

Politecnico
di Torino

Introduzione alle Applicazioni Web

CSS Bootstrap

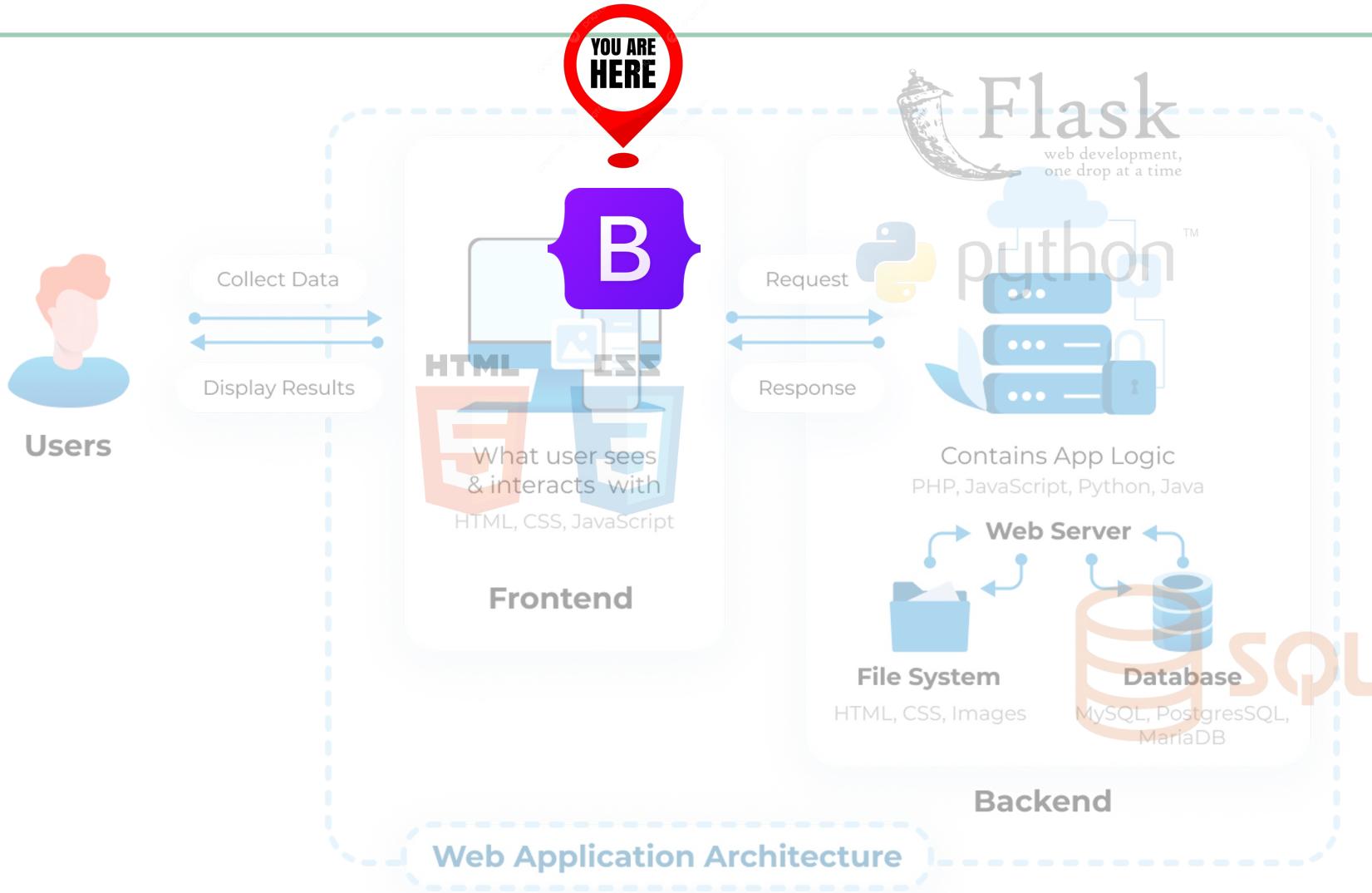
Juan Pablo Sáenz



Goals

- Understand what **Bootstrap** is and why it's useful
- Explore Bootstrap's **grid system**, **components**, and **utilities**
- Learn how to use Bootstrap for **responsive design**
- Practice using Bootstrap **classes**

📍 CSS Bootstrap: where are we?

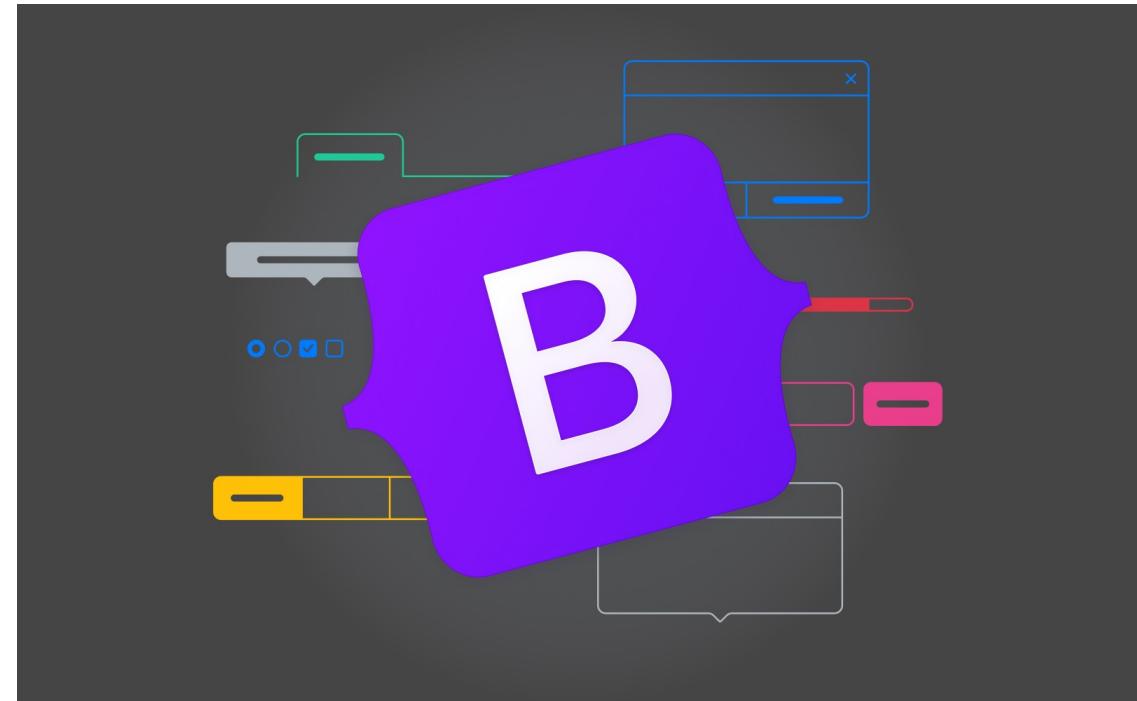


CSS Bootstrap

- A popular **front-end framework** for building responsive websites
 - A collection of **pre-styled components** and a **flexible grid system**
 - It helps create good-looking pages without writing all the CSS from scratch (**ready-to-use classes**)
-  Ensures **mobile-first**, consistent, and modern design
- Developed by **Twitter** in 2011 to standardize web design
 - Continues to evolve, with major updates improving flexibility and responsiveness

CSS Bootstrap: Features

- **Grid System:** A flexible layout structure to create responsive designs with rows and columns.
- **Predefined Components:** Ready-made UI elements like buttons, navigation bars, forms, and modals.
- **Utilities:** Small classes for common tasks like spacing, alignment, and visibility.
- **Icons**
- **Examples:** Ready-to-use templates and code snippets demonstrating the framework in action.



<https://webpixels.io/blog/8-reasons-to-use-bootstrap-or-a-css-framework>

CSS Bootstrap: Grid System

Uses a series of **containers**, **rows**, and **columns** to layout and align content.

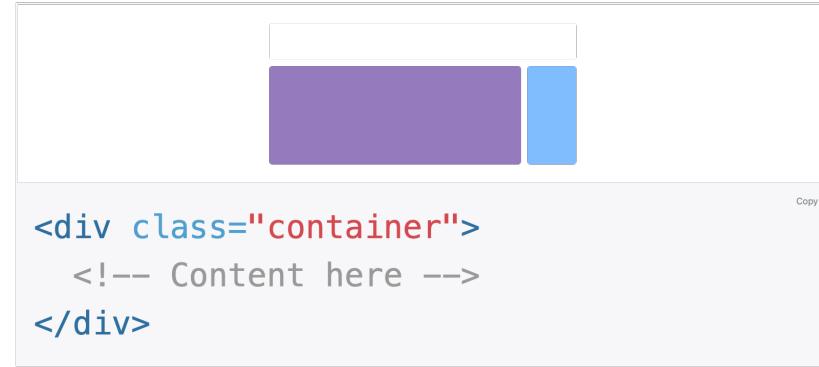
It's built with **flexbox** and is fully responsive

CSS Bootstrap: Grid System

Uses a series of **containers**, **rows**, and **columns** to layout and align content.

It's built with **flexbox** and is fully responsive

- **Containers**: the most basic layout element in Bootstrap and is **required** when using the default grid system



Use **.container-fluid** for a full width container, spanning the entire width of the viewport.



CSS Bootstrap: Grid System

Uses a series of **containers**, **rows**, and **columns** to layout and align content.

It's built with **flexbox** and is fully responsive

- **Containers**: the most basic layout element in Bootstrap and is **required** when using the default grid system
- **Rows are wrappers for columns**

Column	Column	Column
--------	--------	--------

HTML

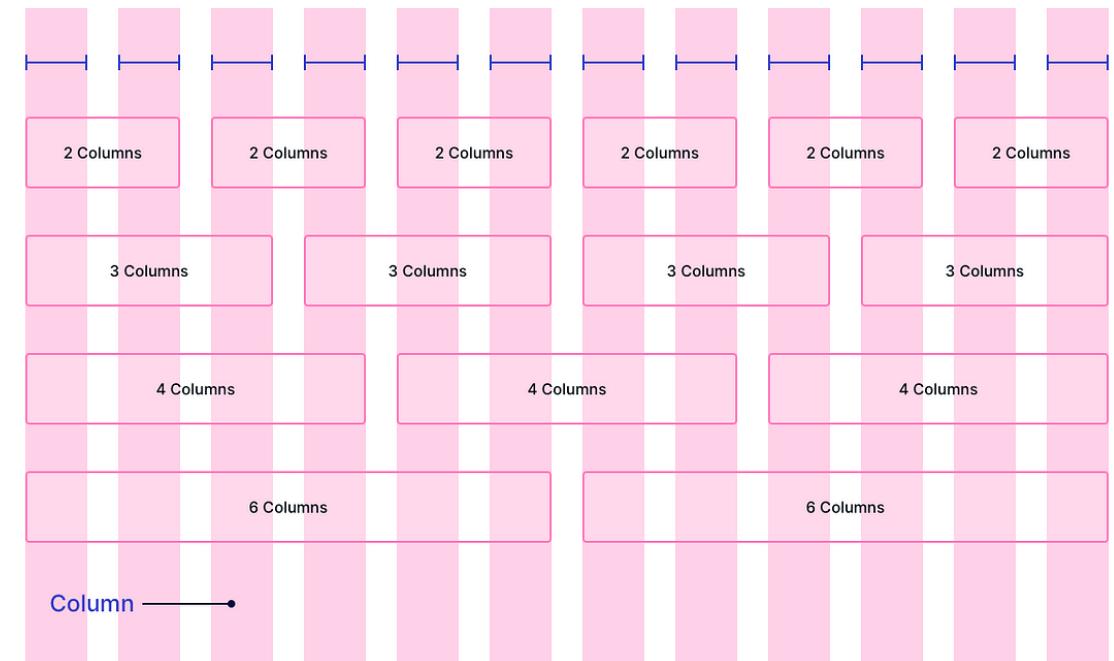
```
<div class="container text-center">
  <div class="row">
    <div class="col">
      Column
    </div>
    <div class="col">
      Column
    </div>
    <div class="col">
      Column
    </div>
  </div>
```

CSS Bootstrap: Grid System

Uses a series of **containers**, **rows**, and **columns** to layout and align content.

It's built with **flexbox** and is fully responsive

- **Containers:** the most basic layout element in Bootstrap and is **required** when using the default grid system
- **Rows are wrappers for columns**
- **There are 12 template columns available per row**



https://miro.medium.com/v2/resize:fit:1200/1*hw8EKWA-DNytrtbzgZ_dEw.png

CSS Bootstrap: Grid System

Uses a series of **containers**, **rows**, and **columns** to layout and align content.

It's built with **flexbox** and is fully responsive

- **Containers**: the most basic layout element in Bootstrap and is **required** when using the default grid system
- Rows are wrappers for columns
- There are **12 template columns** available **per row**

	xs <576px	sm ≥576px	md ≥768px	lg ≥992px	xl ≥1200px	xxl ≥1400px
Container max-width	None (auto)	540px	720px	960px	1140px	1320px
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-	.col-xxl-
# of columns	12					
Gutter width	1.5rem (.75rem on left and right)					
Custom gutters	Yes					
Nestable	Yes					
Column ordering	Yes					

```
<div class="container">
  <div class="row">
    <div class="col-8">col-8</div>
    <div class="col-4">col-4</div>
  </div>
</div>
```

CSS Bootstrap: Grid System

Uses a series of **containers**, **rows**, and **columns** to layout and align content.

It's built with **flexbox** and is fully responsive

- **Containers**: the most basic layout element in Bootstrap and is **required** when using the default grid system
- **Rows** are wrappers for columns
- There are **12 template columns** available per row
- Includes **six default breakpoints** for building responsively

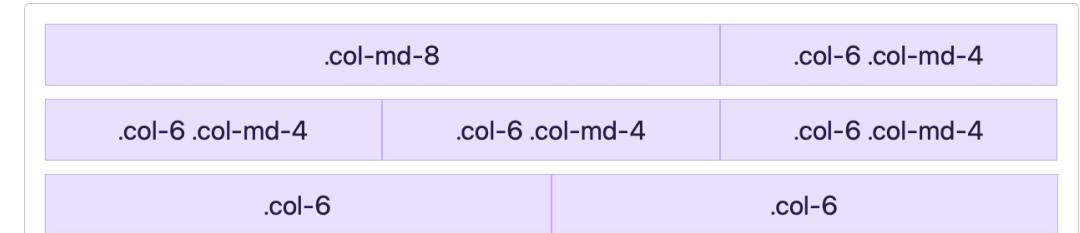
Breakpoint	Class infix	Dimensions
Extra small	<code>None</code>	<576px
Small	<code>sm</code>	≥576px
Medium	<code>md</code>	≥768px
Large	<code>lg</code>	≥992px
Extra large	<code>xl</code>	≥1200px
Extra extra large	<code>xxl</code>	≥1400px

CSS Bootstrap: Grid System

Uses a series of **containers**, **rows**, and **columns** to layout and align content.

It's built with **flexbox** and is fully responsive

- **Containers:** the most basic layout element in Bootstrap and is **required** when using the default grid system
- **Rows are wrappers for columns**
- There are **12 template columns** available per row
- Includes **six default breakpoints** for building responsively



The diagram illustrates the CSS Bootstrap Grid System with three rows of columns. Row 1 contains one column with class .col-md-8 and another with class .col-6 .col-md-4. Row 2 contains three columns, each with class .col-6 .col-md-4. Row 3 contains two columns, each with class .col-6.

HTML

```
<div class="container text-center">
  <!-- Stack the columns on mobile by making one full-width and the other
       two half-width -->
  <div class="row">
    <div class="col-md-8">.col-md-8</div>
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
  </div>

  <!-- Columns start at 50% wide on mobile and bump up to 33.3% wide on
       desktop -->
  <div class="row">
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
  </div>

  <!-- Columns are always 50% wide, on mobile and desktop -->
  <div class="row">
    <div class="col-6">.col-6</div>
    <div class="col-6">.col-6</div>
  </div>
</div>
```

CSS Bootstrap: Predefined Components

Ready-made UI elements like buttons, navigation bars, forms, and modals

- <https://getbootstrap.com/docs/5.3>

The screenshot shows a web browser displaying the Bootstrap documentation at <https://getbootstrap.com/docs/5.3>. The page is titled "Accordion" under the "Components" section. The left sidebar lists various UI components, including "Forms" (Overview, Form control, Select, Checks & radios, Range, Input group, Floating labels, Layout, Validation) and "Components" (Accordion, Alerts, Badge, Breadcrumb, Buttons, Button group, Card, Carousel, Close button, Collapse, Dropdowns, List group, Modal, Navbar, Navs & tabs). The main content area shows an accordion component with three items: "Accordion Item #1", "Accordion Item #2", and "Accordion Item #3". A note explains that the first item is expanded by default. Below the accordion, there is an "HTML" section containing the source code for the accordion example:

```
<div class="accordion" id="accordionExample">
  <div class="accordion-item">
    <h2 class="accordion-header">
      <button class="accordion-button" type="button" data-bs-toggle="collapse" data-bs-target="#collapseOne">
        Accordion Item #1
      </button>
    </h2>
    <div id="collapseOne" class="accordion-collapse collapse show" data-bs-parent="#accordionExample">
      <div class="accordion-body">
        <strong>This is the first item's accordion body.</strong> It is shown by default, until the collapse plugin adds the appropriate classes that we use to style each element. These classes control the overall appearance, as well as the showing and hiding via CSS transitions. You can modify any of this with custom CSS or overriding our default variables. It's also worth noting that just about any HTML<br>can go within the .accordion-body, though the transition does limit overflow.
    </div>
  </div>
  <div class="accordion-item">
    <h2 class="accordion-header">
      <button class="accordion-button" type="button" data-bs-toggle="collapse" data-bs-target="#collapseTwo">
        Accordion Item #2
      </button>
    </h2>
    <div id="collapseTwo" class="accordion-collapse collapse" data-bs-parent="#accordionExample">
      <div class="accordion-body">
        ...
      </div>
    </div>
  </div>
  <div class="accordion-item">
    <h2 class="accordion-header">
      <button class="accordion-button" type="button" data-bs-toggle="collapse" data-bs-target="#collapseThree">
        Accordion Item #3
      </button>
    </h2>
    <div id="collapseThree" class="accordion-collapse collapse" data-bs-parent="#accordionExample">
      <div class="accordion-body">
        ...
      </div>
    </div>
  </div>
</div>
```

CSS Bootstrap: Examples

Ready-made UI elements like buttons, navigation bars, forms, and modals

- <https://getbootstrap.com/docs/5.3/examples/>

The screenshot shows the Bootstrap Examples page at getbootstrap.com, specifically the 'Snippets' section. The page has a purple header with the Bootstrap logo, 'Docs', 'Examples', 'Icons', 'Themes', and 'Blog' links, a search bar, and social media icons. Below the header, the title 'Snippets' is displayed, followed by a subtitle: 'Common patterns for building sites and apps that build on existing components and utilities with custom CSS and more.' The page is divided into four main sections: 'Headers', 'Heroes', 'Features', and 'Sidebars' on the top row, and 'Footers', 'Dropdowns', 'List groups', and 'Modals' on the bottom row. Each section contains a small screenshot of a web interface and a brief description.

- Headers**
Display your branding, navigation, search, and more with these header components.
- Heroes**
Set the stage on your homepage with heroes that feature clear calls to action.
- Features**
Explain the features, benefits, or other details in your marketing content.
- Sidebars**
Common navigation patterns ideal for offcanvas or multi-column layouts.
- Footers**
Finish every page strong with an awesome footer, big or small.
- Dropdowns**
Enhance your dropdowns with filters, icons, custom styles, and more.
- List groups**
Extend list groups with utilities and custom styles for any content.
- Modals**
Transform modals to serve any purpose, from feature tours to dialogs.

CSS Bootstrap: Utilities

Small classes for **common tasks** like spacing, alignment, and visibility

- <https://getbootstrap.com/docs/5.3>

The screenshot shows a web browser displaying the Bootstrap 5.3 documentation at getbootstrap.com/docs/5.3. The page is titled "Background". The left sidebar has a "Background" section selected, showing utility classes for vertical rules, visually hidden content, and various background properties like borders, colors, and shadows. The main content area discusses background utilities, noting they don't set color by default but can be used with color utilities. It shows examples of primary, secondary, success, and danger background colors with their corresponding subtle variations.

Vertical rule
Visually hidden

(*) Utilities
API
Background

- Borders
- Colors
- Display
- Flex
- Float
- Interactions
- Link
- Object fit
- Opacity
- Overflow
- Position
- Shadows
- Sizing
- Spacing
- Text
- Vertical align
- Visibility
- Z-index

Extend

- Approach
- Icons

alternative means, such as additional text hidden with the `.visually-hidden` class.

Background color

Similar to the contextual text color classes, set the background of an element to any contextual class. Background utilities **do not set color**, so in some cases you'll want to use `.text-* color utilities`.

Background utilities like `.bg-*` that generated from our original `$theme-colors` Sass map don't yet respond to color modes, however, any `.bg-*subtle` utility will. This will be resolved in v6.

<code>.bg-primary</code>
<code>.bg-primary-subtle</code>
<code>.bg-secondary</code>
<code>.bg-secondary-subtle</code>
<code>.bg-success</code>
<code>.bg-success-subtle</code>
<code>.bg-danger</code>
<code>.bg-danger-subtle</code>

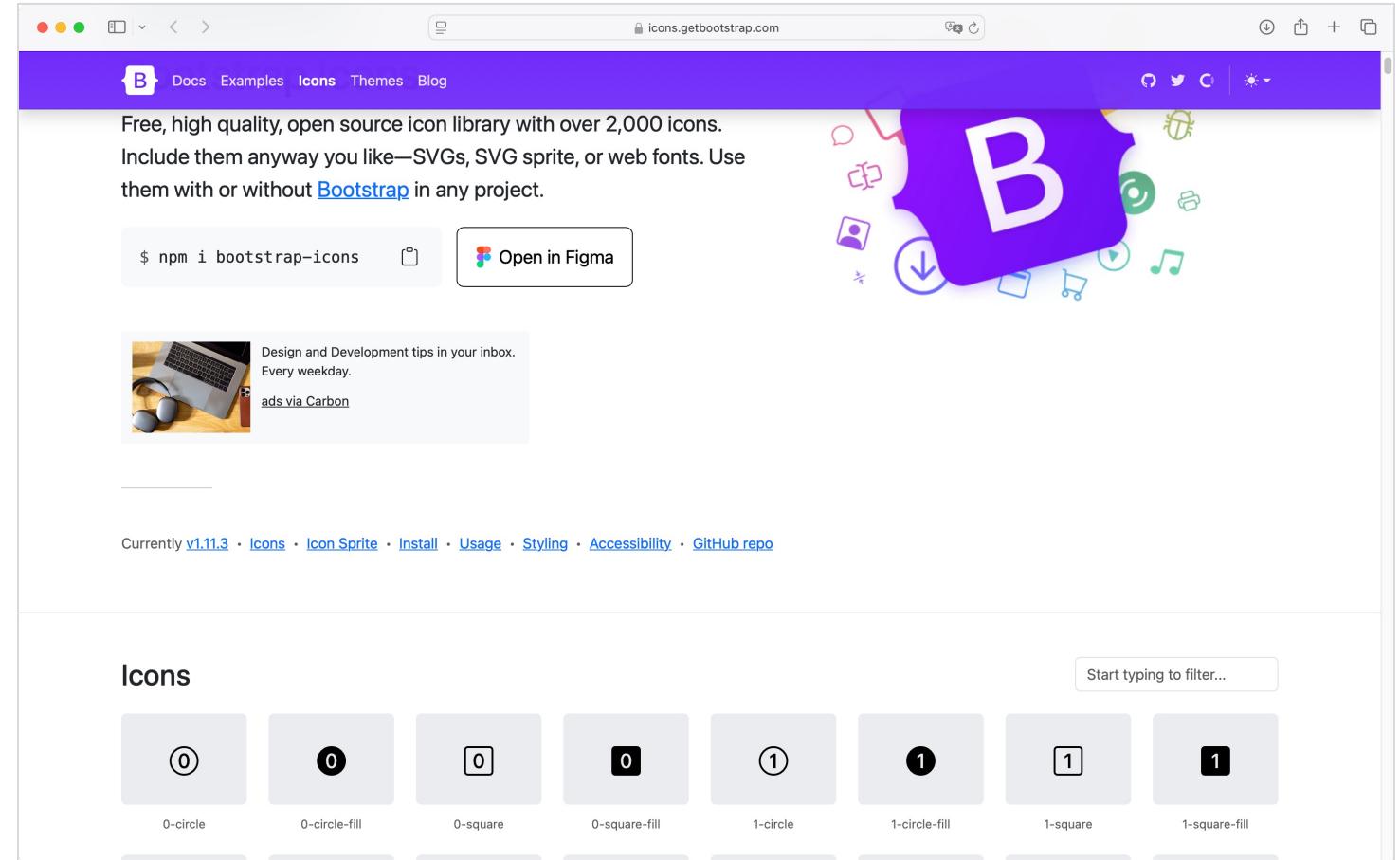
On this page

- Background color
- Background gradient
- Opacity
- How it works
- Example
- CSS
- Sass variables
- Sass maps
- Sass mixins
- Sass utilities API

CSS Bootstrap: Icons

Free, high-quality, open-source **icon library**

- <https://icons.getbootstrap.com>



Getting Started with Bootstrap

Step 1:

Add the **Bootstrap CSS** and **JavaScript links** to the HTML `<head>` section

Just as we have done with our local CSS file

The **links** are available at:

- <https://getbootstrap.com>

```
<head>
  <meta charset="utf-8">
  <title>Bootstrap demo</title>
  <link href="styles.css" rel="stylesheet">
  <link
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
    rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2
    bRjXh0JMhjY6hW+ALEwIH"
    crossorigin="anonymous">
</head>
```

Getting Started with Bootstrap

Step 2:

Add the responsive **viewport** meta tag to the `<head>` section

- **width=device-width**:

Sets the width of the viewport to match the device's screen width (so it adapts to any screen size)

- **initial-scale=1**:

Ensures that the page is displayed at a normal scale (not zoomed in or out) when first loaded

```
<head>
  <meta charset="utf-8">
  <meta name="viewport"
        content="width=device-width,
        initial-scale=1">
  <title>Bootstrap demo</title>
</head>
```

Getting Started with Bootstrap

Step 3:

⚠ Start with a `<div class="container">` to use the grid system

Step 4:

Use Bootstrap classes

(<https://getbootstrap.com/docs/5.3>)

Add predefined classes for styling and layout

🎨 It can be **fully customized** by adding our own CSS styles, allowing you to **override** or **extend** its default design

```
<body>
  <div class="container">
    <div class="row">
      <div class="col-8">col-8</div>
      <div class="col-4">col-4</div>
    </div>
  </div>
</body>
```

Let's see it in practice



Licenza

- These slides are distributed under a Creative Commons license "**Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)**"
- **You are free to:**
 - **Share** – copy and redistribute the material in any medium or format
 - **Adapt** – remix, transform, and build upon the material
 - The licensor cannot revoke these freedoms as long as you follow the license terms.
- **Under the following terms:**
 - **Attribution** – You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
 - **NonCommercial** – You may not use the material for commercial purposes.
 - **ShareAlike** – If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
 - **No additional restrictions** – You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.
- <https://creativecommons.org/licenses/by-nc-sa/4.0/>