taken from NNG article "[Checkboxes vs. Radio Buttons](#)"

States

Checkboxes & radio buttons can be selected or unselected, and Toggles can be Off or On. All of them have enabled, disabled, hover, focused, and pressed states. And I know it looks like a lot), but it's good to create all those states for reliable interaction.

States

Enabled Selected/On	<input checked="" type="checkbox"/>	<input checked="" type="radio"/>	<input checked="" type="button"/>
Disabled Selected/On	<input checked="" type="checkbox"/>	<input checked="" type="radio"/>	<input checked="" type="button"/>
Hover Selected/On	<input checked="" type="checkbox"/> 	<input checked="" type="radio"/> 	<input checked="" type="button"/> 
Focus Selected/On	<input checked="" type="checkbox"/>	<input checked="" type="radio"/>	<input checked="" type="button"/>
Pressed Selected/On	<input checked="" type="checkbox"/>	<input checked="" type="radio"/>	<input checked="" type="button"/>
Enabled Unselected/Off	<input type="checkbox"/>	<input type="radio"/>	<input type="button"/>
Disabled Unselected/Off	<input type="checkbox"/>	<input type="radio"/>	<input type="button"/>
Hover Unselected/Off	<input type="checkbox"/> 	<input type="radio"/> 	<input type="button"/> 
Focus Unselected/Off	<input type="checkbox"/>	<input type="radio"/>	<input type="button"/>
Pressed Unselected/Off	<input type="checkbox"/>	<input type="radio"/>	<input type="button"/>

by Taras Bakusevych

Don't forget about the indeterminate state

State for a checkbox that is neither checked nor unchecked. The state is not fully determined, hence the name- indeterminate. Created to handle a scenario when there's a parent checkbox that has multiple children, some of which are selected and others are not.

Indeterminate States

Enabled Indeterminate	
Disabled Indeterminate	
Hover Indeterminate	
Focus Indeterminate	
Pressed Indeterminate	

▼ Colors

- Red
- Green
- Brown
- Purple

► □ Shapes

Indeterminate Checkboxes

by Taras Bakusevych

Stop misusing toggle switches

Do not create hierachal structures with toggles. It's more visually distracting and can create a false impression that all child options are On/Off.

👉 Useful tip

Stop misusing toggle switches

▼ Colors

- Reds

▼ Greens

- Lime
- Mint
- Jade

► Shapes

▼ Colors

- Reds

▼ Greens

- Lime
- Mint
- Jade

► □ Shapes

Don't

Do

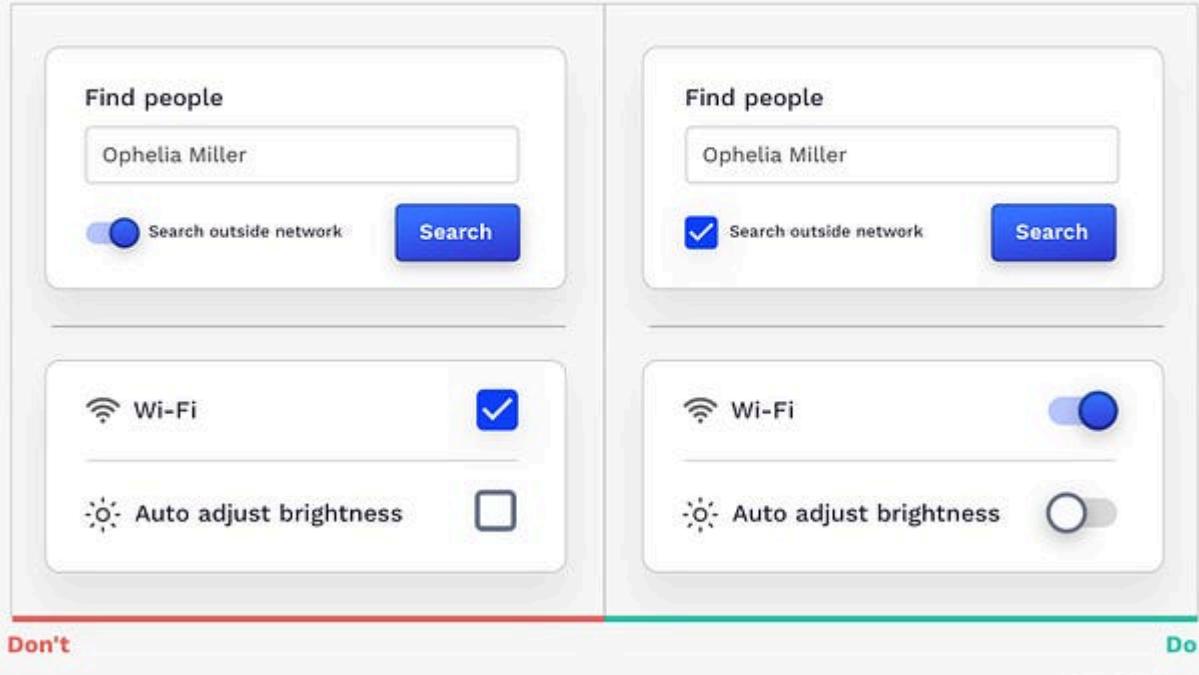
by Taras Bakusevych

Use toggles when selection will immediately take effect

switch with a single checkbox.

👉 Useful tip

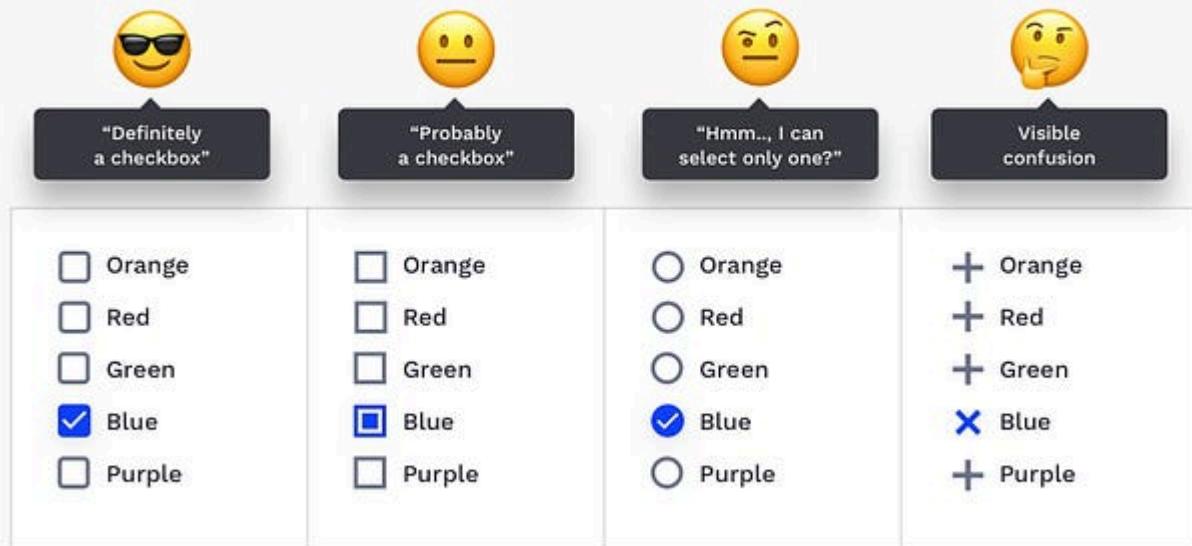
Use toggles when selections will immediately take effect



Un-orthodox styling can confuse users

Any deviation from what is considered standard for the platform you design will add additional cognitive load for users. I often see circular checkboxes that can be easily confused with radio buttons.

Un-orthodox styling can mislead users

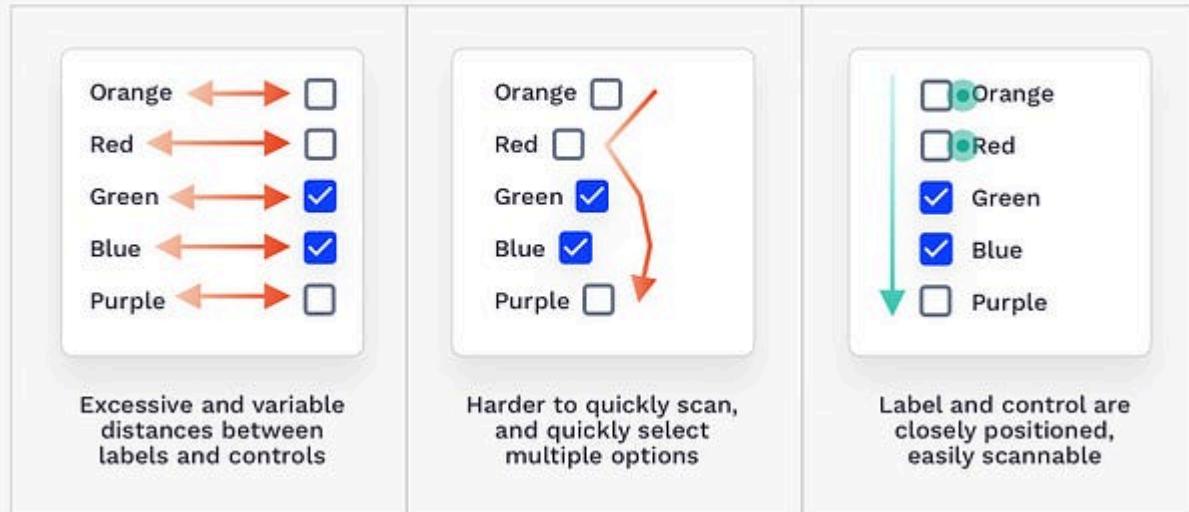


by Taras Bakusevych

Presenting options in the vertical list will help users easily scan them

Left-aligned controls and labels will work best as they closely positioned to each other. This supports the fastest completion times and fewer mistakes. Right-aligned controls have some benefits on mobile, as labels and buttons, can't get too far apart. Having a button close to the right edge makes it more easily reachable when holding the device in one hand. Also, we are not obscuring the labels with a finger during selection.

Left-aligned controls & labels work best



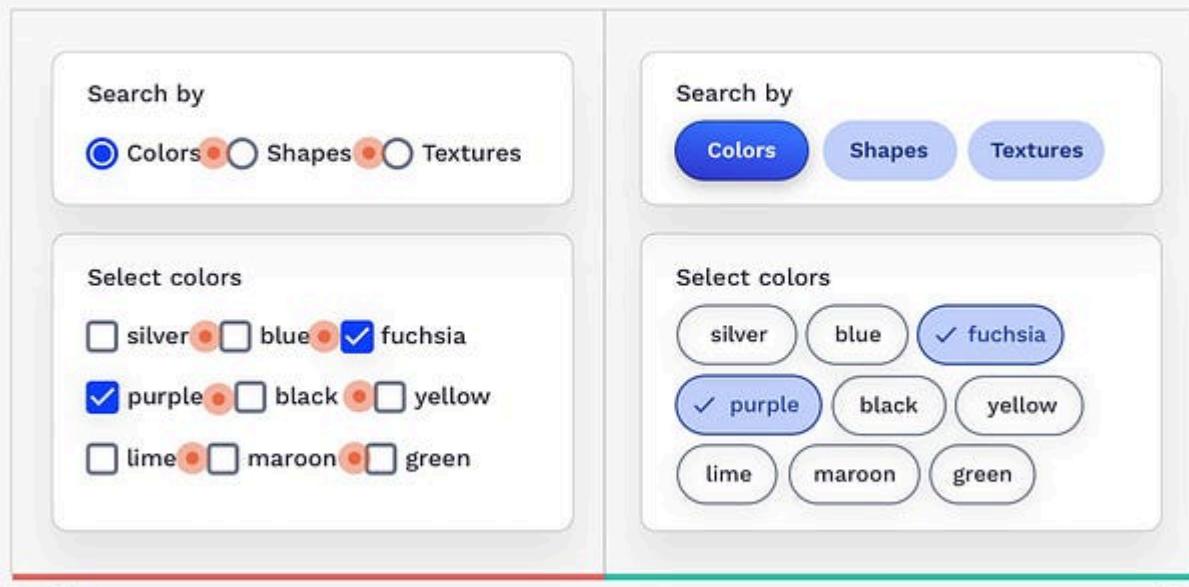
by Taras Bakusevych

Consider chips if vertical space is an issue

You should prevent a situation when it's really difficult to see which control is selected (make sure to space the buttons and labels). Consider using chips to clearly visually separate options.

👉 Useful tip

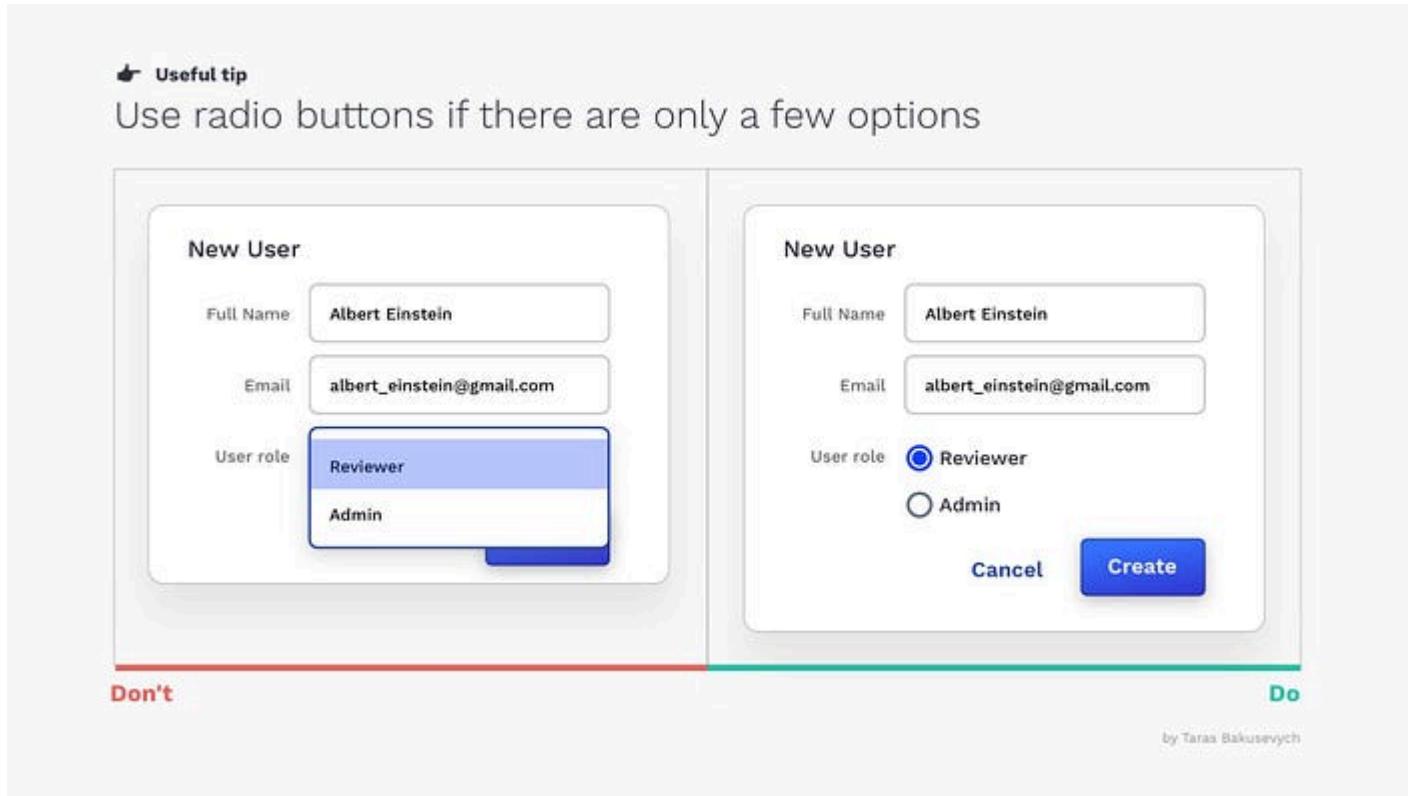
Consider selection chips if vertical space is an issue



by Taras Bakusevych

If possible, use radio buttons rather than drop-down menus

Making all options permanently visible so that users can easily compare them reduces cognitive load and help forms be more transparent.



Use dropdowns for numerous or similar options

If the number of options is more than 6–7 you should consider putting them in the dropdown as users anyway will not be able to keep all of them in mind. The same will apply to predictable, similar, or incremental options like (zoom – 10%, 20%, 30% ..).

Use dropdowns for numerous or similar options

The image shows two examples of user interface design for selecting a country. The left example, labeled 'Don't', features a horizontal list of countries with radio buttons. The right example, labeled 'Do', shows a dropdown menu where the first item, 'Afghanistan', is highlighted with a blue background and white text, while the other options are in a smaller, standard font.

Personal info

Date of birth

24 Jan 2001

Country

Afghanistan

Albania

Andorra

Antigua and Barbuda

Argentina

Personal info

Date of birth

24 Jan 2001

Country

Afghanistan

Albania

Andorra

Antigua and Barbuda

Don't
Do

by Taras Bakusevych

It is better to have a selected radio button by default

Users cannot deselect and set the radio button back to its original state once one has been selected. If users might need to refrain from making a selection, you should provide a radio button labeled "None." Structure the list of options in a logical order, and harness the power of suggestion.



Select a single radio button by default in most cases

The image shows two examples of user interface design for selecting a color theme. The left example, labeled 'Don't', shows a list of four color options: Red, Green, Blue, and Purple, each with an empty radio button. The right example, labeled 'Do', shows the same list, but the first option, 'Red', has a filled radio button, indicating it is the selected default.

Color theme

Red

Green

Blue

Purple

Color theme

Red

Green

Blue

Purple

Don't
Do

by Taras Bakusevych

Write clear labels, avoid negations

Avoid situations when users need to tick the box in order for something **not** to happen.

👉 Useful tip

Write clear labels, avoid negations

The image shows two side-by-side registration forms. Both forms have a text input field for 'Name' containing 'Ophelia Miller'. Below it is a checkbox labeled 'I don't want to receive updates about the latest deals and promotions.' A red arrow points to the word 'don't' in this label. The 'Don't' version has a blue checkmark in the box. The 'Do' version has a blue checkmark in the box and the label is changed to 'I want to receive updates about the latest deals and promotions.' Both forms have a 'Register' button at the bottom.

Don't **Do**

by Taras Bakusevych

Highlight selected options to draw users attention

Visually differentiate selected options from others, especially important for rows selection in data tables.

👉 Useful tip

Highlight selected options to draw users attention

The image shows two side-by-side data tables. Both tables have columns for 'Name', 'Status', and 'Last update'. They list five items: Orange (Active, 12-06-2020), Red (Active, 12-06-2020), Green (Review, 12-06-2020), Blue (Draft, 12-06-2020), and Purple (Draft, 12-06-2020). In the 'Don't' version, the checkboxes are standard gray. In the 'Do' version, the selected checkboxes ('Green' and 'Blue') are highlighted with a blue border and a blue checkmark. The 'Do' version also features a light blue background for the entire row containing the selected items.

Don't **Do**

by Taras Bakusevych

Support bulk select and clear

Selecting or clearing multiple items at once should be effortless for users.

👉 Useful tip

Support bulk select and clear for large lists of options

Name	Status	Last update
<input type="checkbox"/> Orange	Active	12-06-2020
<input type="checkbox"/> Red	Active	12-06-2020
<input checked="" type="checkbox"/> Green	Review	12-06-2020
<input checked="" type="checkbox"/> Blue	Draft	12-06-2020
<input type="checkbox"/> Purple	Draft	12-06-2020

-		24 Items selected	Clear
Name	Status	Last update	
<input type="checkbox"/> Orange	Active	12-06-2020	
<input type="checkbox"/> Red	Active	12-06-2020	
<input checked="" type="checkbox"/> Green	Review	12-06-2020	
<input checked="" type="checkbox"/> Blue	Draft	12-06-2020	

Don't **Do**

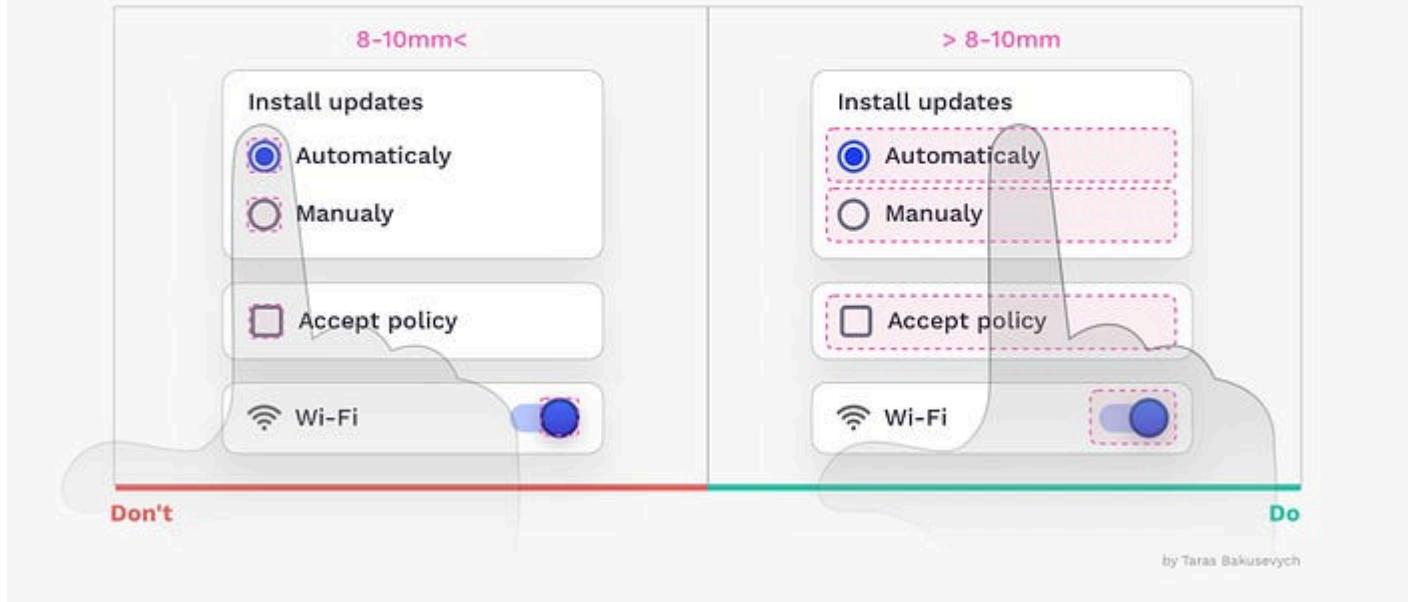
by Taras Balusevych

Make clickable areas large enough for reliable interaction

Include both buttons and labels in the generous clickable/tappable area.

According to Fitts's Law the size of the target plays an important role. Checkboxes and radio buttons are generally tiny and can be tricky to click or tap, especially on mobile.

Make controls clickable areas large enough for reliable interaction



You may also like

Components series — a detailed look on the components we use every day and how best to design them.

Text fields & Forms design — UI components series

Simple rules that will help you design forms users will like to complete

uxdesign.cc

The anatomy of a button — UI component series

All you need to know about button design, one of the main interactive building blocks for creating a user interface.

uxdesign.cc



an option. Build the design community you believe in.

#ui

#mobile

#product-design

#visual-design

#ux