

# LEARN BOOTSTRAP

---

MEAN/MERN STACK

## LEARNING OBJECTIVES:

---

- ▶ Explain the core concept of Bootstrap.
- ▶ What is Bootstrap?
- ▶ What is Responsive Web Design?
- ▶ Get Bootstrap 5
- ▶ Layout and style your website using Bootstrap
- ▶ Bootstrap Components

# WHAT IS BOOTSTRAP?

---

- Bootstrap is a free front-end framework for faster and easier web development
- Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins
- Bootstrap also gives you the ability to easily create responsive designs

What is Responsive Web Design?

Responsive web design is about creating web sites which automatically adjust themselves to look good on all devices, from small phones to large desktops.

# WHAT IS BOOTSTRAP?

---

Get Bootstrap 5:

There are two methods to use Bootstrap:

1. Download Bootstrap 5 from [getbootstrap.com](http://getbootstrap.com).
2. Include Bootstrap 5 from a CDN (Content Delivery Network): Adding these two line is you <head> will give you access to Bootstrap v5.3.

**<!-- Latest compiled and minified CSS -->**

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
```

**<!-- Latest compiled JavaScript -->**

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
```

## 5 LAYOUT AND STYLE YOUR WEBSITE USING BOOTSTRAP

---

### 1. Containers:

- `.container` class: The `.container` class provides a responsive fixed width container
- `.container-fluid` class: The `.container-fluid` class provides a full width container, spanning the entire width of the viewport

Below image is for your reference to understand the difference between `.container` and `.container-fluid` class.

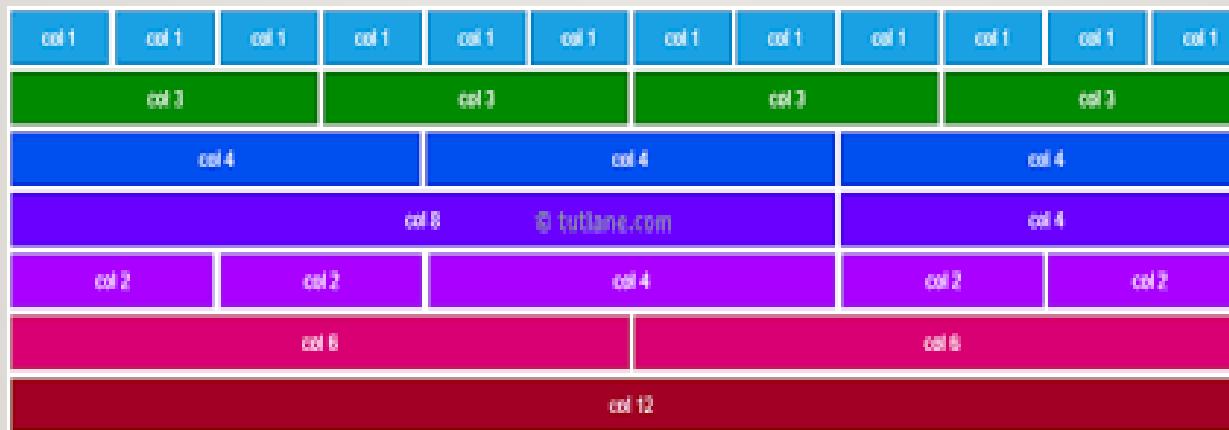


## 6 LAYOUT AND STYLE YOUR WEBSITE USING BOOTSTRAP

---

### 2. Grid:

- Bootstrap's grid system is built with flexbox and allows up to 12 columns across the page.
- The grid system is responsive, and the columns will re-arrange automatically depending on the screen size.
- Make sure that the sum adds up to 12 or fewer (it is not required that you use all 12 available columns).



# 7

# LAYOUT AND STYLE YOUR WEBSITE USING BOOTSTRAP

---

## 2. Grid:

The Bootstrap 5 grid system has six classes:

- .col- (extra small devices - screen width less than 576px)
- .col-sm- (small devices - screen width equal to or greater than 576px)
- .col-md- (medium devices - screen width equal to or greater than 768px)
- .col-lg- (large devices - screen width equal to or greater than 992px)
- .col-xl- (xlarge devices - screen width equal to or greater than 1200px)
- .col-xxl- (xxlarge devices - screen width equal to or greater than 1400px)

```
<div class="row">
    <div class="col-sm-3">.col-sm-3</div>
    <div class="col-sm-3">.col-sm-3</div>
    <div class="col-sm-3">.col-sm-3</div>
    <div class="col-sm-3">.col-sm-3</div>
</div>
```

*Output:*

.col-sm-3

.col-sm-3

.col-sm-3

.col-sm-3

# 8 LAYOUT AND STYLE YOUR WEBSITE USING BOOTSTRAP

---

## 3. Breakpoints:

- Breakpoints are the building blocks of responsive design. Use them to control when your layout can be adapted at a particular viewport or device size.
- Mobile first, responsive design is the goal. Bootstrap's CSS aims to apply the bare minimum of styles to make a layout work at the smallest breakpoint, and then layers on styles to adjust that design for larger devices. This optimizes your CSS, improves rendering time, and provides a great experience for your visitors.

The following tables shows the break points.

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

# 9 LAYOUT AND STYLE YOUR WEBSITE USING BOOTSTRAP

---

## 3. Gutters

- Gutters are the padding between your columns, used to responsively space and align content in the Bootstrap grid system.
- Gutters are the gaps between column content, created by horizontal padding. We set padding-right and padding-left on each column, and use negative margin to offset that at the start and end of each row to align content.

Custom column padding

Custom column padding

```
<div class="container px-4">
  <div class="row gx-5">
    <div class="col">
      <div class="p-3 border bg-light">Custom column padding</div>
    </div>
    <div class="col">
      <div class="p-3 border bg-light">Custom column padding</div>
    </div>
  </div>
</div>
```

# COMPONENTS:

## 1. Alerts:

Provide contextual feedback messages for typical user actions with the handful of available and flexible alert messages.

Examples: Alerts are available for any length of text, as well as an optional close button. For proper styling, use one of the eight required contextual classes (e.g., .alert-success).

```
<div class="alert alert-primary" role="alert">  
  A simple primary alert—check it out!  
</div>  
<div class="alert alert-secondary" role="alert">  
  A simple secondary alert—check it out!  
</div>  
<div class="alert alert-success" role="alert">  
  A simple success alert—check it out!  
</div>  
<div class="alert alert-danger" role="alert">  
  A simple danger alert—check it out!  
</div>  
<div class="alert alert-warning" role="alert">  
  A simple warning alert—check it out!  
</div>  
<div class="alert alert-info" role="alert">  
  A simple info alert—check it out!  
</div>  
<div class="alert alert-light" role="alert">  
  A simple light alert—check it out!  
</div>  
<div class="alert alert-dark" role="alert">  
  A simple dark alert—check it out!  
</div>
```

# COMPONENTS:

---

## 1. Alerts:

Provide contextual feedback messages for typical user actions with the handful of available and flexible alert messages.

Examples: Alerts are available for any length of text, as well as an optional close button. For proper styling, use one of the eight required contextual classes (e.g., .alert-success).

A simple primary alert—check it out!

A simple secondary alert—check it out!

A simple success alert—check it out!

A simple danger alert—check it out!

A simple warning alert—check it out!

A simple info alert—check it out!

A simple light alert—check it out!

A simple dark alert—check it out!

# COMPONENTS:

---

## 2. Badge

- Badges is for small count and labeling component.
- Badges scale to match the size of the immediate parent element by using relative font sizing and em units.

Example:

Headings

Example heading New

```
<h1>Example heading <span class="badge bg-secondary">New</span></h1>
<h2>Example heading <span class="badge bg-secondary">New</span></h2>
<h3>Example heading <span class="badge bg-secondary">New</span></h3>
<h4>Example heading <span class="badge bg-secondary">New</span></h4>
<h5>Example heading <span class="badge bg-secondary">New</span></h5>
<h6>Example heading <span class="badge bg-secondary">New</span></h6>
```

# COMPONENTS:

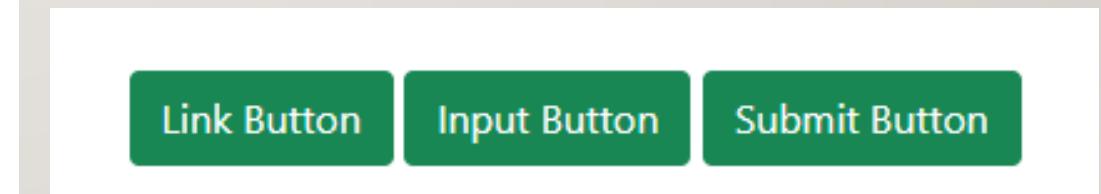
## 3.Buttons:

- Use Bootstrap's custom button styles for actions in forms, dialogs, and more with support for multiple sizes, states, and more. Button class can also be used on `<a>` and `<input>` elements.



```
<button type="button" class="btn">Basic</button>
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-dark">Dark</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-link">Link</button>
```

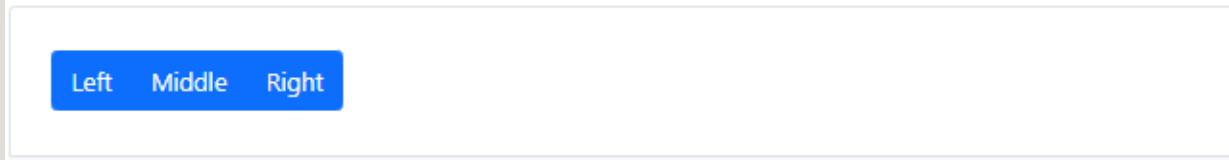
```
<a href="#" class="btn btn-success">Link Button</a>
<input type="button" class="btn btn-success" value="Input Button">
<input type="submit" class="btn btn-success" value="Submit Button">
```



## COMPONENTS:

---

4. Button group: Group a series of buttons together on a single line or stack them in a vertical column.



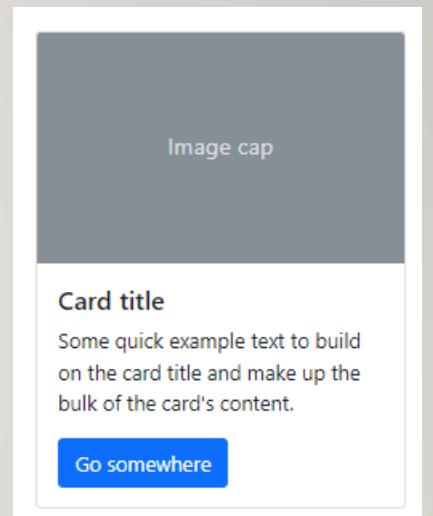
```
<div class="btn-group" role="group" aria-label="Basic example">
  <button type="button" class="btn btn-primary">Left</button>
  <button type="button" class="btn btn-primary">Middle</button>
  <button type="button" class="btn btn-primary">Right</button>
</div>
```

# COMPONENTS:

## 5. Card

- Bootstrap's cards provide a flexible and extensible content container with multiple variants and options.
- Example:
  - Below is an example of a basic card with mixed content and a fixed width. Cards have no fixed width to start, so they'll naturally fill the full width of its parent element. This is easily customized with our various sizing options.

```
<div class="card" style="width: 18rem;">
  
  <div class="card-body">
    <h5 class="card-title">Card title</h5>
    <p class="card-text">Some quick example text to build on the card title and make up the bulk of the card's content.</p>
    <a href="#" class="btn btn-primary">Go somewhere</a>
  </div>
</div>
```



## COMPONENTS:

---

### 6. Carousel Slider

: A slideshow component for cycling through elements—images or slides of text—like a carousel.

## COMPONENTS:

---

### 8. Collapse:

- The collapse is used to show and hide content.
- Buttons or anchors are used as triggers that are mapped to specific elements you toggle.
- Collapsing an element will animate the height from its current value to 0.
- Given how CSS handles animations, you cannot use padding on a .collapse element. Instead, use the class as an independent wrapping element.

For examples refer the [documentation](#).

### 9. Dropdowns

- Dropdowns are toggleable, contextual overlays for displaying lists of links and more.
- They're made interactive with the included Bootstrap dropdown JavaScript plugin.
- They're toggled by clicking, not by hovering; this is an intentional design decision.

## COMPONENTS:

10. List group: List groups are a flexible and powerful component for displaying a series of content. Modify and extend them to support just about any content within.

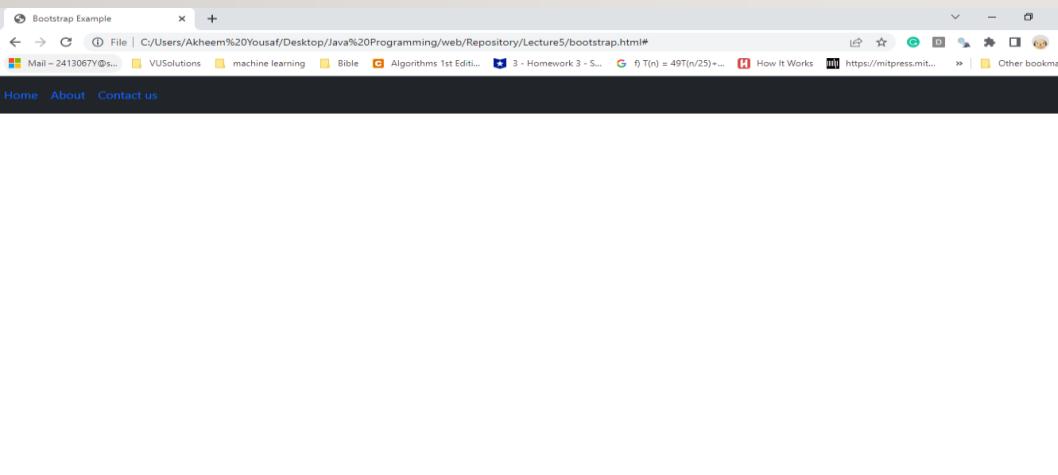


```
<ul class="list-group">
  <li class="list-group-item">An item</li>
  <li class="list-group-item">A second item</li>
  <li class="list-group-item">A third item</li>
  <li class="list-group-item">A fourth item</li>
  <li class="list-group-item">And a fifth one</li>
</ul>
```

# COMPONENTS:

## 11. Navbar:

A navigation bar (also called a Navbar) is a user interface element within a webpage that contains links to other sections of the website.



```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">  
  <div class="container-fluid">  
    <!-- Links -->  
    <ul class="navbar-nav">  
      <li class="nav-item">  
        <a class="nav-link" href="#">Home</a>  
      </li>  
      <li class="nav-item">  
        <a class="nav-link" href="#">About</a>  
      </li>  
      <li class="nav-item">  
        <a class="nav-link" href="#">Contact us</a>  
      </li>  
    </ul>  
  </div>  
</nav>
```

# COMPONENTS:

---

## 12. Pagination

- Pagination is to indicate a series of related content exists across multiple pages.
- We use a large block of connected links for our pagination, making links hard to miss and easily scalable, all while providing large hit areas.
- Pagination is built with list HTML elements so screen readers can announce the number of available links.
- Use a wrapping `<nav>` element to identify it as a navigation section to screen readers and other assistive technologies.

Example:

```
<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>
```

Previous 1 2 3 Next

# COMPONENTS:

---

## 13. Progress

- Progress components are built with two HTML elements, some CSS to set the width, and a few attributes.
- We use the `.progress` as a wrapper to indicate the max value of the progress bar.
- We use the inner `.progress-bar` to indicate the progress so far.
- The `.progress-bar` requires an inline style, utility class, or custom CSS to set their width.
- The `.progress-bar` also requires some role and aria attributes to make it accessible.

Put that all together, and you have the following examples.



# COMPONENTS:

## 13. Progress

Put that all together, and you have the following examples.

```
<div class="progress">
  <div class="progress-bar" role="progressbar" aria-valuenow="0"
aria-valuemin="0" aria-valuemax="100"></div>
</div>
<div class="progress">
  <div class="progress-bar" role="progressbar" style="width: 25%"
aria-valuenow="25" aria-valuemin="0" aria-valuemax="100"></div>
</div>
<div class="progress">
  <div class="progress-bar" role="progressbar" style="width: 50%"
aria-valuenow="50" aria-valuemin="0" aria-valuemax="100"></div>
</div>
<div class="progress">
  <div class="progress-bar" role="progressbar" style="width: 75%"
aria-valuenow="75" aria-valuemin="0" aria-valuemax="100"></div>
</div>
<div class="progress">
  <div class="progress-bar" role="progressbar" style="width: 100%"
aria-valuenow="100" aria-valuemin="0" aria-valuemax="100"></div>
</div>
```

Adding labels:

- Add labels to your progress bars by placing text within the .progress-bar.



```
<div class="progress">
  <div class="progress-bar" role="progressbar" style="width: 25%;"
aria-valuenow="25" aria-valuemin="0" aria-valuemax="100">25%</div>
</div>
```

# COMPONENTS:

---

## 14. Spinners

- Indicate the loading state of a component or page with Bootstrap spinners.
- Their appearance, alignment, and sizing can be easily customized with our amazing utility classes.
- For accessibility purposes, each loader includes role="status" and a nested <span class="visually-hidden">Loading...</span>.

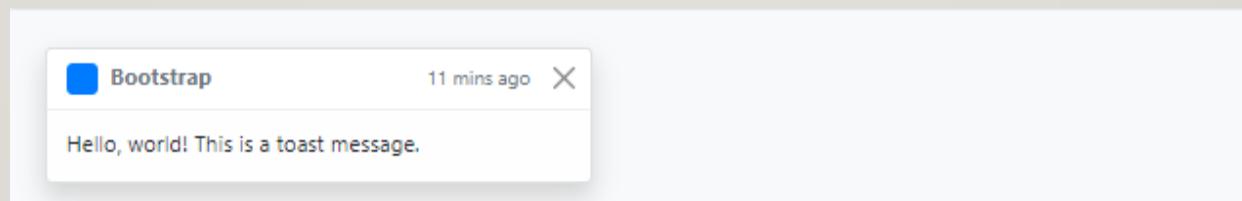
```
<div class="spinner-border" role="status">
    <span class="visually-hidden">Loading...</span>
</div>
```

# COMPONENTS:

---

## 15. Toasts

- Push notifications to your visitors with a toast, a lightweight and easily customizable alert message.
- Toasts are lightweight notifications designed to mimic the push notifications that have been popularized by mobile and desktop operating systems.
- Toasts are opt-in for performance reasons, so you must initialize them yourself.
- Toasts will automatically hide if you do not specify autohide: false.
- To encourage extensible and predictable toasts, we recommend a header and body. Toast headers use display: flex, allowing easy alignment of content thanks to our margin and flexbox utilities.
- Toasts are as flexible as you need and have very little required markup. At a minimum, we require a single element to contain your “toasted” content and strongly encourage a dismiss button.



# COMPONENTS:

---

## 15. Toasts

```
div class="toast" role="alert" aria-live="assertive" aria-atomic="true">
  <div class="toast-header">
    
    <strong class="me-auto">Bootstrap</strong>
    <small>11 mins ago</small>
    <button type="button" class="btn-close" data-bs-dismiss="toast"
aria-label="Close"></button>
  </div>
  <div class="toast-body">
    Hello, world! This is a toast message.
  </div>
</div>
```

# QUESTIONS ?