# Reboot

## Contents

## Approach

Reboot builds upon Normalize, providing many HTML elements with somewhat opinionated styles using only element selectors. Additional styling is done only with classes. For example, we reboot some `<table>` styles for a simpler baseline and later provide `.table`, `.table-bordered`, and more.

Here are our guidelines and reasons for choosing what to override in Reboot:

- Update some browser default values to use `rem`s instead of `em`s for scalable component spacing.

- Avoid `margin-top`. Vertical margins can collapse, yielding unexpected results. More importantly though, a single direction of `margin` is a simpler mental model.
- For easier scaling across device sizes, block elements should use `rem`s for `margin`s.
- Keep declarations of `font`-related properties to a minimum, using `inherit` whenever possible.

## CSS variables

Added in v5.1.1

With v5.1.1, we standardized our required `@import`s across all our CSS bundles (including `bootstrap.css`, `bootstrap-reboot.css`, and `bootstrap-grid.css` to include `_root.scss` . This adds `:root` level CSS variables to all bundles, regardless of how many of them are used in that bundle. Ultimately Bootstrap 5 will continue to see more CSS variables added over time.

## Page defaults

The `<html>` and `<body>` elements are updated to provide better page-wide defaults. More specifically:

- The `box-sizing` is globally set on every element—including `*::before` and `*::after`, to `border-box`. This ensures that the declared width of element is never exceeded due to padding or border.
  - No base `font-size` is declared on the `<html>`, but `16px` is assumed (the browser default). `font-size: 1rem` is applied on the `<body>` for easy responsive type-scaling via media queries while respecting user preferences and ensuring a more accessible approach. This browser default can be overridden by modifying the `$font-size-root` variable.
- The `<body>` also sets a global `font-family`, `font-weight`, `line-height`, and `color`. This is inherited later by some form elements to prevent font inconsistencies.
- For safety, the `<body>` has a declared `background-color`, defaulting to `#fff`.

## Native font stack

Bootstrap utilizes a "native font stack" or "system font stack" for optimum text rendering on every device and OS. These system fonts have been designed specifically with today's devices in mind, with improved rendering on screens, variable font support, and more. Read more about native font stacks in this *Smashing Magazine* article.

```
$font-family-sans-serif:
  // Cross-platform generic font family (default user interface font)
  system-ui,
  // Safari for macOS and iOS (San Francisco)
  -apple-system,
  // Windows
  "Segoe UI",
  // Android
  Roboto,
  // Basic web fallback
  "Helvetica Neue", Arial,
  // Linux
  "Noto Sans",
  "Liberation Sans",
  // Sans serif fallback
  sans-serif,
  // Emoji fonts
  "Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol", "Noto Color Emoji" !default;
```

Note that because the font stack includes emoji fonts, many common symbol/dingbat unicode characters will be rendered as multi-colored pictographs. Their appearance will vary, depending on the style used in the browser/platform's native emoji font, and they won't be affected by any CSS `color` styles.

This `font-family` is applied to the `<body>` and automatically inherited globally throughout Bootstrap. To switch the global `font-family`, update `$font-family-base` and recompile Bootstrap.

## CSS variables

As Bootstrap 5 continues to mature, more and more styles will be built with CSS variables({{< docsref "/customize/css-variables" >}}) as a means to provide more real-time customization without the need to always recompile Sass. Our approach is to take our source Sass variables and transform them into CSS variables. That way, even if you don't use CSS variables, you still have all the power of Sass. **This is still in-progress and will take time to fully implement.**

For example, consider these `:root` CSS variables for common `<body>` styles:

{{< scss-docs name="root-body-variables" file="scss/_root.scss" >}}

In practice, those variables are then applied in Reboot like so:

{{< scss-docs name="reboot-body-rules" file="scss/_reboot.scss" >}}

Which allows you to make real-time customizations however you like:

```html
<body style="--bs-body-color: #333;">
  <!-- ... -->
</body>
```

## Headings and paragraphs

All heading elements—e.g., `<h1>`—and `<p>` are reset to have their `margin-top` removed. Headings have `margin-bottom: .5rem` added and paragraphs `margin-bottom: 1rem` for easy spacing.

Heading

Example

{{< markdown >}}`<h1></h1>`{{< /markdown >}}

h1. Bootstrap heading

{{< markdown >}}`<h2></h2>`{{< /markdown >}}

h2. Bootstrap heading

{{< markdown >}}`<h3></h3>`{{< /markdown >}}

h3. Bootstrap heading

{{< markdown >}}`<h4></h4>`{{< /markdown >}}

h4. Bootstrap heading

{{< markdown >}}`<h5></h5>`{{< /markdown >}}

h5. Bootstrap heading

{{< markdown >}}`<h6></h6>`{{< /markdown >}}

h6. Bootstrap heading

## Lists

All lists—`<ul>`, `<ol>`, and `<dl>`—have their `margin-top` removed and a `margin-bottom: 1rem`. Nested lists have no `margin-bottom`. We've also reset the `padding-left` on `<ul>` and `<ol>` elements.

{{< markdown >}} * All lists have their top margin removed * And their bottom margin normalized * Nested lists have no bottom margin * This way they have a more even appearance * Particularly when followed by more list items * The left padding has also been reset

1. Here's an ordered list
2. With a few list items
3. It has the same overall look
4. As the previous unordered list {{< /markdown >}}

For simpler styling, clear hierarchy, and better spacing, description lists have updated `margins`. `<dd>`s reset `margin-left` to `0` and add `margin-bottom: .5rem`. `<dt>`s are **bolded**.

Description lists

A description list is perfect for defining terms.

Term

Definition for the term.

A second definition for the same term.

Another term

Definition for this other term.

## Inline code

Wrap inline snippets of code with `<code>`. Be sure to escape HTML angle brackets.

{{< example >}} For example, <section> should be wrapped as inline. {{< /example >}}

## Code blocks

Use `<pre>`s for multiple lines of code. Once again, be sure to escape any angle brackets in the code for proper rendering. The `<pre>` element is reset to remove its `margin-top` and use `rem` units for its `margin-bottom`.

{{< example >}}

{{< /example >}}

## Variables

For indicating variables use the `<var>` tag.

{{< example >}} y = mx + b {{< /example >}}

## User input

Use the `<kbd>` to indicate input that is typically entered via keyboard.

{{< example >}} To switch directories, type cd followed by the name of the directory. To edit settings, press ctrl + , {{< /example >}}

## Sample output

For indicating sample output from a program use the `<samp>` tag.

{{< example >}} This text is meant to be treated as sample output from a computer program. {{< /example >}}

## Tables

Tables are slightly adjusted to style `<caption>`s, collapse borders, and ensure consistent `text-align` throughout. Additional changes for borders, padding, and more come with [the `.table` class]({{< docsref "/content/tables" >}}).

This is an example table, and this is its caption to describe the contents.

Table heading

Table heading

Table heading

Table heading

Table cell

Table cell

Table cell

Table cell

Table cell

Table cell

Table cell

Table cell

Table cell

Table cell

Table cell

Table cell

## Forms

Various form elements have been rebooted for simpler base styles. Here are some of the most notable changes:

- `<fieldset>`s have no borders, padding, or margin so they can be easily used as wrappers for individual inputs or groups of inputs.
- `<legend>`s, like fieldsets, have also been restyled to be displayed as a heading of sorts.
- `<label>`s are set to `display: inline-block` to allow `margin` to be applied.
- `<input>`s, `<select>`s, `<textarea>`s, and `<button>`s are mostly addressed by Normalize, but Reboot removes their `margin` and sets `line-height: inherit`, too.

- `<textarea>`s are modified to only be resizable vertically as horizontal resizing often “breaks” page layout.
- `<button>`s and `<input>` button elements have `cursor: pointer` when `:not(:disabled)`.

These changes, and more, are demonstrated below.

Example legend

Example input

Example email

Example telephone

Example url

Example number

Example search

Example range

Example file input

Example select Choose… Option 1 Option 2 Option 3 Option 4 Option 5 Option 6

Check this checkbox

Option one is this and that Option two is something else that’s also super long to demonstrate the wrapping of these fancy form controls. Option three is disabled

Example textarea

Example date

Example time

Example password

Example datetime-local

Example week

Example month

Example color

Example output

100

Button submit

Button submit

{{< callout warning >}} {{< partial “callout-warning-input-support.md” >}} {{< /callout >}}

### Pointers on buttons

Reboot includes an enhancement for `role="button"` to change the default cursor to `pointer`. Add this attribute to elements to help indicate elements are interactive. This role isn't necessary for `<button>` elements, which get their own `cursor` change.

{{< example >}} Non-button element button {{< /example >}}

## Misc elements

### Address

The `<address>` element is updated to reset the browser default `font-style` from `italic` to `normal`. `line-height` is also now inherited, and `margin-bottom: 1rem` has been added. `<address>`s are for presenting contact information for the nearest ancestor (or an entire body of work). Preserve formatting by ending lines with `<br>`.

Twitter, Inc. 1355 Market St, Suite 900 San Francisco, CA 94103 P: (123) 456-7890

Full Name first.last@example.com

### Blockquote

The default `margin` on blockquotes is `1em 40px`, so we reset that to `0 0 1rem` for something more consistent with other elements.

A well-known quote, contained in a blockquote element.

Someone famous in Source Title

### Inline elements

The `<abbr>` element receives basic styling to make it stand out amongst paragraph text.

The HTML abbreviation element.

### Summary

The default `cursor` on summary is `text`, so we reset that to `pointer` to convey that the element can be interacted with by clicking on it.

Some details

More info about the details.

Even more details

Here are even more details about the details.

## HTML5 `[hidden]` attribute

HTML5 adds a new global attribute named `[hidden]`, which is styled as `display: none` by default. Borrowing an idea from PureCSS, we improve upon this default by making `[hidden] { display: none !important; }` to help prevent its `display` from getting accidentally overridden.

```
<input type="text" hidden>
```

{{< callout warning >}} ##### jQuery incompatibility

`[hidden]` is not compatible with jQuery's `$(...).hide()` and `$(...).show()` methods. Therefore, we don't currently especially endorse `[hidden]` over other techniques for managing the `display` of elements. {{< /callout >}}

To merely toggle the visibility of an element, meaning its `display` is not modified and the element can still affect the flow of the document, use [the `.invisible` class]({{< docsref "/utilities/visibility" >}}) instead.