Deloitte.

Bootstrap

Deloitte Technology Academy (DTA)

Agenda

</>

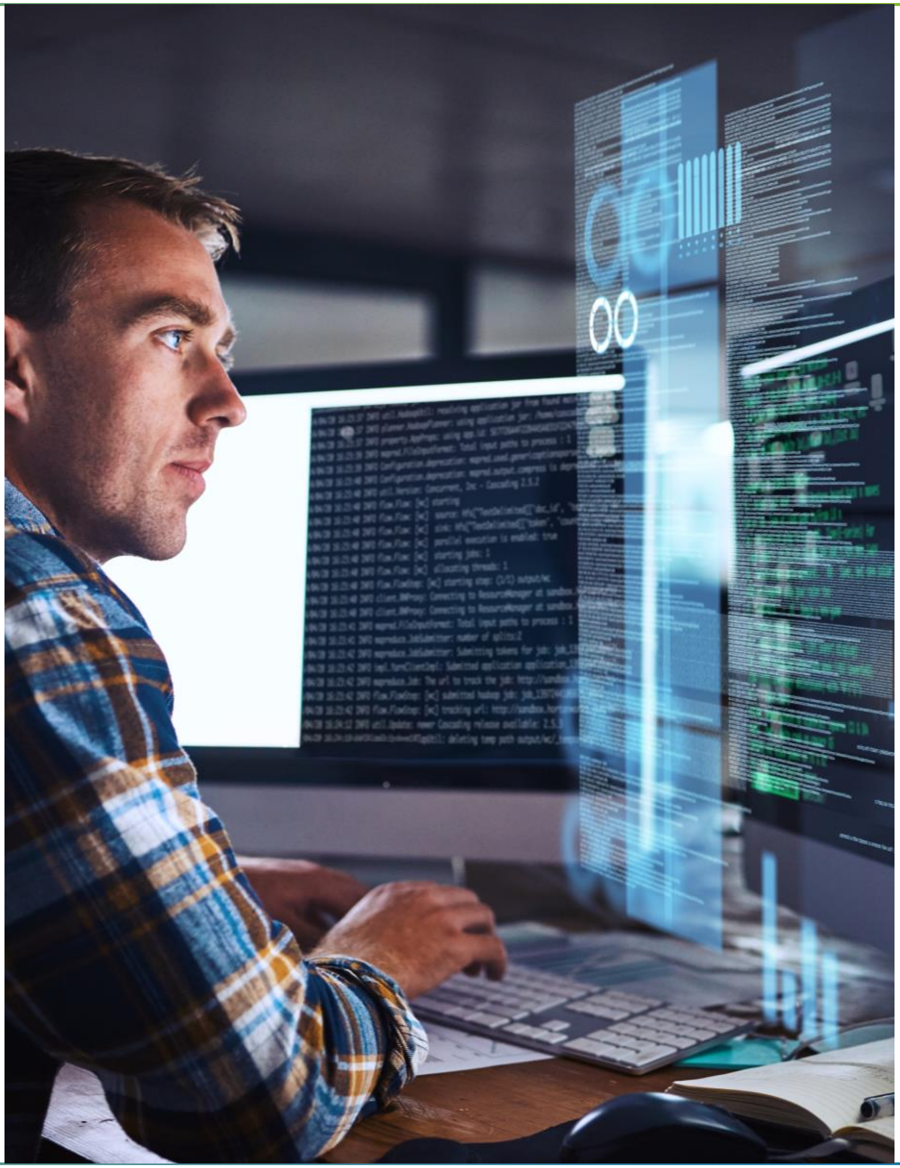
Topics	Descriptions	Duration
Overview	Overview, Getting Started, Download Content, and Base Template	6 hours
Layout	Containers, Grid System, Columns, and Gutters	
Content	Colors, Typography, Images, and Tables	
Forms	Form Controls, Checks and Radios, Input Groups, Floating Labels, and Form Validation	
Components and Utilities	Buttons, Alerts, Progress Bar, Pagination, Navbar, Modal, Helpers, Utilities, and Vector Icons	

Learning Objectives

By the end of this session, you will be able to:

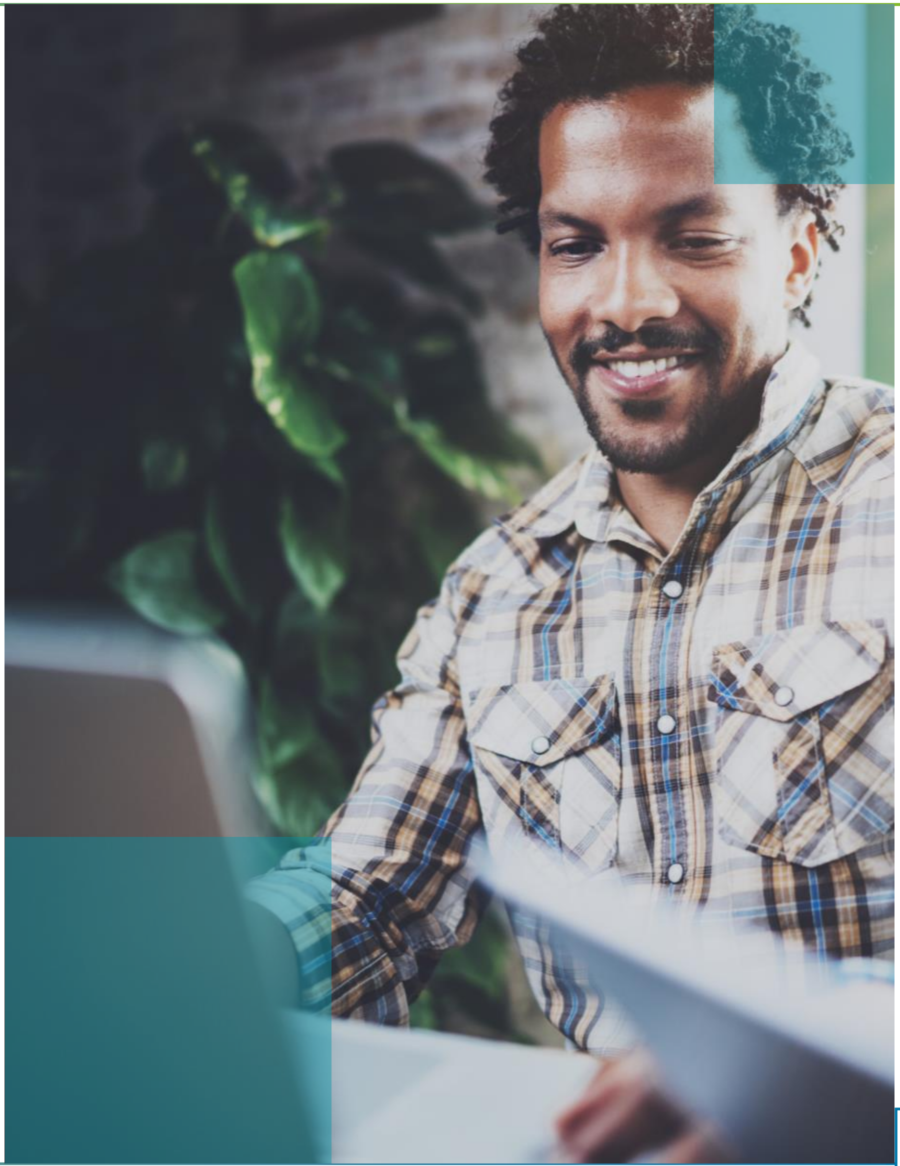
- Describe the basics of Bootstrap, including its history
- Explain Bootstrap classes for form controls
- Describe custom controls, such as input groups and floating labels
- Explain the usage of Bootstrap form validation
- Explain Bootstrap helper and utility classes
- Describe Bootstrap icons and their usage

Overview



Bootstrap

- Features of Bootstrap :
  - Is a free, open-source Cascading Style Sheets (CSS) framework
  - Provides rich, responsive, mobile-first, front-end web development options
  - Is amongst the most popular front-end framework for faster and easier web development
  - Gives the ability to easily create responsive designs
  - Enables developers to take advantage of CSS classes defined in Bootstrap to further customize the appearance of their content



Overview of Bootstrap

- Includes Hypertext Markup Language (HTML) and CSS-based design templates for forms, tables, buttons, typography, modals, navigation, image carousels and many other, as well as optional JavaScript plug-ins.
- Integrally provides responsive web design, which is used to target sizes of multiple User Interfaces (UI)—for desktops, tablets, and mobile devices.
- Several JavaScript components are included that do not require other libraries. They provide additional user interface elements such as tooltips, dialog boxes, navigation drop-down boxes, progress bars, and carousels.

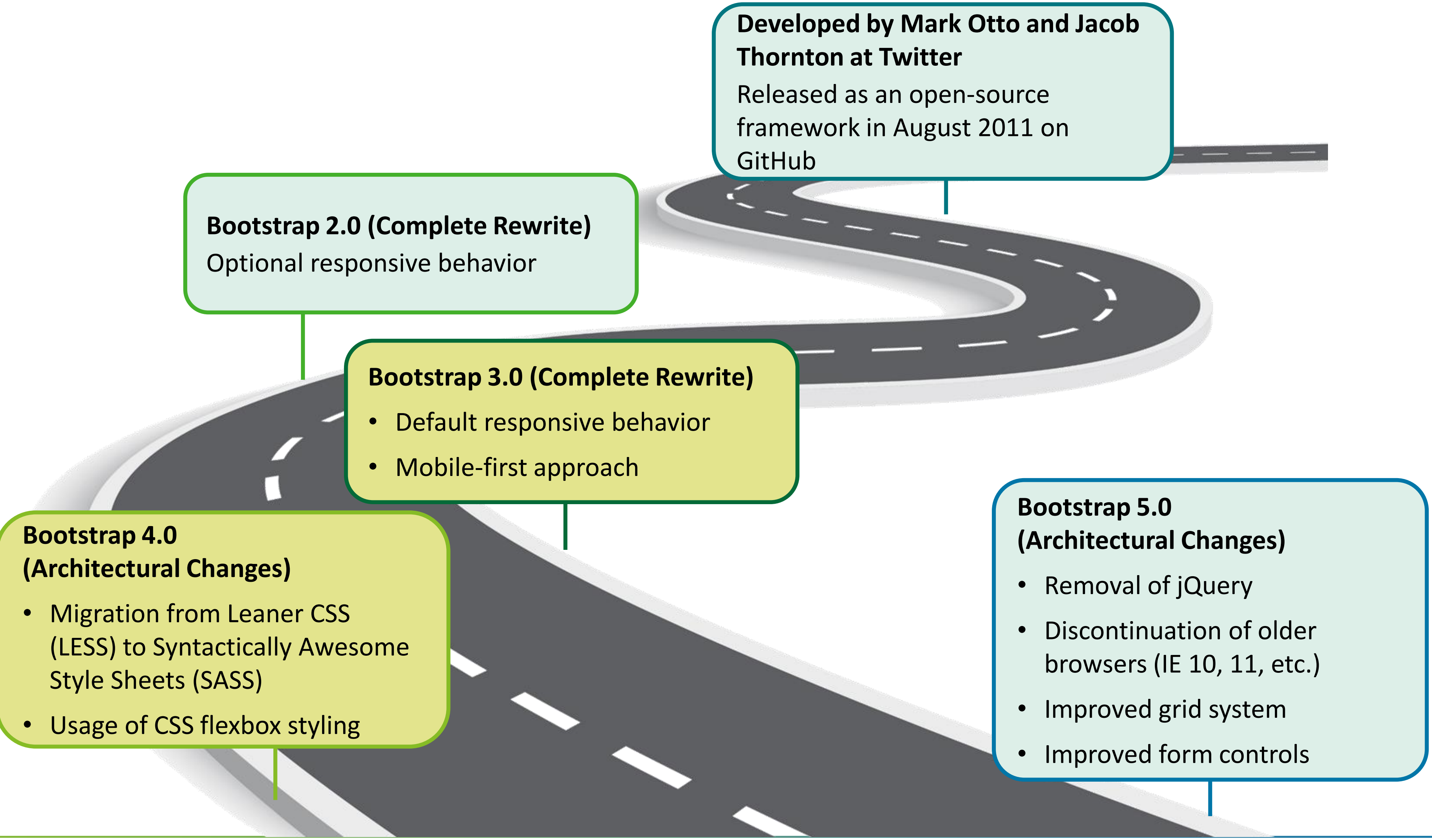
Advantages of Using Bootstrap

- Browser compatibility
- Faster development
- Responsive layout
- Consistent design
- Easy-to-use features

Disadvantages of Using Bootstrap

- Same look and feel across all the websites might give it a monotonous appearance
- No backward compatibility between different versions

History



Developed by Mark Otto and Jacob Thornton at Twitter

Released as an open-source framework in August 2011 on GitHub

Bootstrap 2.0 (Complete Rewrite)

Optional responsive behavior

Bootstrap 3.0 (Complete Rewrite)

- Default responsive behavior
- Mobile-first approach

Bootstrap 4.0 (Architectural Changes)

- Migration from Leaner CSS (LESS) to Syntactically Awesome Style Sheets (SASS)
- Usage of CSS flexbox styling

Bootstrap 5.0 (Architectural Changes)

- Removal of jQuery
- Discontinuation of older browsers (IE 10, 11, etc.)
- Improved grid system
- Improved form controls

CSS Preprocessor

- Enables you to generate CSS from the preprocessor’s own proprietary language/syntax
- Provides some features that do not exist in plain vanilla CSS
- Makes CSS structure more readable and maintainable, when used
- Includes SASS, LESS, Stylus, and PostCSS among CSS preprocessors

Getting Started

Primarily, there are three options to reference Bootstrap in webpages.

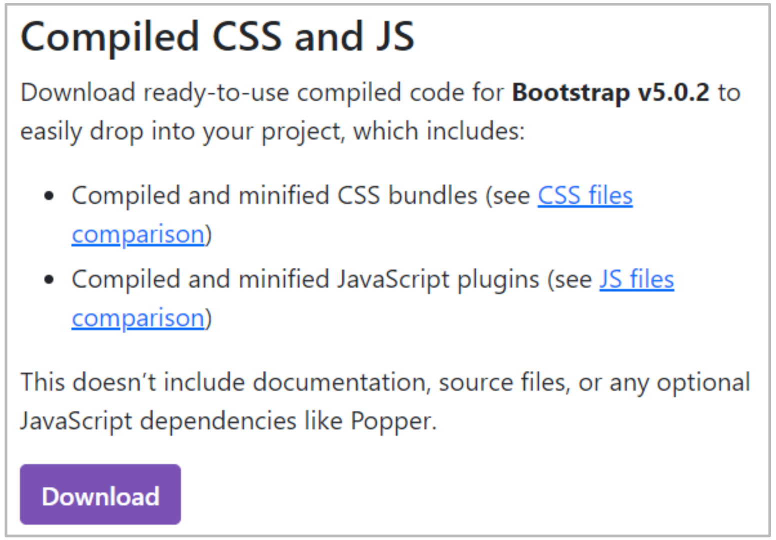
- **Load Bootstrap via CDN (content delivery network)**
  - To get Bootstrap in a web page, we can use the following code in the head section :

```
<link rel="stylesheet“
href="https://cdn.cdnName.net/npm/bootstrap@5.0.2/dist/css/
bootstrap.min.css">
```

- **Host Bootstrap Locally**
  - Download the reference files locally from the project path
  - Use the recommended approach for production applications as below:

```
<link rel="stylesheet"
href="bootstrap/css/bootstrap.min.css">
```

- **Package Managers**
  - Install with npm (`npm install bootstrap`)
  - Install with yarm (`yarn add bootstrap`)
  - Install with NuGet (`Install-Package bootstrap`)

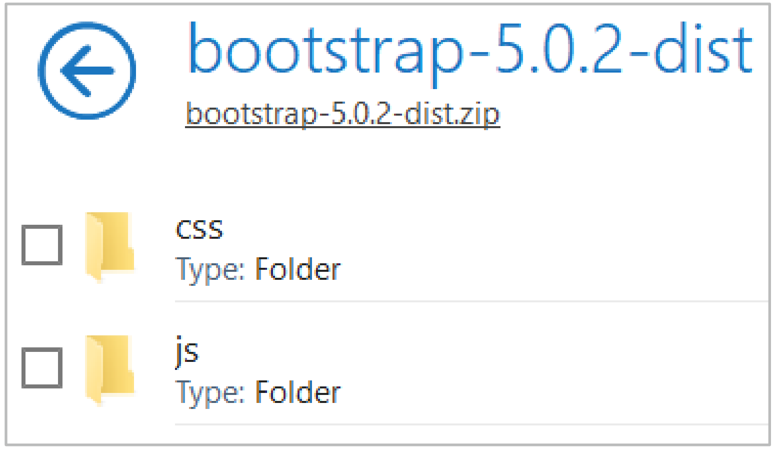


Compiled CSS and JS

Download ready-to-use compiled code for **Bootstrap v5.0.2** to easily drop into your project, which includes:

- Compiled and minified CSS bundles (see CSS files comparison)
- Compiled and minified JavaScript plugins (see JS files comparison)

This doesn't include documentation, source files, or any optional JavaScript dependencies like Popper.



bootstrap-5.0.2-dist

bootstrap-5.0.2-dist.zip

□ css
Type: Folder

□ js
Type: Folder

Download—Contents

Download includes CSS and JavaScript folders

CSS Files

Extensions	Descriptions
`*.css`	Normal CSS files
`*.rtl.css`	Right-to-left CSS files
`*.min.css`	Minified CSS files
`*.map`	Source map files

```
bootstrap/
├── css/
│   ├── bootstrap-grid.css
│   ├── bootstrap-grid.css.map
│   ├── bootstrap-grid.min.css
│   ├── bootstrap-grid.min.css.map
│   ├── bootstrap-grid.rtl.css
│   ├── bootstrap-grid.rtl.css.map
│   ├── bootstrap-grid.rtl.min.css
│   ├── bootstrap-grid.rtl.min.css.map
│   ├── bootstrap-reboot.css
│   ├── bootstrap-reboot.css.map
│   ├── bootstrap-reboot.min.css
│   ├── bootstrap-reboot.min.css.map
│   ├── bootstrap-reboot.rtl.css
│   ├── bootstrap-reboot.rtl.css.map
│   ├── bootstrap-reboot.rtl.min.css
│   ├── bootstrap-reboot.rtl.min.css.map
│   ├── bootstrap-utilities.css
│   ├── bootstrap-utilities.css.map
│   ├── bootstrap-utilities.min.css
│   ├── bootstrap-utilities.min.css.map
│   ├── bootstrap-utilities.rtl.css
│   ├── bootstrap-utilities.rtl.css.map
│   ├── bootstrap-utilities.rtl.min.css
│   ├── bootstrap-utilities.rtl.min.css.map
│   ├── bootstrap.css
│   ├── bootstrap.css.map
│   ├── bootstrap.min.css
│   ├── bootstrap.min.css.map
│   ├── bootstrap.rtl.css
│   ├── bootstrap.rtl.css.map
│   ├── bootstrap.rtl.min.css
│   └── bootstrap.rtl.min.css.map
│
```

Download—Contents (Cont.)

JavaScript Files

Extension	Description
*.js	Normal JavaScript files
*.min.js	Minified JavaScript files
*.map	Source map files

```
bootstrap/
└── js/
    ├── bootstrap.bundle.js
    ├── bootstrap.bundle.js.map
    ├── bootstrap.bundle.min.js
    ├── bootstrap.bundle.min.js.map
    ├── bootstrap.esm.js
    ├── bootstrap.esm.js.map
    ├── bootstrap.esm.min.js
    ├── bootstrap.esm.min.js.map
    ├── bootstrap.js
    ├── bootstrap.js.map
    ├── bootstrap.min.js
    └── bootstrap.min.js.map
```

Base Template

- Mobile first—the code is optimized for mobile devices first, and then components are scaled up as necessary by using the CSS media queries.
- Responsive viewport meta tag is added to ensure proper rendering and touch zooming for all devices.
- CSS reference provides all the necessary styling classes for Bootstrap to function on the webpage.
- Other JavaScript references are optional and can be included on need basis.

```
<!doctype html>
<html lang="en">
 <head>
 <!-- Required meta tags -->
 <meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<!-- Bootstrap CSS -->
 <link
href="https://cdnName/sampleDeloitteCSSpath/bootstrap.min.css"
   <title>Hello, world!</title>
 </head>
 <body>
<hl>Hello, world!</hl>
<!-- Optional JS: Bootstrap Bundle with Popper -->
 <script
 src="https://cdnName/sampleJSpath/bootstrap.bundle.min.js
</body>
</html>
```

</>

Layout

Containers

- Are the most basic layout elements in Bootstrap for the default grid system
- Can be nested; however, generally it’s not needed while creating layout
- Are used to pad content inside them:
  - The `.container` class provides a responsive fixed-width container.
  - The class infix can be used with the `.container` class.
  - The `.container-fluid` class provides a full-width container, spanning the entire width of the viewport.

```
<div class="container">
  <!-- Content here -->
</div>
```

```
<div class="container-fluid">
  <!-- Content here -->
</div>
```

Breakpoint	Extra Small	Small	Medium	Large	Extra Large	Extra Extra Large
Class Infix	none	sm	md	lg	xl	xxl
Width	< 576 px	>= 576 px	>= 768 px	>= 992 px	>= 1200 px	>= 1400 px

Grid

- The grid system of Bootstrap uses containers, rows, and columns (col) to layout/align content.
- All column cells are wrapped in a row.
- Maximum 12 columns can be placed in a row.

<table>
<tr><td colspan="3">Col-3</td><td colspan="3">Col-3</td><td colspan="3">Col-3</td><td colspan="3">Col-3</td></tr>
<tr><td colspan="4">Col-4</td><td colspan="4">Col-4</td><td colspan="4">Col-4</td></tr>
<tr><td colspan="6">Col-6</td><td colspan="6">Col-6</td></tr>
<tr><td colspan="2">Col-2</td><td colspan="2">Col-2</td><td colspan="2">Col-2</td><td colspan="2">Col-2</td><td colspan="2">Col-2</td><td colspan="2">Col-2</td></tr>
<tr><td>Col-1</td><td>Col-1</td><td>Col-1</td><td>Col-1</td><td>Col-1</td><td>Col-1</td><td>Col-1</td><td>Col-1</td><td>Col-1</td><td>Col-1</td><td>Col-1</td><td>Col-1</td></tr>
</table>

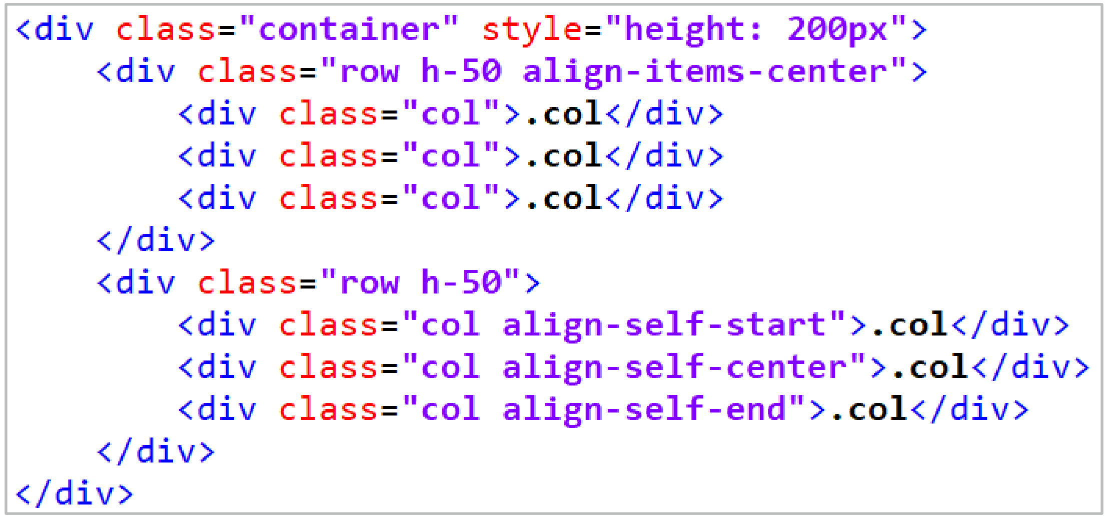
Grid (Cont.)

- Column width is set in percentage; hence, it is always fluid and relative to its parent element.
- The grid structure can be nested.

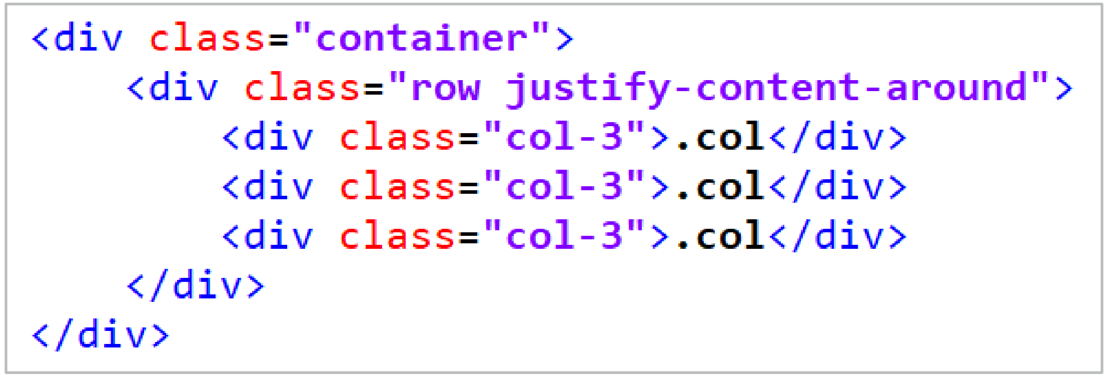
Columns	• The default class column takes up all the available space and divides it equally (a total of 12 columns) between all the columns in a row. • The 3-grid system has four tiers of classes in Bootstrap : xs (phones), sm (tablets), md (desktops), and lg (larger desktops). • The size can be specified individually for one or more columns. (for example, col-3 or col-5). • The class 'col-auto' will take up only the required space. (for example, col-auto). • The responsive columns can be created using multiple breakpoint classes in one column. (for example, col-12 col-md-6 col-lg-3).
Rows	• The class set to rows applies to all columns within. (for example, row row-cols-auto // row row-cols-2). [!] cols-2 here is not col size. It's the number of cols in a row. • Responsive cols can be created using multiple breakpoint classes. (for example, row row-cols-1 row-cols-md-2 row-cols-lg-3 row-cols-xl-4).

Column

- Vertical alignment (Row level)
  - `.row .align-items-start`
  - `.row .align-items-center`
  - `.row .align-items-end`
- Vertical alignment (Column level)
  - `.col .align-self-start`
  - `.col .align-self-center`
  - `.col .align-self-end`
- Horizontal alignment (Row level)
  - `.row .justify-content-start`
  - `.row .justify-content-center`
  - `.row .justify-content-end`
  - `.row .justify-content-around`
  - `.row .justify-content-between`
  - `.row .justify-content-evenly`
- Column wrapping and break
  - Columns beyond 12 size wraps to the next row
  - Use of class `.w-100` on a division (div) forces subsequent cols to wrap



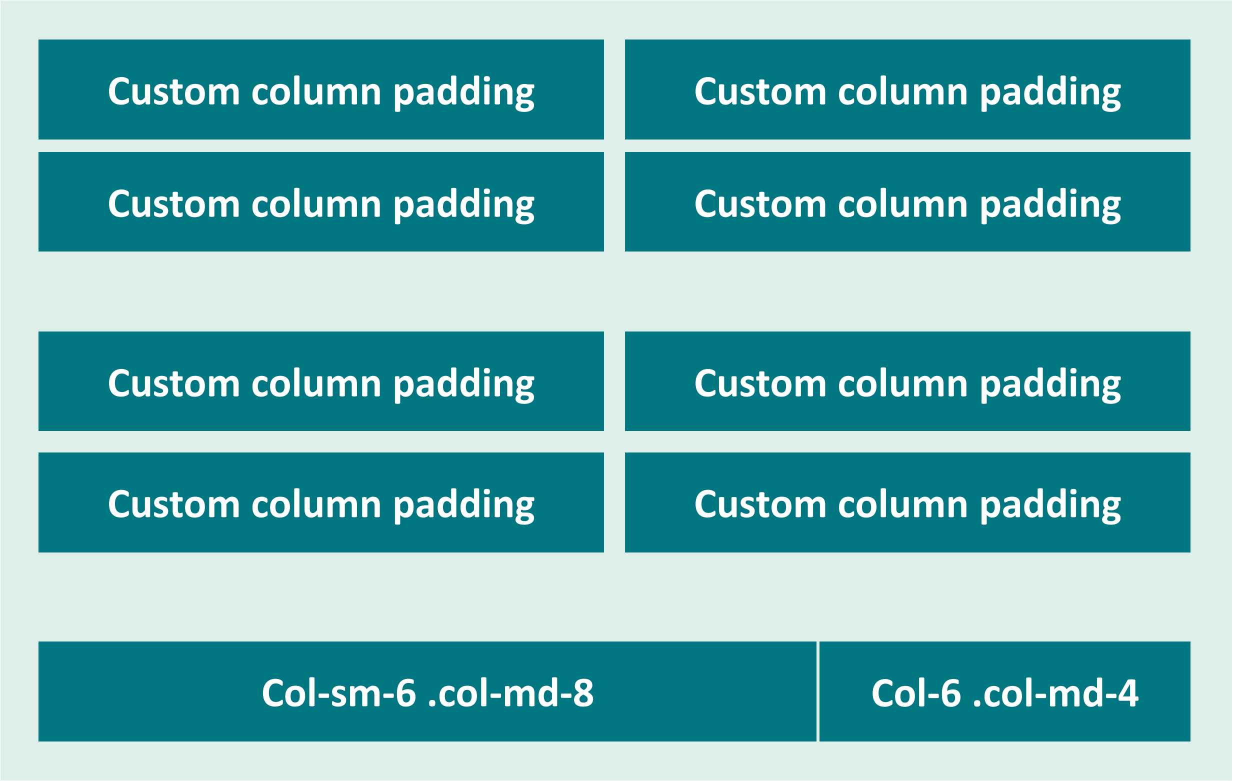
```html
<div class="container" style="height: 200px">
    <div class="row h-50 align-items-center">
        <div class="col">.col</div>
        <div class="col">.col</div>
        <div class="col">.col</div>
    </div>
    <div class="row h-50">
        <div class="col align-self-start">.col</div>
        <div class="col align-self-center">.col</div>
        <div class="col align-self-end">.col</div>
    </div>
</div>
```



```html
<div class="container">
    <div class="row justify-content-around">
        <div class="col-3">.col</div>
        <div class="col-3">.col</div>
        <div class="col-3">.col</div>
    </div>
</div>
```

Gutters

- Gutters have margins (space/gap) between columns set at the row level.
- Class `.gx-*` can be used for horizontal gutter.
- Class `.gy-*` can be used for vertical gutter.
- Class `.g-*` can be used to set a common value for both horizontal gutters and vertical gutters.
- Class `.g-0` will reduce gutter to zero, which means there will be no gutter.
- Gutters can be responsively adjusted (example, g-2 g-lg-4).



Custom column padding | Custom column padding

Custom column padding | Custom column padding

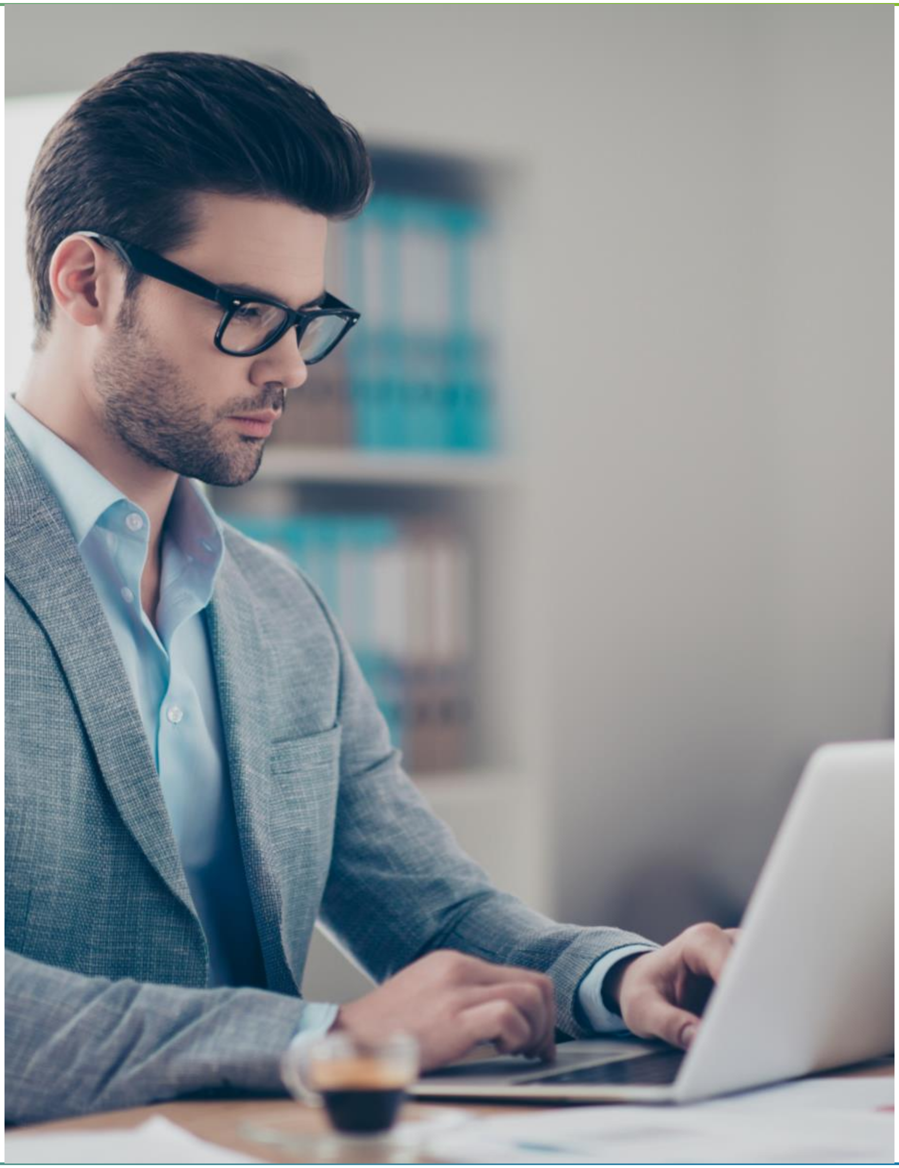
Custom column padding | Custom column padding

Custom column padding | Custom column padding

Col-sm-6 .col-md-8 | Col-6 .col-md-4

</ >

Content



Contextual Colors

- Colors can be used as the background using `bg-*color`
- Class .gr-gradient can be added along with `bg-*color`
- Colors are used in different contexts with different class names:
  - `.btn-primary`
  - `.border-primary`
  - `.text-primary`
  - `.alert-primary`
  - `.link-primary`

Primary link

Secondary link

Success link

Danger link

Warning link

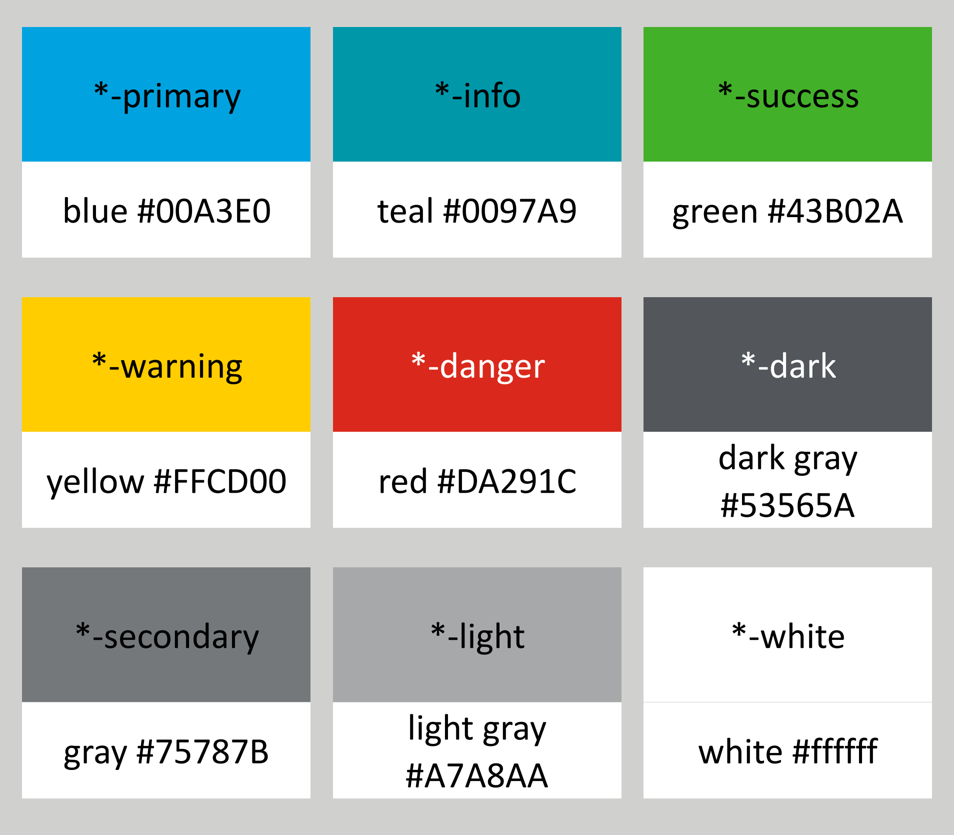
Info link

Light link

Dark link

Muted link

White link



*-primary	*-info	*-success
blue #00A3E0	teal #0097A9	green #43B02A
*-warning	*-danger	*-dark
yellow #FFCD00	red #DA291C	dark gray #53565A
*-secondary	*-light	*-white
gray #75787B	light gray #A7A8AA	white #ffffff

</ >

Typography

- The HTML follows a default font size of 16px.
- The HTML follows a default line height of 1.5.
- The HTML tag `<small>` with .text-muted class wrapped in regular `<h1>` … `<h6>` tags creates secondary headings.
- The HTML `<h1>` tag with class `.display-1` … `.display-6` creates display headings.
- The HTML tag `<p>` with class .lead creates lead paragraphs.
- The HTML tag `<abbr>` with title attribute and `.initialism` creates abbreviation style.

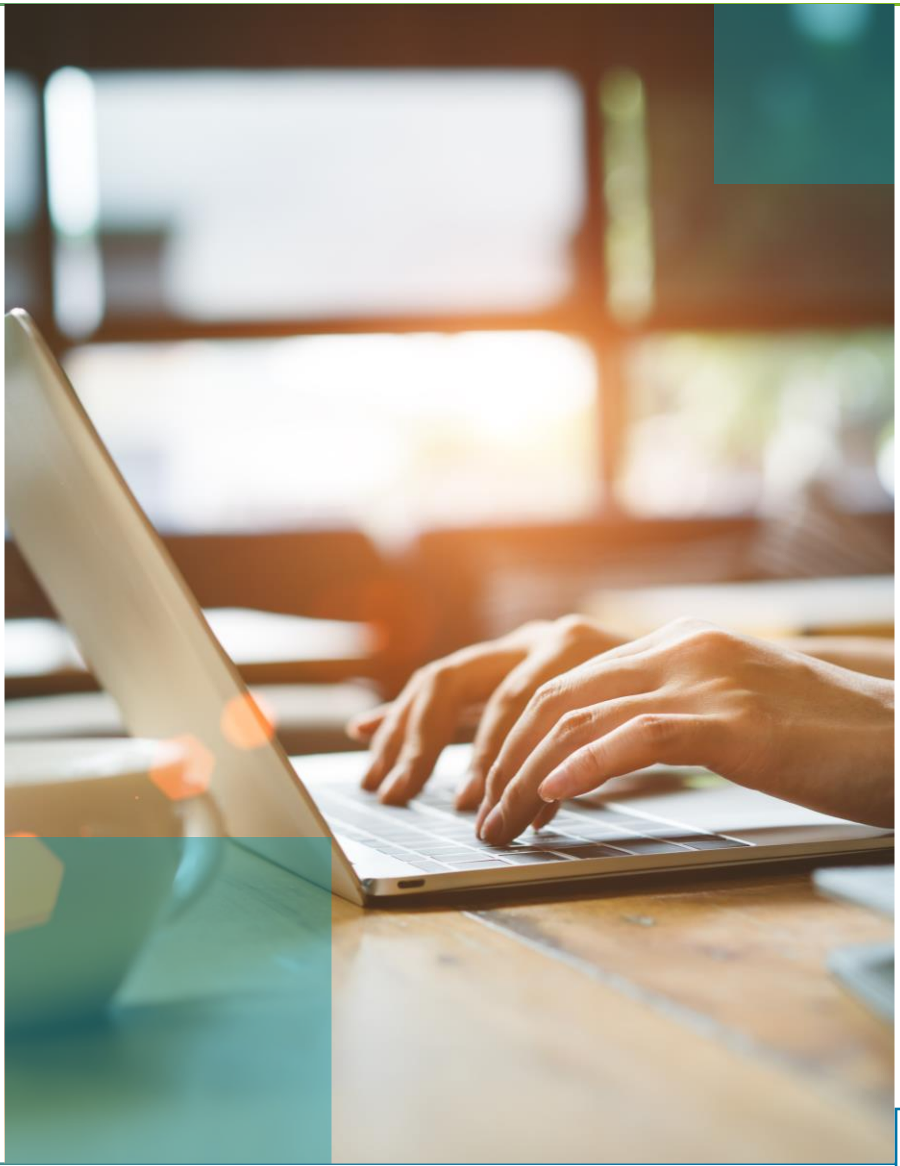
Styles	HTML Tags	Bootstrap Classes
Headings	`<h1>` … `<h6>`	`h1` … `h6`
Highlighted Text	`<mark>`	`.mark`
Deleted Text	`<del>`	--
Strike-Through Text	`<s>`	`.text-decoration-line-through`

Typography (Cont.)

Styles	HTML Tags	Bootstrap Classes
Inserted Text	`<ins>`	--
Underline Text	`<u>`	`.text-decoration-underline`
Fine-print Text	`<small>`	`.small`
Strong Text	`<strong>`	--
Emphasized Text	`<em>`	--
Bold Text	`<b>`	--
Italic Text	`<i>`	--

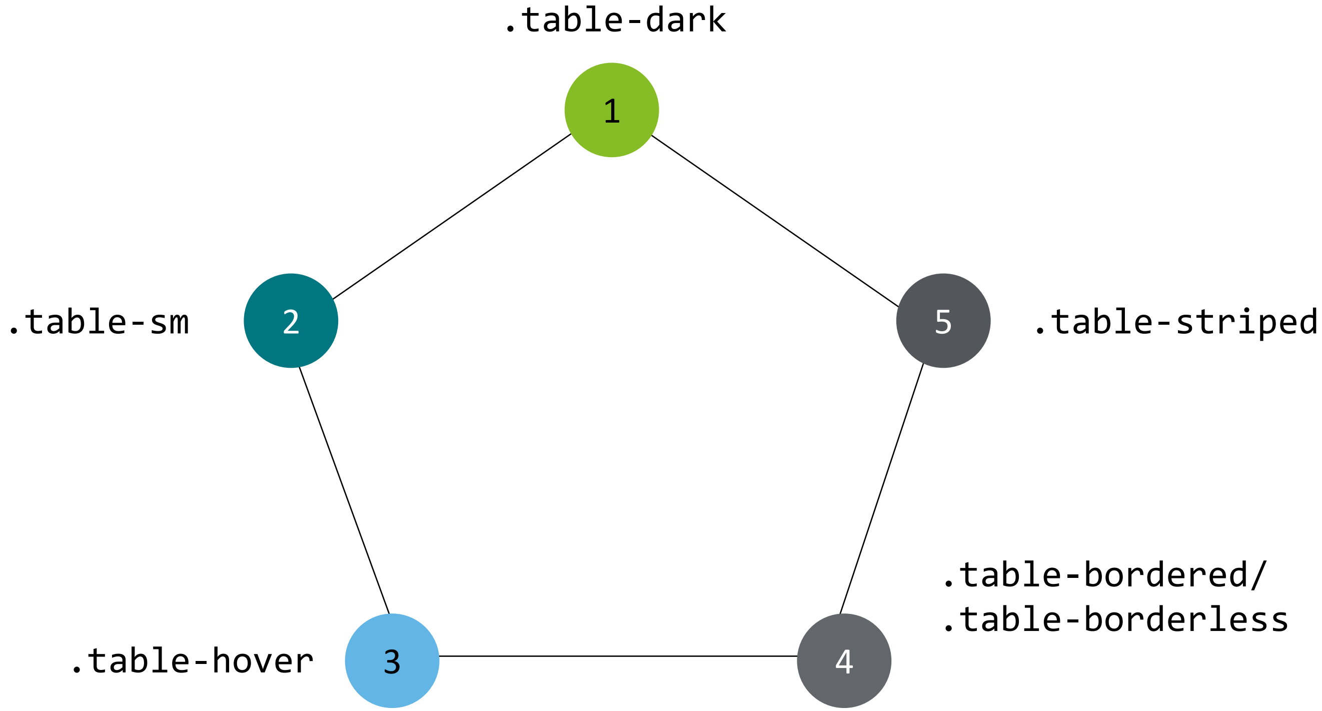
Images

- Appearance:
  - Class `.img-fluid` is used to render images as responsive.
  - Class `.rounded` is used to add rounded corners to an image.
  - Class `.img-thumbnail` is used to create a thumbnail look.
- Float images:
  - Class `.float-start` is used to align an image to the left.
  - Class `.clearfix` utility is used on the parent division.
  - Class `.float-end` is used to align an image to the right.



Tables

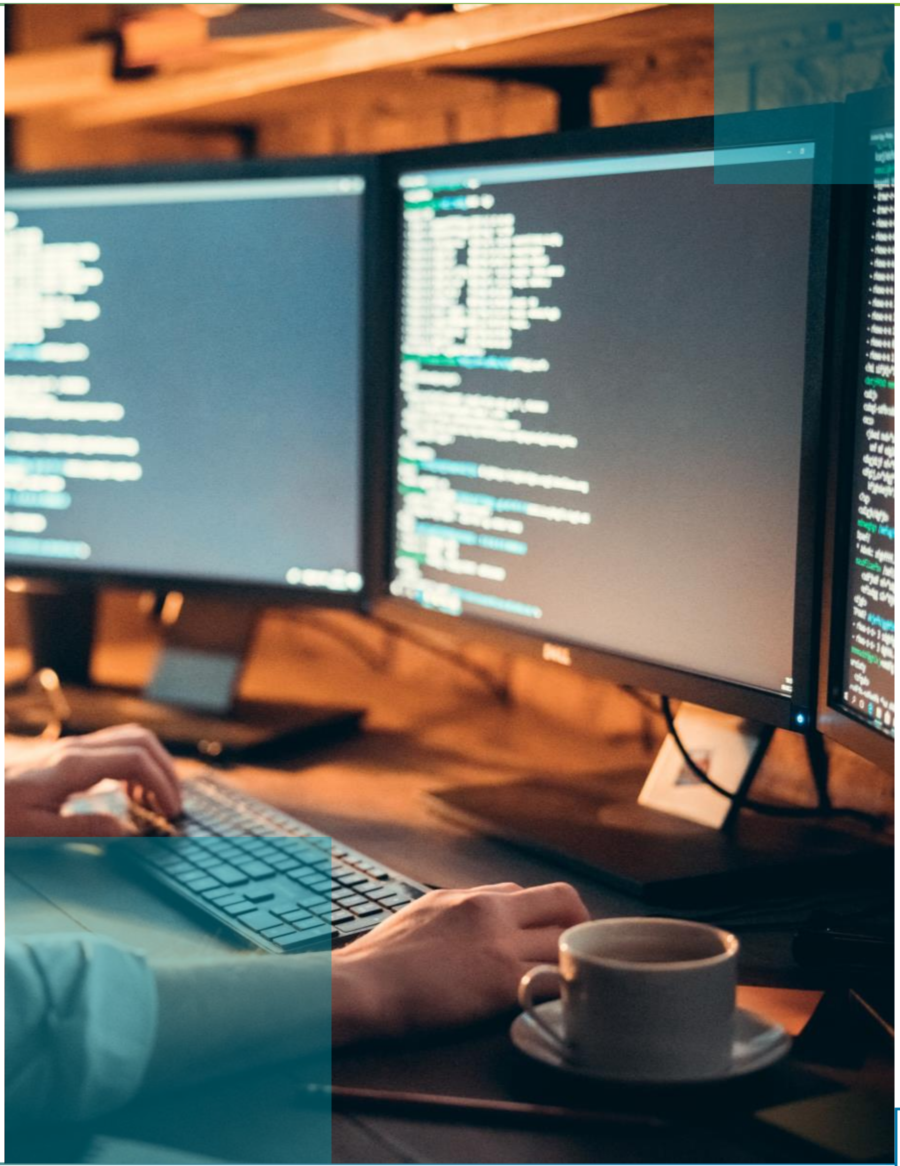
- Class `.table` is used for the HTML <table> tag.
- Class `.table-light` or `.table-dark` styles the HTML <thead> tag.
- Additional classes can be applied with the basic ‘table’ style, and these classes can be used in combination.



With borders, contextual colors can be used.
(Example: `.border-primary`)

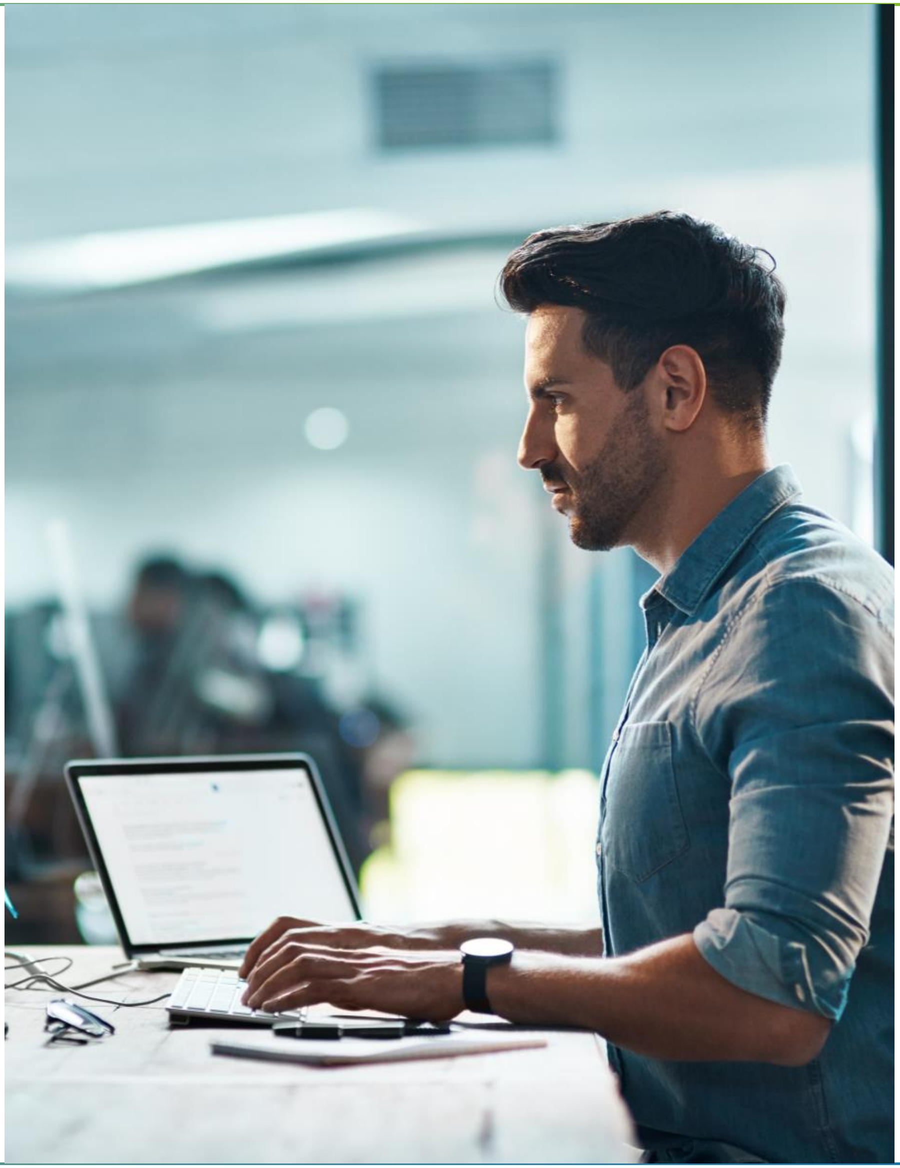
Tables (Cont.)

- Class `.table-responsive` class makes the table area scrollable.
- One row (<tr>) or cell (<td>) can be marked as active by applying class `.table-active` to it.
- Contextual classes, such as primary and secondary, can be used with the prefix table at the entire table (<table>), row, (<tr>), or cell (<td>) level.
- Class `.align-middle`, `.align-top`, and `.align-bottom` can be used with the entire table (<table>), row (<tr>), or cell (<td>) for vertical alignment of the content within.
- Nested tables do not inherit styles from the parent.



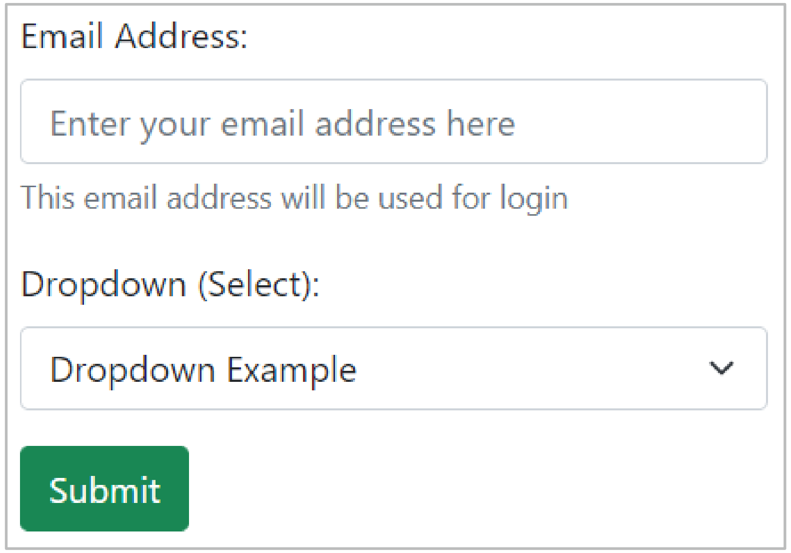
</ >

Forms



Controls

- Class `.form-label` is used for label control.
- Class `.form-control` is used for input elements, such as text, color, or text area.
- Class `.form-select` is used for drop-down controls, multiple attributes and size attribute in HTML can be used as is.
- Class infix small (sm) and large (lg) can be added to primary class (example, form-control form-control-sm).



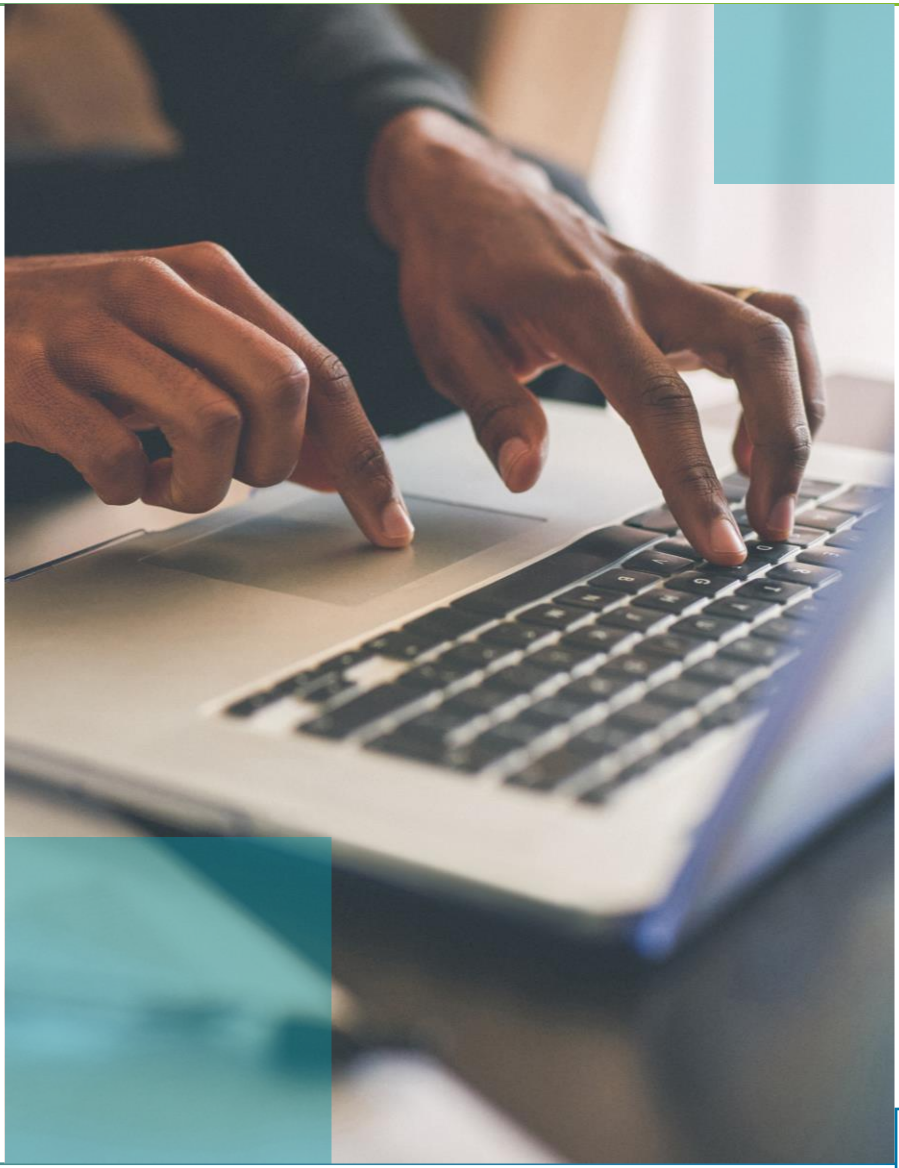
Email Address:

Enter your email address here

This email address will be used for login

Dropdown (Select):

Dropdown Example



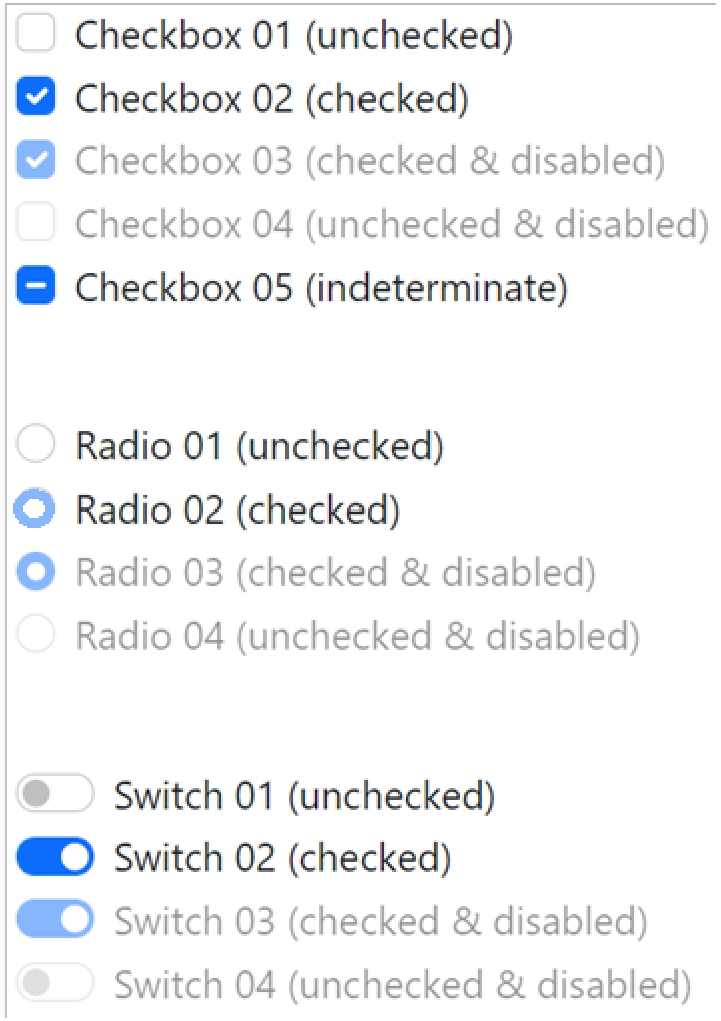
Controls (Cont.)

- Class `.form-text` is used for paragraphs or text in form.
- For read-only behavior, either add read-only HTML attribute to the control (shows control) or use class .form-control-plaintext (for text only) appearance.
- Horizontal forms can be created using rows and columns.
- The disabled attribute in HTML is used to disable an individual element or all controls in a field set.

```
<div class="mb-3 mt-3">
 <label for="emailAddress" class="form-label">
     Email Address:
   </label>
   <input type="email" class="form-control"
       placeholder="Enter your email address here"
       id="emailAddress" />
    <p class="form-text">Email address for login</p>
</div>
<div class="mb-3">
     <label for="ddOptions" class="form-label">
         Dropdown (Select):
     </label>
     <select class="form-select" id="ddOptions">
       <option selected>Dropdown Example</option>
       <option value="1">One</option>
       <option value="2">Two</option>
       <option value="3">Three</option>
     </select>
 </div>
<button type="submit" class="btn btn-success">
 Submit
 </button>
```

Checks and Radios

- Controls need to be wrapped in a division with Class `.form-check` in order to enhance the cross-browser rendering.
- Class `.form-check-input` is used for input control with type equals to checkbox or radio.
- Class `.form-check-label` is used for corresponding text label.
- The exact same value needs to be provided for the HTML name attribute on radio controls to enforce mutually exclusive selection.
- The standard checked and disabled attributes in HTML can be used for desired behaviors.
- Additional classes for wrapper division is to be applied along with `.form-check.`
  - `.form-switch` converts input type ="checkbox" to a slider switch.
  - `.form-check-inline` converts default vertical stacked into horizontal inline.



☐ Checkbox 01 (unchecked)

☑ Checkbox 02 (checked)

☑ Checkbox 03 (checked & disabled)

☐ Checkbox 04 (unchecked & disabled)

⊟ Checkbox 05 (indeterminate)

○ Radio 01 (unchecked)

● Radio 02 (checked)

● Radio 03 (checked & disabled)

○ Radio 04 (unchecked & disabled)

Switch 01 (unchecked)

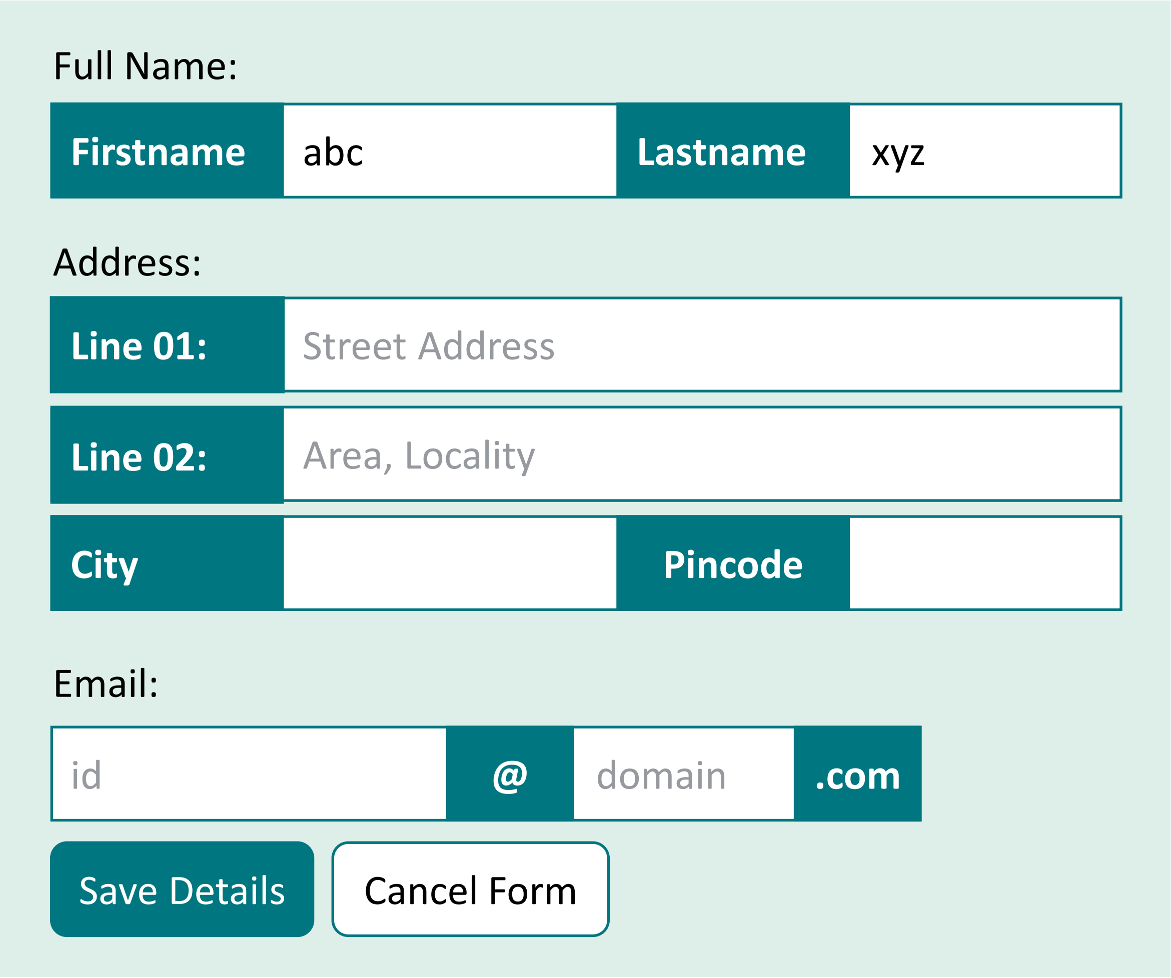
Switch 02 (checked)

Switch 03 (checked & disabled)

Switch 04 (unchecked & disabled)

Input Group

- Easily extend form input controls by adding texts, buttons, and drop-down boxes.
- Wrapper division with Class `.input-group` is required to enclose multiple controls.
- Multiple controls and multiple add-ons can be added to one input group.
- Class infix `sm` and `lg` can be added to the wrapper division (example, `input-group input-group-sm).`
- For text elements in group, use `.input-group-text class.`
- Checkboxes and radio controls can also be placed in a division container with the `.input-group-text` class.
- Buttons, select, and other Bootstrap controls, including ul/li-based drop-down boxes and segmented buttons, can be added.



Full Name:

Firstname	abc	Lastname	xyz

Address:

Line 01:	Street Address
Line 02:	Area, Locality

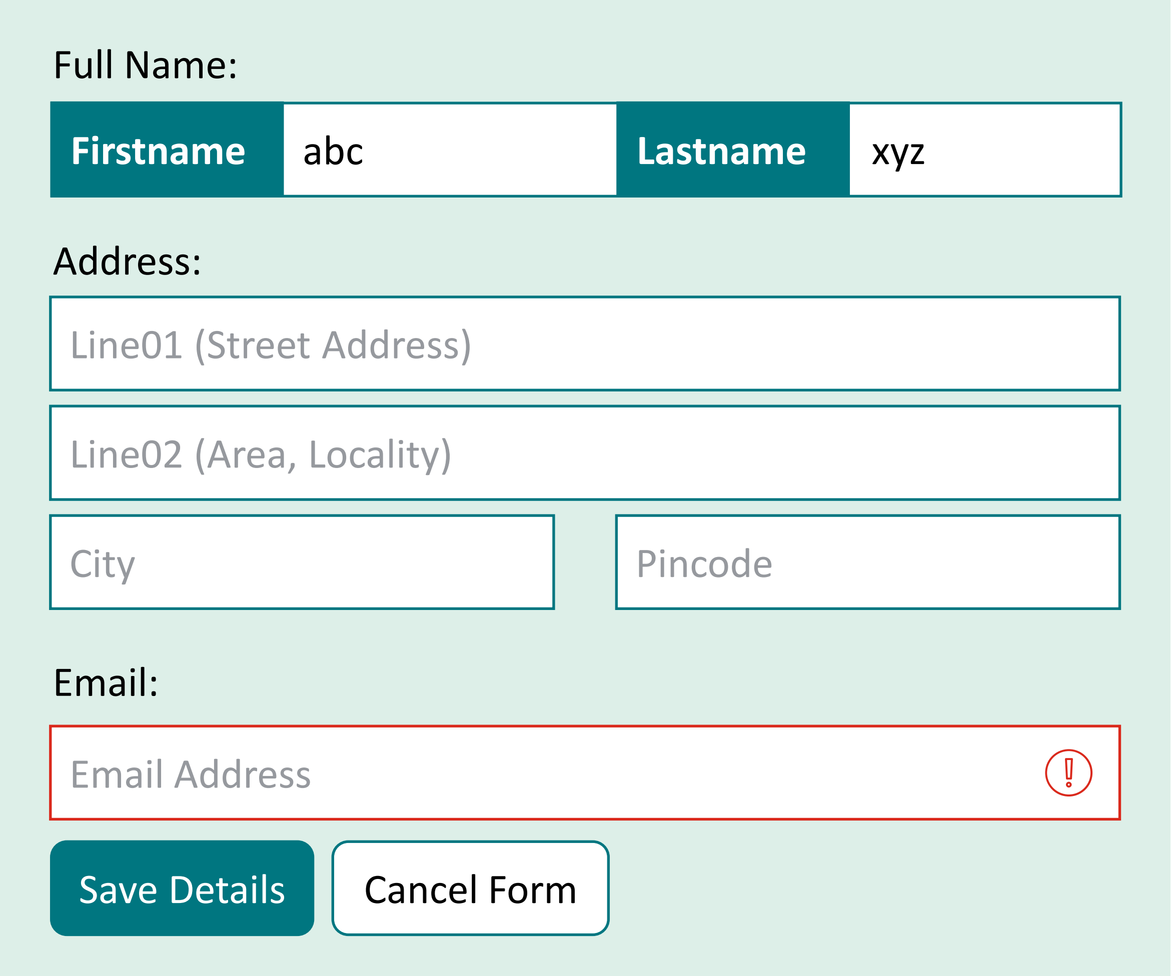
City		Pincode	

Email:

id	@	domain	.com

Floating Labels

- Floating labels move input control placeholder text above the value, when empty label text appears as placeholder of the control.
- Wrapper division with class `.form-floating` is required to enclose controls.
- One set of input and label control is placed in form-floating division.
- The text label plays a dual role of a control placeholder as well as a floating label.
- The text label turns into a floating label, when an empty input control gets focus or if a control has set value.
- The placeholder attribute is not displayed; however, it's required for floating labels to work properly.
- Form validations works fine with floating label controls.



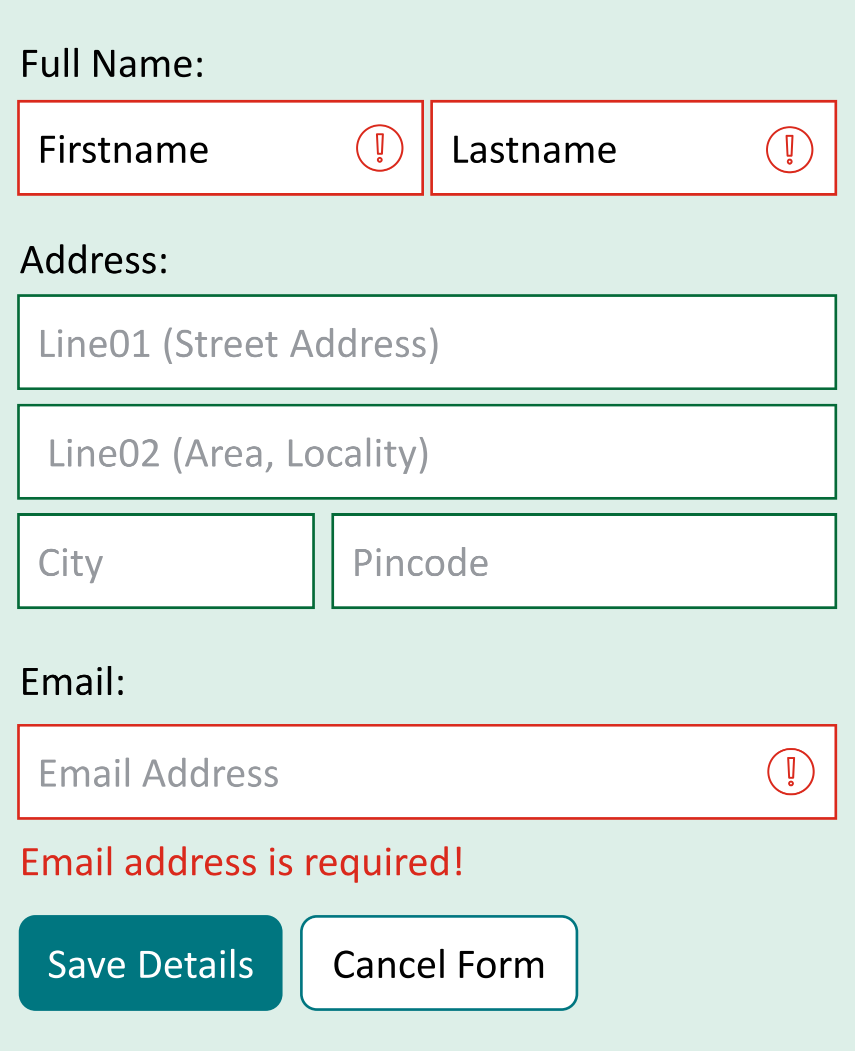
Full Name:

Firstname abc

Lastname xyz

Validation

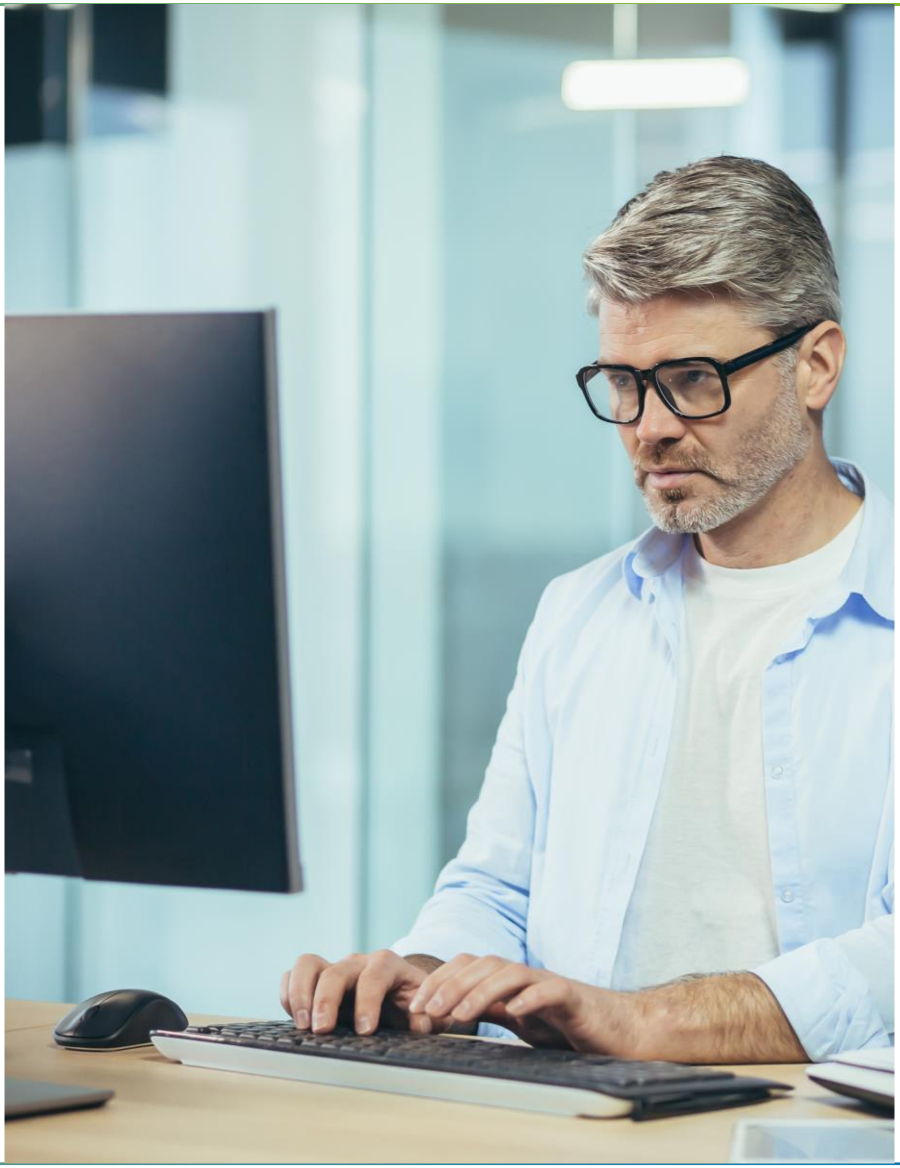
- Add attribute `novalidate` and `.needs-validation` class in HTML to `<form>` tag.
- Add the required attribute in HTML on all the mandatory input controls.
- Add JavaScript to prevent the default form submissions and add the required classes to form.
- Keep all element, label, and validation messages related to input controls in a wrapper division.
- Upon form submission, by default all fields marked as required and kept empty would be highlighted with an error icon, while the remaining fields would be considered as valid, even if kept empty.
- Additionally, messages can be displayed:
  - Div with class `.valid-feedback / .valid-tooltip` to display success message
  - Div with class `.invalid-feedback / .invalid-tooltip` to display error message
- With custom JavaScript or any other server-side custom validation logics, `.is-valid` and `.is-invalid` classes can be applied to the corresponding input elements. Any input groups in form would require `.has-validation` class for proper functioning of validations.



Email address is required!

</>

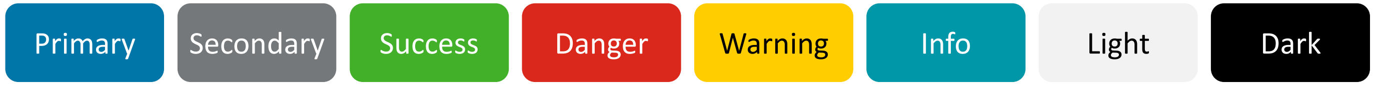
Components and Utilities



Buttons

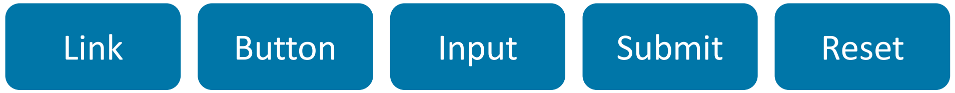
Bootstrap provides us with different classes that enable you to change the styles and sizes of buttons.

```
<button type="button" class="btn btn-primary">Primary</button>
```

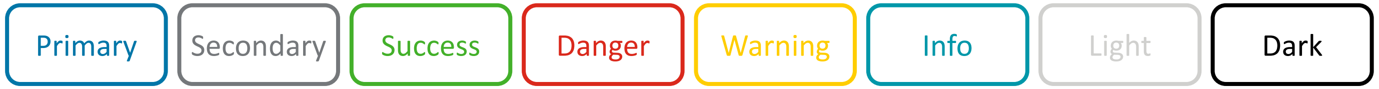


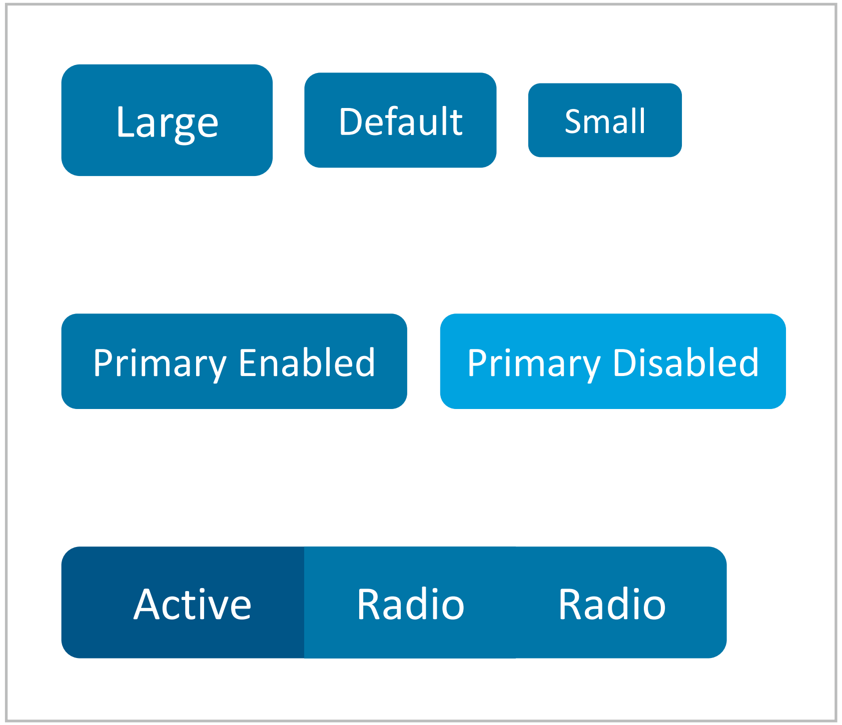
```
<button type="button" class="btn btn-link">Link</button>
```

```
<a class="btn btn-primary" href="#" role="button">Link</a>
<button class="btn btn-primary" type="submit">Button</button>
<input class="btn btn-primary" type="button" value="Input">
<input class="btn btn-primary" type="submit" value="Submit">
<input class="btn btn-primary" type="reset" value="Reset">
```



```
<button type="button" class="btn btn-outline-primary">Primary</button>
```





Alerts

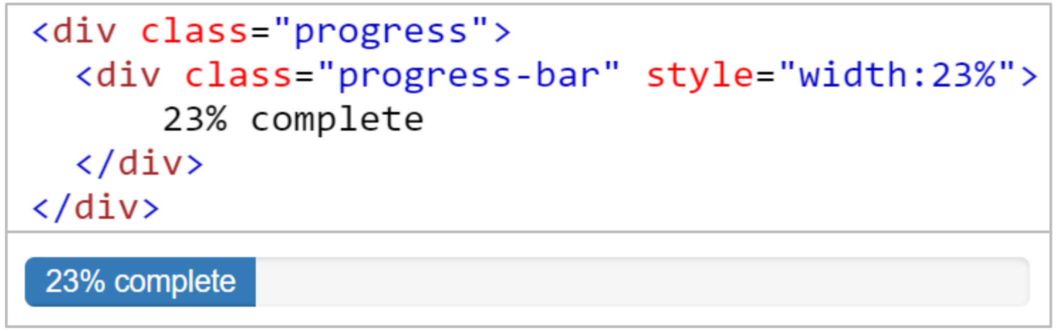
- For typical user actions alerts provide contextual feedback messages with a handful of available and flexible alert messages.
- For a basic alert, the ‘alert’ and contextual class is added to a division.
- Multiple lines (with <p>, <br>, and <hr> tags) can be created.

```
<div class="alert alert-primary" role="alert">
  Primary Alert!
</div>
```

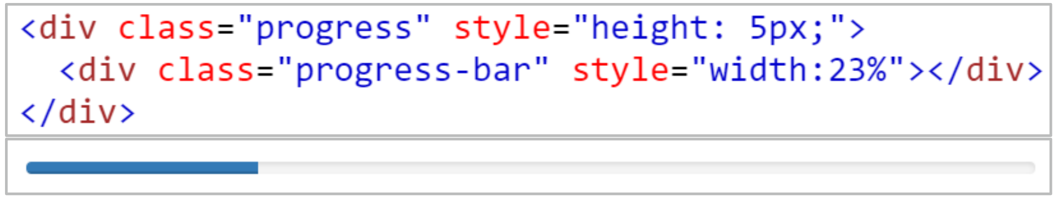
- The Close button can be added to an alert using ‘alert-dismissible.’ class.

Progress Bar

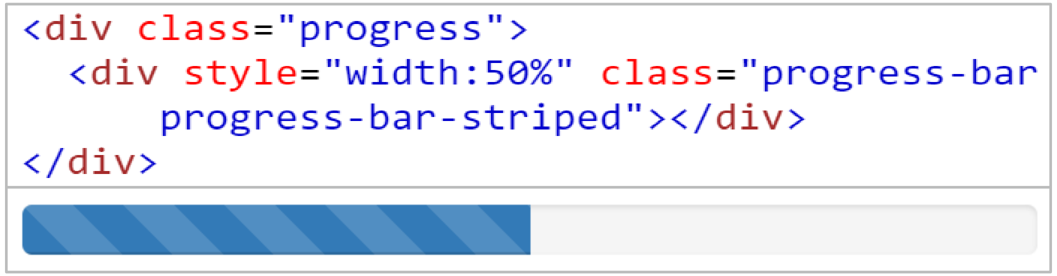
- A wrapper division is needed with the `.progress` class.
- An inner division is needed with the `.progress-bar` class.
- The width of the inner division (with the `.progress-bar` class) determines the progress. The width can be set with inline style, utility class, or custom CSS.
- Labels can be applied by adding text as content to the inner division (`.progress-bar`) element.
- Height can be applied on the wrapper division (`.progress`) element.
- Contextual styles (`bg-success, bg-info`) can be used along with `.progress-bar` to give a background color.
- Multiple inner division elements can be added in a wrapper division to divide progress in multiple parts with different colors.
- The `progress-bar-striped` class can be used along with the contextual color background classes for a striped look.



```
<div class="progress">
  <div class="progress-bar" style="width:23%">
      23% complete
  </div>
</div>
```



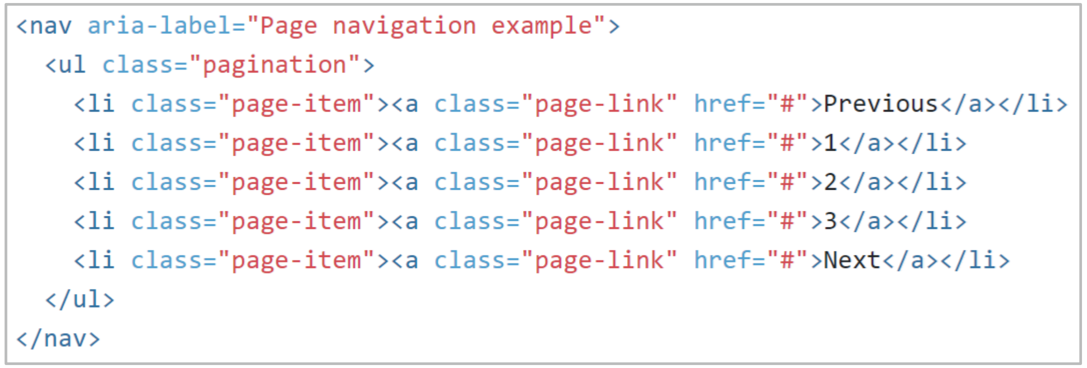
```
<div class="progress" style="height: 5px;">
  <div class="progress-bar" style="width:23%"></div>
</div>
```



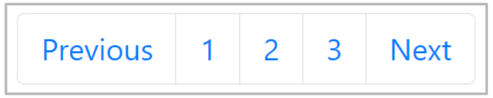
```
<div class="progress">
  <div style="width:50%" class="progress-bar
      progress-bar-striped"></div>
</div>
```

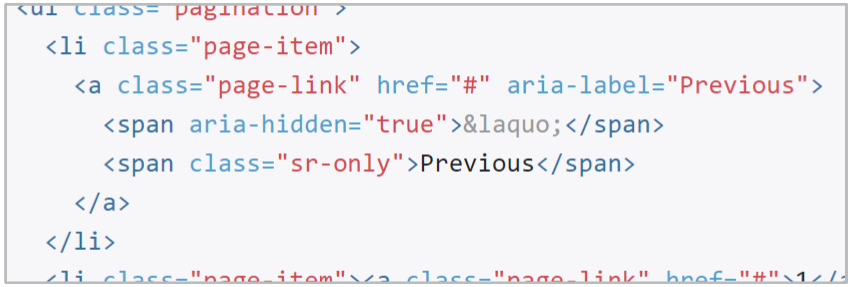
Pagination

- Html element <ul> can be used as a wrapper with the .pagination class. Its child element <li> needs to use the .page-link class for an individual page link.
- As shown in the example, the previous and the next text can be converted into icons.
- The .active or .disabled classes can be used with .page-item to mark the corresponding page link appropriately.
- The .pagination-lg or .pagination-sm classes can be used with .pagination for the desired size.
- By default, the pagination is aligned to the left. To align it to center or to right, .justify-content-center or .justify-content-end classes can be used with .pagination.

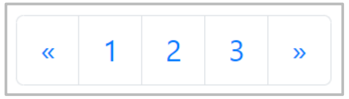


```
<nav aria-label="Page navigation example">
  <ul class="pagination">
    <li class="page-item"><a class="page-link" href="#">Previous</a></li>
    <li class="page-item"><a class="page-link" href="#">1</a></li>
    <li class="page-item"><a class="page-link" href="#">2</a></li>
    <li class="page-item"><a class="page-link" href="#">3</a></li>
    <li class="page-item"><a class="page-link" href="#">Next</a></li>
  </ul>
</nav>
```



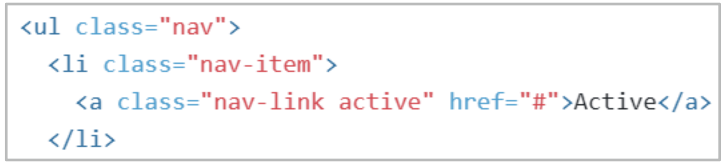


```
<ul class="pagination">
  <li class="page-item">
    <a class="page-link" href="#" aria-label="Previous">
      <span aria-hidden="true">&laquo;</span>
      <span class="sr-only">Previous</span>
    </a>
  </li>
  <li class="page-item"><a class="page-link" href="#">1</
```

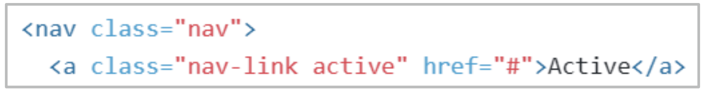


Navs

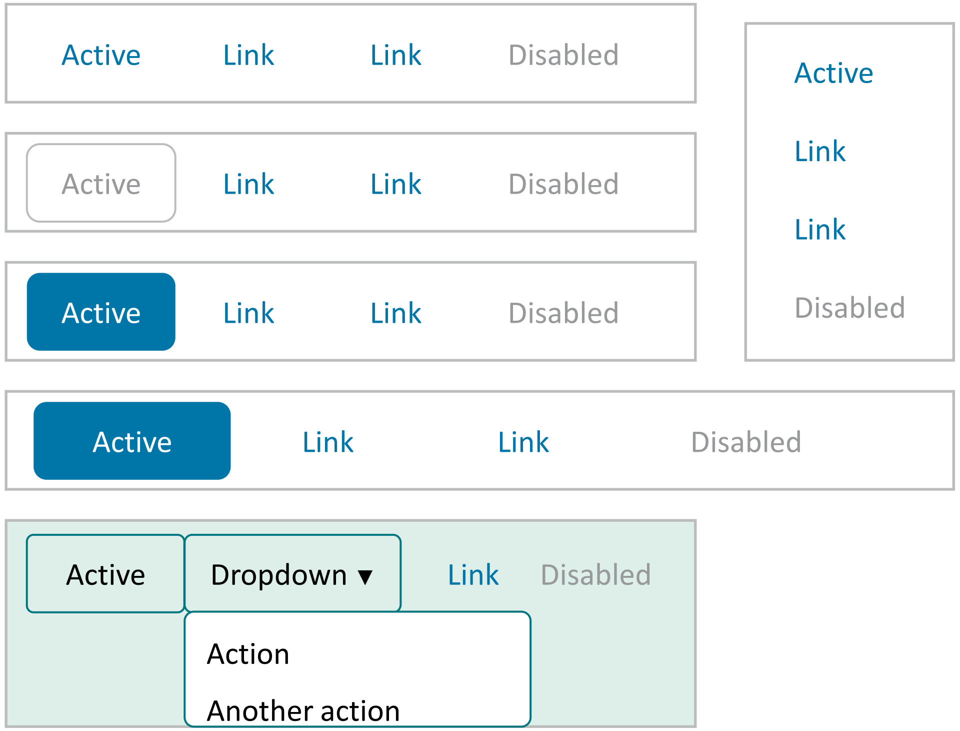
- The basic navigation bar is created with the <ul> and <li> hierarchy with classes .nav and .nav-item, respectively. The anchor element (<a>) with the .nav-link class is placed in <li>.
- Alternatively, the HTML 5 <nav> element can be used with the .nav and anchor links with .nav-link classes.
- By default, links are aligned to the left of the container. .justify-content-center or justify-content-end can be used with .nav to align it to center or to right.
- The .flex-column class can be used to change the alignment to a vertical layout.
- The .nav-tabs, .nav-pills or .nav-fill classes are used for different style and appearance.
- The dropdown boxes can be added to the navigation bar using dropdown, dropdown-menu, and dropdown-item classes.
- The activating tabs use the data-toggle attribute and .tab-content and .tab-pane classes.



```
<ul class="nav">
  <li class="nav-item">
    <a class="nav-link active" href="#">Active</a>
  </li>
```

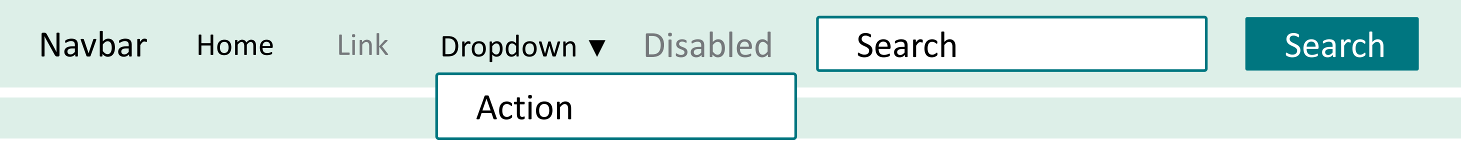


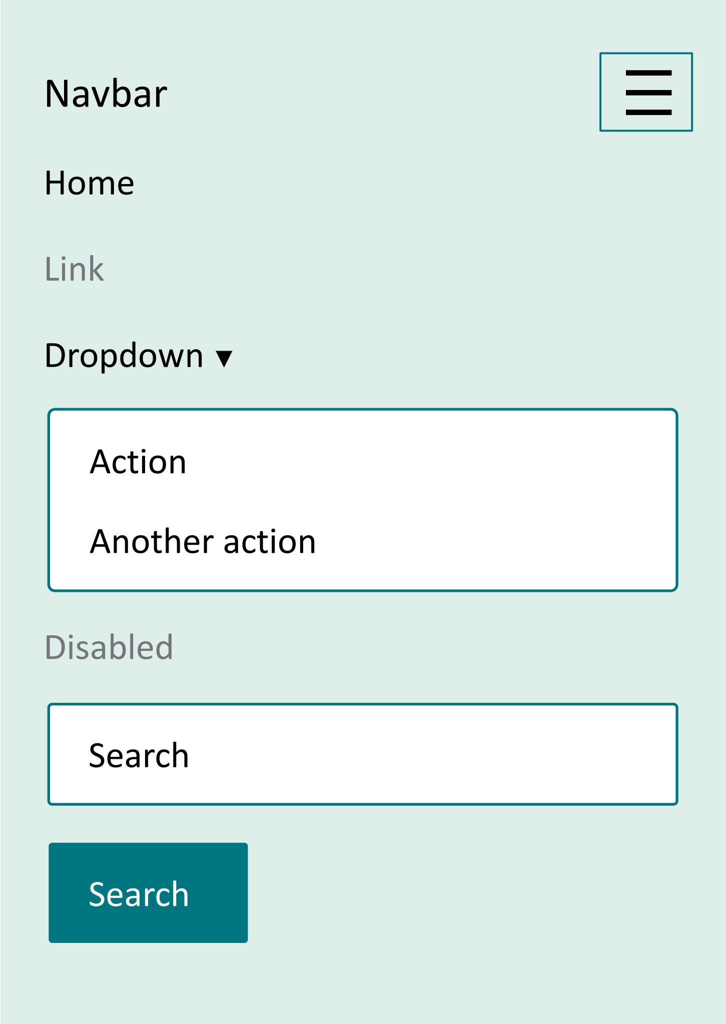
```
<nav class="nav">
  <a class="nav-link active" href="#">Active</a>
```


Active Link Link Disabled
Active Link Link Disabled
Active Link Link Disabled
Active Link Link Disabled
Active Link Link Disabled
Active Dropdown ▾ Link Disabled
Action
Another action

Navbar

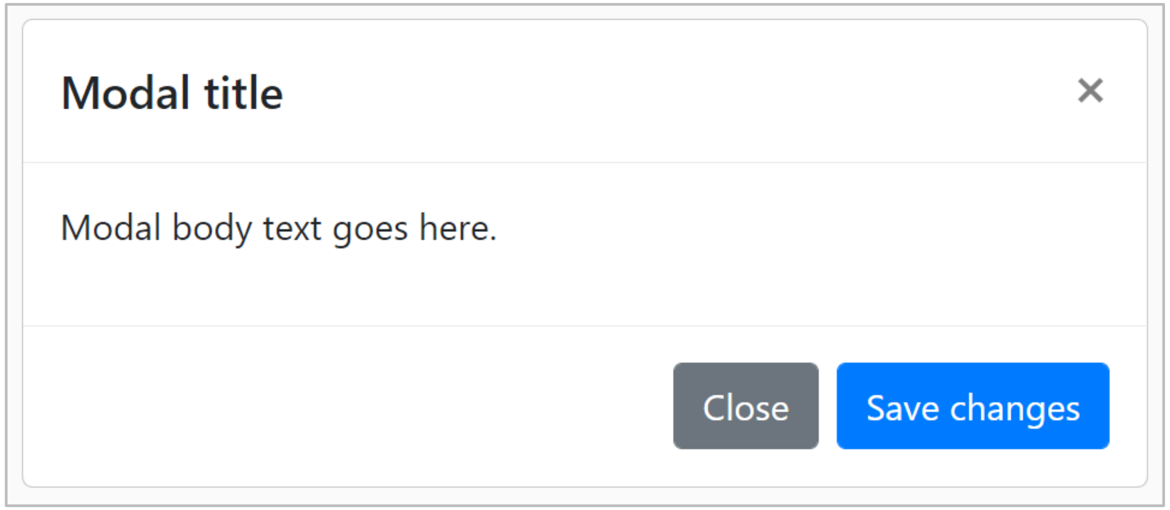
- The navigation bar includes the title, links, and search functionality and supports the following sub-components:
  - `.navbar-brand` can be used for your company, product, or project name.
  - `.navbar-nav can be used` for a full-height and lightweight navigation (including support for drop-down boxes).
  - `.navbar-toggler` can be used with our collapse plugin and other navigation toggling behaviors.
  - `.form-inline` can be used for any form controls and actions.
  - `.navbar-text` can be used for adding vertically centered strings of text.
  - `.collapse` and `.navbar-collapse` can be used for grouping and hiding navbar contents by a parent breakpoint.
  - `modal-lg` and `.modal-sm` can be used with modal-dialog to change modal size.
- Contextual `bg-color` classes can be used to provide background color to the navigation bar.
- `.fixed-top`, `fixed-bottom`, or `sticky-top` can be used for placement of navbar.





Modal

- Modals are built with CSS, HTML, and JavaScript. They remove scroll from the <body> so that modal content scrolls instead and are positioned over everything else in the document.
- Modal is created by creating division elements with respective Bootstrap classes in the following hierarchy:
  - `.modal`–dialog wrapper
    - `.modal-dialog`–dialog document
  - `.modal-content`–`content` wrapper
    - `.modal-header`
    - `.modal-body`
    - `.modal-footer`
- `.modal-lg` and `.modal-sm` can be used with `modal-dialog` to change modal size.
- `.fade` can be added with `.modal` for the faded effect.



Modal title

Modal body text goes here.

Utilities

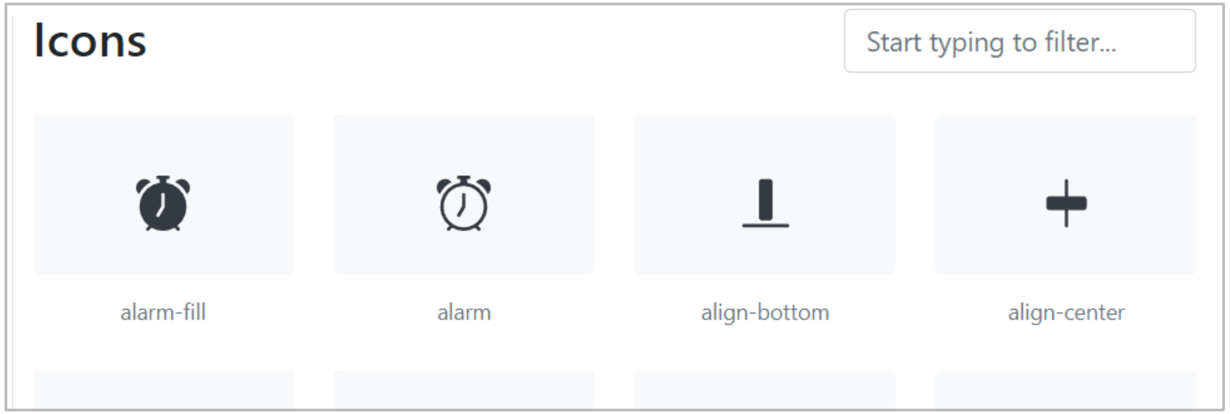
Borders	• Use `border, border-top, border-right` classes to add element borders. • Remove border by suffixing `-0` (example, `border-0 and border-top-0).` • Contextual classes can be used for adding border color (example, `border-primary).` • Rounded class is used to add or remove (using -0 at end) radius (example, `rounded,` and `rounded-top`).
Clearfix	With the use of `Clearfix` class on wrapper, and `float-left,` or `float-right` at child element level, floating effect can be achieved.
Spacing	The `mx-auto` class horizontally aligns division to center as auto is set to `left-right` margin. (`m` represents margin, x represents left and right)
Sizing	• Fixed-width classes `w-25, w-50, w-75` and `w-100` represent 25%, 50%, 75% and 100% width, respectively. • Fixed-height classes `h-25, h-50, h-75` and `h-100` represent 25%, 50%, 75% and 100% height, respectively.

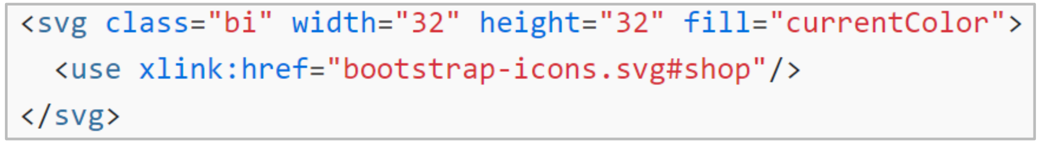
Utilities (Cont.)

Vertical Align	`.align-baseline, .align-top, .align-middle, .align-bottom, .align-text-bottom, and .align-text-top` classes are available for vertical alignment.
Visibility	`.visible` or `.invisible` classes can be used to control the visibility of elements.
Position	• Non-responsive common classes—`.position-static, .position-relative, .position-absolute, .position-fixed, .position-sticky` • Fixed position classes—`.fixed-top, .fixed-bottom, .sticky-top`
Colors	`.text-primary, bg-primary` (all contextual colors)
Close Icon	`.close` class is used to style a close icon.

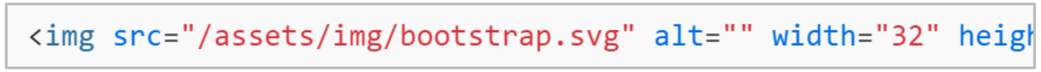
Icons

- Site URL: https://icons.getbootstrap.com/
  - All Bootstrap Icons are SVGs and can be installed with npm or downloaded from its site.
- Usage
  - Embedded: embeds icons SVG in HTML.
  - Sprite: references Bootstrap’s SVG file and then uses sprite ‘use’ tag to add the required icon.
  - External Image: copies SVGs to a local directory and references as image.
  - CSS Image URL: can be used in background—image as url parameter. # should be replaced with %23.
- Bootstrap styles can be applied to SVGs.

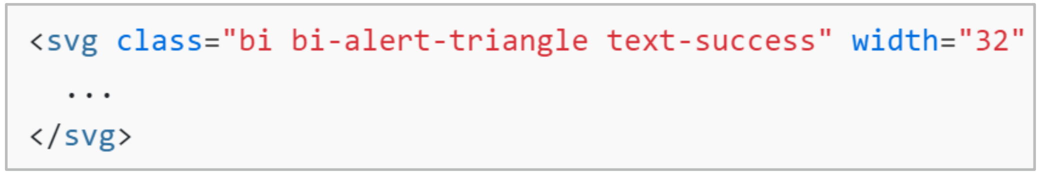




```
<svg class="bi" width="32" height="32" fill="currentColor">
  <use xlink:href="bootstrap-icons.svg#shop"/>
</svg>
```

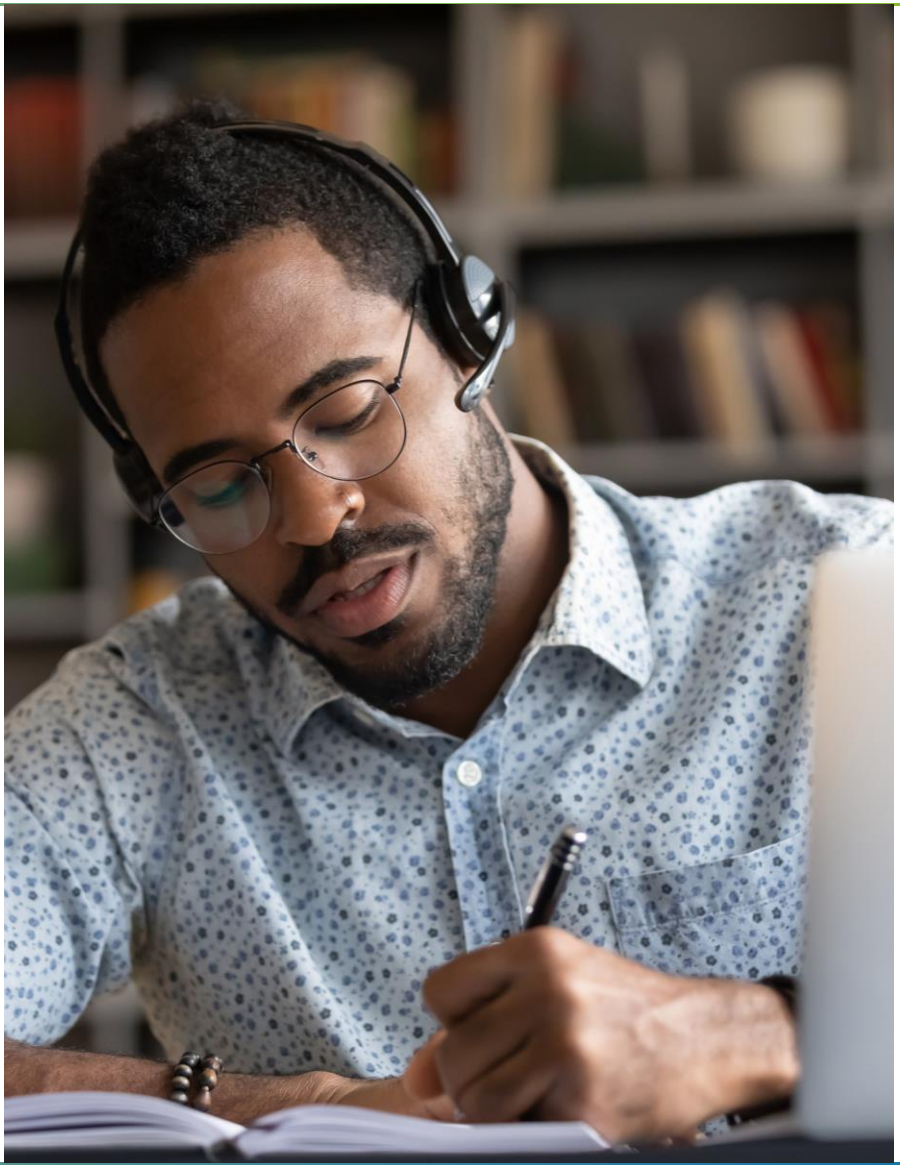


```
<img src="/assets/img/bootstrap.svg" alt="" width="32" heigh
```



```
<svg class="bi bi-alert-triangle text-success" width="32"
  ...
</svg>
```

Hands-On Labs



Hands-On Activity 1

Data Layout—Grid System	
Activity Details	Grid System • Create a Grid layout using division to display tabular data. • Make a responsive grid with: – 4 cells in a row on a large screen – 2 cells in a row on a medium screen – 1 cell in a row on a small screen Table • Create a tabular data using tables. • Make the table responsive. Images • Add an image thumbnail. • Add a responsive image.

Hands-On Activity 2

User Input	
Activity Details	• Create the HTML form with the following: – Form Controls – Submit Button – Reset Button – Input groups – Floating Labels – Required Field Validation • Create a Horizontal form using a Grid system. • Show alerts when the form submission is successful. • Add the close button to alert. • Add icons to form buttons. • Add a responsive image.

Summary

Here are the key learning points of the module.

- Bootstrap is amongst the most popular front-end frameworks for faster and easier web development.
- Bootstrap can be referred to, web pages via CDN hosting locally or through package managers.
- The grid system of Bootstrap uses containers, rows, and columns to layout/align content.
- Multiple controls and multiple add-ons can be added to the input group of Bootstrap.

Thank You

This communication contains general information only, and none of Deloitte Touche Tohmatsu Limited (“DTTL”), its global network of member firms or their related entities (collectively, the “Deloitte organization”) is, by means of this communication, rendering professional advice or services. Before making any decision or taking any action that may affect your finances or your business, you should consult a qualified professional adviser.

No representations, warranties or undertakings (express or implied) are given as to the accuracy or completeness of the information in this communication, and none of DTTL, its member firms, related entities, employees or agents shall be liable or responsible for any loss or damage whatsoever arising directly or indirectly in connection with any person relying on this communication. DTTL and each of its member firms, and their related entities, are legally separate and independent entities.