

4: Advanced Bootstrap

Bootstrap Navigation: Navbar, Navs, and Breadcrumbs.

Navbar:

A navigation bar is a navigation header that is placed at the top of the page:

Basic Navbar:

With Bootstrap, a navigation bar can extend or collapse, depending on the screen size.

A standard navigation bar is created with the `.navbar` class, followed by a responsive collapsing class: `.navbar-expand-xxl|xl|lg|md|sm` (stacks the navbar vertically on xxlarge, extra large, large, medium or small screens).

To add links inside the navbar, use either an `` element (or a `<div>`) with `class="navbar-nav"`. Then add `` elements with a `.nav-item` class followed by an `<a>` element with a `.nav-link` class:

Vertical Navbar:

Remove the `.navbar-expand-*` class to create a navigation bar that will always be vertical:

Link 1
Link 2
Link 3

Centered Navbar

Add the `.justify-content-center` class to center the navigation bar:

Link 1 Link 2 Link 3

Colored Navbar

Use any of the `.bg-color` classes to change the background color of the navbar (`.bg-primary`, `.bg-success`, `.bg-info`, `.bg-warning`, `.bg-danger`, `.bg-secondary`, `.bg-dark` and `.bg-light`)

Tip: Add a white text color to all links in the navbar with the `.navbar-dark` class, or use the `.navbar-light` class to add a black text color.

Active/disabled state:

Add the `.active` class to an `<a>` element to highlight the current link, or the `.disabled` class to indicate that the link is un-clickable.

Brand / Logo

The `.navbar-brand` class is used to highlight the brand/logo/project name of your page:

When using the `.navbar-brand` class with images, Bootstrap 5 will automatically style the image to fit the navbar vertically.

Navbar Text

When using the `.navbar-brand` class with images, Bootstrap 5 will automatically style the image to fit the navbar vertically.

Navbar Toggler Button:

Very often, especially on small screens, you want to hide the navigation links and replace them with a button that should reveal them when clicked on.

To create a collapsible navigation bar, use a button with `class="navbar-toggler"`, `data-bs-toggle="collapse"` and `data-bs-target="#thetarget"`. Then wrap the navbar content (links, etc) inside a `<div>` element with `class="collapse navbar-collapse"`, followed by an id that matches the `data-bs-target` of the button: `"thetarget"`.

Navs:

If you want to create a simple horizontal menu, add the `.nav` class to a `` element, followed by `.nav-item` for each `` and add the `.nav-link` class to their links:

Aligned Nav:

Add the `.justify-content-center` class to center the nav, and the `.justify-content-end` class to right-align the nav.

Vertical Nav:

Add the `.flex-column` class to create a vertical nav:

Tabs:

Turn the nav menu into navigation tabs with the `.nav-tabs` class. Add the `.active` class to the active/current link.

Pills:

Turn the nav menu into navigation pills with the `.nav-pills` class.:

Justified Tabs/pills:

Justify the tabs/pills with the `.nav-justified` class (equal width):

Toggleable / Dynamic Tabs:

To make the tabs toggleable, add the `data-toggle="tab"` attribute to each link. Then add a `.tab-pane` class with a unique ID for every tab and wrap them inside a `<div>` element with class `.tab-content`.

If you want the tabs to fade in and out when clicking on them, add the `.fade` class to `.tab-pane`:

The same code applies to pills; only change the `data-toggle` attribute to `data-bs-toggle="pill"`:

BreadCrumbs:

Indicate the current page's location within a navigational hierarchy that automatically adds separators via CSS. Use an ordered or unordered list with linked list items to create a minimally styled breadcrumb.

Dividers:

Dividers are automatically added in CSS through `::before` and `content`. They can be changed by modifying a local CSS custom property `--bs-breadcrumb-divider`.

Bootstrap Responsive Utilities: hide, show, and responsive breakpoints.

Display Utilities (d-*):

- `d-none`: Hide the element on all screen sizes.
- `d-block`: Display the element as a block on all screen sizes.
- `d-inline`: Display the element as an inline element on all screen sizes.
- `d-inline-block`: Display the element as an inline block element on all screen sizes.
- `d-flex`: Display the element as a flex container on all screen sizes.
- `d-inline-flex`: Display the element as an inline flex container on all screen sizes.
- `d-table`: Display the element as a table on all screen sizes.
- `d-table-row`: Display the element as a table row on all screen sizes.
- `d-table-cell`: Display the element as a table cell on all screen sizes.
- `d-grid`: Display the element as a grid container on all screen sizes.
- `d-inline-grid`: Display the element as an inline grid container on all screen sizes.

Hidden/Visible Utilities:

- `d-*-hidden`: Hide the element on a specific screen size (replace `*` with the desired breakpoint: `xs`, `sm`, `md`, `lg`, or `xl`).
- `visible` and `invisible`: These classes are used to control the visibility of elements without affecting layout. `visible` makes the element visible and `invisible` hides it, but it still occupies space in the layout.

Bootstrap utilizes a grid system that is divided into 12 columns across the viewport. The responsive design is facilitated by defining breakpoints at which the layout can be adjusted to fit different screen sizes. Bootstrap includes five responsive breakpoints:

Extra small (xs): This is the default breakpoint and applies to screens smaller than 576 pixels wide. It's typically used for mobile devices.

Small (sm): Applies to screens equal to or greater than 576 pixels wide. It's commonly used for small tablets and larger mobile devices.

Medium (md): Applies to screens equal to or greater than 768 pixels wide. It's often used for tablets and smaller desktop screens.

Large (lg): Applies to screens equal to or greater than 992 pixels wide. It's typically used for larger desktop screens.

Extra large (xl): Applies to screens equal to or greater than 1200 pixels wide. It's commonly used for larger desktop screens and some extra-large displays.

XX large (xxl): Applies to screens equal to or greater than 1400 pixels wide. It's commonly used for larger desktop screens and some extra-large displays.

Bootstrap allows you to define classes specific to these breakpoints. For example:

- `col-6 col-sm-4 col-md-3 col-lg-2`: This column will take up 6 columns on extra small screens, 4 columns on small screens, 3 columns on medium screens, and 2 columns on large screens.
- `d-none d-md-block`: This element will be hidden on screens smaller than medium size.
- `d-lg-none`: This element will be hidden on screens larger than or equal to large size.

These classes enable you to create responsive designs that adapt gracefully to different screen sizes, ensuring a consistent user experience across various devices.

Bootstrap JavaScript components: Modals, Carousels, and Tooltips.

Modals

The Modal component is a dialog box/popup window that is displayed on top of the current page:

Add animation

Use the `.fade` class to add a fading effect when opening and closing the modal:

Modal Size

- Change the size of the modal by adding the `.modal-sm` class for small modals (max-width 300px), `.modal-lg` class for large modals (max-width 800px), or `.modal-xl` for extra large modals (max-width 1140px). Default is 500px max-width.
- Add the size class to the `<div>` element with class `.modal-dialog`:

Fullscreen Modals

If you want the modal to span the whole width and height of the page, use the `.modal-fullscreen` class:

Centered Modal

Center the modal vertically and horizontally within the page, with the `.modal-dialog-centered` class:

Scrolling Modal

- When you have a lot of content inside the modal, a scrollbar is added to the page.
- However, it is possible to only scroll inside the modal, instead of the page itself, by adding `.modal-dialog-scrollable` to `.modal-dialog`:

Carousel

The Carousel is a slideshow for cycling through elements:

.carousel Creates a carousel

.carousel-indicators Adds indicators for the carousel. These are the little dots at the bottom of each slide (which indicates how many slides there are in the carousel, and which slide the user are currently viewing)

.carousel-inner Adds slides to the carousel

.carousel-item Specifies the content of each slide

.carousel-control-prev Adds a left (previous) button to the carousel, which allows the user to go back between the slides

.carousel-control-next Adds a right (next) button to the carousel, which allows the user to go forward between the slides

.carousel-control-prev-icon Used together with `.carousel-control-prev` to create a "previous" button

.carousel-control-next-icon Used together with `.carousel-control-next` to create a "next" button

.slide Adds a CSS transition and animation effect when sliding from one item to the next. Remove this class if you do not want this effect.

Add elements inside `<div class="carousel-caption">` within each `<div class="carousel-item">` to create a caption for each slide:

Tooltips

The Tooltip component is small pop-up box that appears when the user moves the mouse pointer over an element:

How To Create a Tooltip

To create a tooltip, add the `data-bs-toggle="tooltip"` attribute to an element.

Use the `title` attribute to specify the text that should be displayed inside the tooltip:

Note: Tooltips must be initialized with JavaScript to work.

Positioning Tooltips

By default, the tooltip will appear on top of the element.

Use the `data-bs-placement` attribute to set the position of the tooltip on top, bottom, left or the right side of the element: