

Bootstrap

Introduction to Bootstrap

Definition:

Bootstrap is a popular front-end CSS framework used to create responsive, mobile-first websites quickly.

Why Bootstrap is Needed

- Saves time with pre-built CSS & components
- Ensures responsive design (desktop, tablet, mobile)
- Standardized grid system and UI components
- Works well with JS frameworks like React & Angular

Key Features

1. Responsive Grid System - 12-column layout
2. Pre-styled Components - buttons, navbars, forms, modals
3. CSS Utilities - spacing, colors, display
4. JavaScript Plugins - carousel, dropdowns, tooltips
5. Customizable - can override styles easily

Getting Started

Include Bootstrap via CDN

```
<!-- CSS -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">

<!-- JS Bundle -->
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
```

Example: Responsive Grid

```
<div class="container">
  <div class="row">
    <div class="col-md-6 col-sm-12">Column 1</div>
    <div class="col-md-6 col-sm-12">Column 2</div>
  </div>
</div>


- col-md-6 → half width on medium devices
- col-sm-12 → full width on small devices

```

Example: Button & Card

```
<button class="btn btn-primary">Click Me</button>

<div class="card" style="width: 18rem;">
  <div class="card-body">
    <h5 class="card-title">Card Title</h5>
    <p class="card-text">Some example text.</p>
    <a href="#" class="btn btn-success">Go</a>
  </div>
</div>
```

Real-Life Example

- Responsive e-commerce page
- Admin dashboard
- Landing pages and portfolios

Interview-Ready Lines

- Bootstrap → fast, responsive, mobile-first framework
- Uses 12-column grid for layouts
- Provides ready-to-use components & JS plugins

Bootstrap Setup Using CDN

Definition:

CDN (Content Delivery Network) allows you to include Bootstrap CSS & JS files from the internet without downloading them locally.

Why Use CDN

- Quick setup, no local files required
- Latest version automatically available
- Faster load times due to caching on multiple servers
- Easy to start learning and building

Steps to Setup Bootstrap via CDN

1. Include CSS

Place this in <head>:

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
```

2. Include JS Bundle

Place this before </body>:

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
    • Includes Popper.js for dropdowns, tooltips, modals
```

3. Basic HTML Template

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Bootstrap CDN Example</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
    <div class="container mt-5">
        <button class="btn btn-primary">Click Me</button>
    </div>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
```

```
</body>  
</html>
```

Real-Life Example

- Quick landing page prototypes
- Learning Bootstrap components without setup
- Rapid development of test projects

Interview-Ready Lines

- CDN setup → fast, easy, always updated
- CSS link → for styling, JS bundle → for components like dropdown & modal
- Ideal for small projects or learning purposes

Bootstrap Layout & Grid System

Definition:

Bootstrap Grid System is a responsive layout structure that uses rows and columns to arrange content on different screen sizes.

Why Grid System is Needed

- Create responsive, mobile-first layouts
- Align content in columns without custom CSS
- Maintain consistency across devices

Key Concepts

Container

- Holds the grid system
- Types:
 - .container → fixed width
 - .container-fluid → full width

Row

- .row → wraps columns
- Ensures horizontal alignment

Column

- .col- → divides row into columns
- Bootstrap uses 12-column layout

Column Classes

Class	Screen Size	Description
.col-	Extra small (<576px)	Default
.col-sm-	Small (≥576px)	Small devices
.col-md-	Medium (≥768px)	Tablets
.col-lg-	Large (≥992px)	Desktop
.col-xl-	Extra large (≥1200px)	Large desktop
.col-xxl-	Extra extra large (≥1400px)	Wider screens

Basic Example

```
<div class="container">  
  <div class="row">  
    <div class="col-md-6 col-sm-12">Column 1</div>
```

```
<div class="col-md-6 col-sm-12">Column 2</div>
</div>
</div>


- On small screens → columns stack vertically
- On medium+ screens → columns appear side by side

```

Responsive Example

```
<div class="row">
  <div class="col-12 col-md-4">Col 1</div>
  <div class="col-12 col-md-4">Col 2</div>
  <div class="col-12 col-md-4">Col 3</div>
</div>


- .col-12 → full width on mobile
- .col-md-4 → 1/3 width on tablets & desktop

```

Real-Life Example

- Blog layout → sidebar + main content
- E-commerce → product cards in rows
- Admin dashboards → panels & widgets

Interview-Ready Lines

- Bootstrap grid → **12-column, responsive**
- .container holds .row, rows hold .col-*
- Use different column classes for **mobile-first design**

Bootstrap Layout: Container, Row & Column

Definition:

- **Container** → Wraps the entire content and provides **responsive padding**.
- **Row** → Horizontal grouping of columns; ensures **columns align properly**.
- **Column** → Divides the row into **up to 12 parts**; holds actual content.

Why They Are Needed

- Maintain **responsive structure** across devices.
- Avoid **manual CSS for layout**.
- Ensures **grid consistency** for professional web pages.

Key Concepts

Container

- .container → fixed-width, changes with screen size.
- .container-fluid → full-width, spans entire viewport.

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

Row

- .row → wraps columns and applies **negative margins** for alignment.
- Must be **direct child of container**.

```
<div class="container">
  <div class="row">
    <!-- columns go here -->
  </div>
```

```
</div>
```

Column

- .col-* → divide row into 12 equal parts.
- Classes: .col-, .col-sm-, .col-md-, .col-lg-, .col-xl-, .col-xxl-.

```
<div class="row">
  <div class="col-md-6">Column 1</div>
  <div class="col-md-6">Column 2</div>
</div>
```

- On medium screens → each column 50% width.
- On small screens → columns stack (full width).

Technical / IT Example

```
<div class="container">
  <div class="row">
    <div class="col-md-4 col-sm-12">Card 1</div>
    <div class="col-md-4 col-sm-12">Card 2</div>
    <div class="col-md-4 col-sm-12">Card 3</div>
  </div>
</div>
```

- Small screens → columns stack vertically
- Medium+ screens → columns side by side

Interview-Ready Points

- Container → main wrapper
- Row → horizontal alignment of columns
- Column → holds content, responsive width using **12-grid system**
- Best Practice → Always put columns inside rows, rows inside containers

Bootstrap: Responsive Breakpoints & Essential Components

Responsive Breakpoints

Definition:

Breakpoints define screen widths where the layout adapts automatically to different devices.

Bootstrap 5 Breakpoints

Class Prefix Screen Size Description

.col-	<576px	Extra small devices (mobile)
.col-sm-	≥576px	Small devices (phones)
.col-md-	≥768px	Medium devices (tablets)
.col-lg-	≥992px	Large devices (desktops)
.col-xl-	≥1200px	Extra large devices
.col-xxl-	≥1400px	Extra extra large devices

Real-Life Example

```
<div class="row">
  <div class="col-12 col-md-6 col-lg-4">Column 1</div>
  <div class="col-12 col-md-6 col-lg-4">Column 2</div>
  <div class="col-12 col-md-12 col-lg-4">Column 3</div>
```

```
</div>
  • Mobile → 1 column per row
  • Tablet → 2 columns per row
  • Desktop → 3 columns per row
```

Interview-Ready Points

- Mobile-first approach
 - Use **breakpoint prefixes** to control layout
 - Ensures **consistent UI across devices**
-

Essential Bootstrap Components

1. Buttons

```
<button class="btn btn-primary">Primary</button>
<button class="btn btn-success">Success</button>
<button class="btn btn-danger">Danger</button>
  • Classes: btn, btn-primary, btn-secondary, btn-success, etc.
```

2. Navbar

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Logo</a>
  <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navMenu">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navMenu">
    <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>
    </ul>
  </div>
</nav>
  • Responsive, collapses on small screens
```

3. Cards

```
<div class="card" style="width: 18rem;">
  
  <div class="card-body">
    <h5 class="card-title">Card Title</h5>
    <p class="card-text">Some text.</p>
    <a href="#" class="btn btn-primary">Go</a>
  </div>
</div>
  • Used for displaying content in blocks
```

4. Modals

```
<!-- Button trigger modal -->
<button class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#exampleModal">Open Modal</button>

<!-- Modal -->
<div class="modal fade" id="exampleModal">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title">Modal title</h5>
```

```
<button type="button" class="btn-close" data-bs-dismiss="modal"></button>
</div>
<div class="modal-body">Modal content here.</div>
</div>
</div>
• Overlays content, great for alerts/forms
```

5. Alerts

```
<div class="alert alert-success" role="alert">
  Success! Your action was completed.
</div>
• Classes: alert-success, alert-warning, alert-danger
```

Real-Life Example

- Buttons → Submit, Cancel
 - Navbar → Website menu
 - Cards → Product showcase
 - Modals → Login/Signup forms
 - Alerts → Notifications/messages
-

Interview-Ready Points

- Bootstrap is **mobile-first** → use breakpoints
 - Components → **pre-styled & responsive**
 - **Cards, Navbar, Modals, Buttons** → most commonly used in real projects
-
-

Bootstrap: Typography, Buttons, Cards & Forms

Typography

Definition:

Bootstrap provides **predefined text styles** for headings, paragraphs, and inline text.

Key Classes

Class	Description
.h1 to .h6	Headings
.lead	Larger paragraph text
.text-primary, .text-success	Colored text
.text-center, .text-end	Text alignment
.fw-bold, .fw-light	Font weight
.fst-italic	Italic text

Example

```
<h1 class="text-primary">Heading 1</h1>
<p class="lead">This is a lead paragraph.</p>
<p class="text-success text-center">Success message</p>
```

Buttons

Definition:

Pre-styled **buttons** with colors, sizes, and actions.

Key Classes

Class	Description
.btn	Base button
.btn-primary, .btn-success	Button colors
.btn-lg, .btn-sm	Button sizes
.btn-outline-*	Outline style

Example

```
<button class="btn btn-primary btn-lg">Primary Large</button>
<button class="btn btn-outline-success btn-sm">Success Small</button>
```

Cards

Definition:

Cards are flexible content containers with **header**, **body**, and **footer**.

Key Classes

Class	Description
.card	Main container
.card-body	Content area
.card-title	Title
.card-text	Text inside card
.card-img-top	Image at top
.btn inside .card	Button inside card

Example

```
<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.</p>
    <a href="#" class="btn btn-primary">Go</a>
  </div>
</div>
```

Forms

Definition:

Bootstrap forms are **pre-styled input fields, labels, checkboxes, radios, selects, etc.**

Key Classes

Class	Description
.form-control	Input, textarea, select
.form-label	Label for inputs
.form-check	Checkbox / radio container
.form-check-input	Checkbox / radio input
.form-check-label	Label for checkbox / radio
.btn	Submit button

Example

```
<form>
```

```

<div class="mb-3">
  <label for="name" class="form-label">Name</label>
  <input type="text" class="form-control" id="name" placeholder="Enter name">
</div>
<div class="mb-3">
  <label for="email" class="form-label">Email</label>
  <input type="email" class="form-control" id="email" placeholder="Enter email">
</div>
<div class="form-check">
  <input class="form-check-input" type="checkbox" id="check">
  <label class="form-check-label" for="check">Accept Terms</label>
</div>
<button type="submit" class="btn btn-primary mt-3">Submit</button>
</form>

```

Real-Life Example

- **Typography** → headings & messages on websites
- **Buttons** → Submit, Cancel, Navigation
- **Cards** → Product listings, profile info
- **Forms** → Login, signup, contact forms

Interview-Ready Lines

- Bootstrap provides **pre-styled components** for fast development
- **Cards** → flexible content blocks
- **Forms** → consistent input styles and validation
- Use **classes** like `.btn`, `.form-control`, `.card` for professional UI

Bootstrap: Navbar, Modals & Dropdowns

Navbar

Definition:

A **Navbar** is a responsive **navigation header** for your website.

Key Classes

Class	Description
<code>.navbar</code>	Main navbar container
<code>.navbar-expand-lg</code>	Expand navbar on large screens
<code>.navbar-light</code> / <code>.navbar-dark</code>	Text color theme
<code>.bg-light</code> / <code>.bg-dark</code>	Background color
<code>.navbar-brand</code>	Logo or brand name
<code>.navbar-nav</code>	Container for menu items
<code>.nav-item</code>	Menu item
<code>.nav-link</code>	Link inside navbar
<code>.navbar-toggler</code>	Button for collapse on small screens
<code>.collapse</code>	Collapsible menu container

Example

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Brand</a>
```

```

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navMenu">
    <span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navMenu">
    <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</a></li>
    </ul>
</div>
</nav>

```

Modals

Definition:

A **Modal** is a **popup dialog** to display content or forms above the main page.

Key Classes

Class	Description
.modal	Main modal container
.modal-dialog	Wrapper for modal content
.modal-content	Actual content container
.modal-header	Header area (title + close button)
.modal-body	Body content
.modal-footer	Footer (buttons)
data-bs-toggle="modal"	Triggers modal
data-bs-target="#id"	Target modal by ID

Example

```

<!-- Button trigger -->
<button class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#exampleModal">Open Modal</button>

<!-- Modal -->
<div class="modal fade" id="exampleModal">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title">Modal Title</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
            </div>
            <div class="modal-body">Modal content goes here.</div>
            <div class="modal-footer">
                <button class="btn btn-secondary" data-bs-dismiss="modal">Close</button>
                <button class="btn btn-primary">Save changes</button>
            </div>
        </div>
    </div>
</div>

```

Dropdowns

Definition:

Dropdowns are toggleable menus for **selecting options** or **navigation links**.

Key Classes

Class	Description
.dropdown	Container
.dropdown-toggle	Button or link that triggers menu
data-bs-toggle="dropdown"	Activate dropdown
.dropdown-menu	Menu container
.dropdown-item	Menu items

Example

```
<div class="dropdown">
  <button class="btn btn-secondary dropdown-toggle" type="button" data-bs-
  toggle="dropdown">
    Select Option
  </button>
  <ul class="dropdown-menu">
    <li><a class="dropdown-item" href="#">Option 1</a></li>
    <li><a class="dropdown-item" href="#">Option 2</a></li>
    <li><a class="dropdown-item" href="#">Option 3</a></li>
  </ul>
</div>
```

Real-Life Example

- **Navbar** → Website menu
- **Modal** → Login, Signup, Alerts
- **Dropdown** → Profile menu, filters, settings

Interview-Ready Points

- Navbar → responsive menu with collapsible toggle
- Modals → popup content over main page
- Dropdowns → toggleable menus, use .dropdown-toggle & .dropdown-menu
- Always include **Bootstrap JS bundle** for modals & dropdowns to work

Bootstrap Utilities & Helpers

Definition:

Utilities & Helpers are **predefined classes** in Bootstrap that allow you to **quickly style elements** without writing custom CSS.

Why Utilities Are Needed

- Rapid development
- Maintain **consistent design**
- Avoid repetitive CSS
- Easy to **adjust spacing, alignment, colors** on the fly

Key Utility Categories

1. Spacing (Margin & Padding)

Class	Description
m-*	Margin
p-*	Padding
m-0, p-0	Remove margin/padding
m-1...m-5	Small to large margin
mt-3, mb-2, px-4	Top, bottom, horizontal spacing

Example:

```
<div class="p-3 mb-2 bg-light">Content with padding & margin</div>
```

2. Text & Background

Class	Description
.text-primary, .text-success	Text colors
.text-center, .text-end	Text alignment
.bg-primary, .bg-light	Background colors
.fw-bold, .fst-italic	Font weight / style

Example:

```
<p class="text-success fw-bold">Success message</p>
<div class="bg-primary text-white p-2">Blue background</div>
```

3. Display & Visibility

Class	Description
.d-none	Hide element
.d-block, .d-inline, .d-inline-block	Change display type
.d-sm-block, .d-md-none	Responsive visibility

Example:

```
<div class="d-none d-md-block">Visible only on medium+ screens</div>
```

4. Flex Utilities

Class	Description
.d-flex	Display flex
.justify-content-center	Center items horizontally
.align-items-center	Center items vertically
.flex-column	Stack items vertically

Example:

```
<div class="d-flex justify-content-center align-items-center" style="height:100px;">
  <button class="btn btn-primary">Centered Button</button>
</div>
```

5. Sizing Utilities

Class	Description
.w-25, .w-50, .w-75, .w-100	Width percentages
.h-25, .h-50, .h-75, .h-100	Height percentages
.mw-100, .mh-100	Max width / height 100%

Example:

```

```

Real-Life Example

- Spacing → Adjust margins for cards
- Text & background → Highlight alerts or headings

- **Display & flex** → Responsive navigation, centering buttons
- **Sizing** → Images, containers

Interview-Ready Lines

- Utilities = predefined helper classes for spacing, text, display, flex, sizing
- Saves time, avoids custom CSS
- Mobile-first → most utilities support breakpoints

Responsive Design with Bootstrap

Definition:

Responsive design ensures a website adapts automatically to different screen sizes: mobile, tablet, desktop, and large screens. Bootstrap makes it easy to implement responsiveness using its grid system, breakpoints, and utilities.

Why Responsive Design Is Needed

- Improve user experience across devices
- Avoid horizontal scrolling on mobile
- Essential for modern web development
- Improves SEO and accessibility

Key Features in Bootstrap for Responsive Design

1. Grid System & Breakpoints

- 12-column grid divides the page
- Breakpoints adjust layout per device

Prefix Screen Size Example

.col-	<576px	Extra small devices
.col-sm-	≥576px	Small devices
.col-md-	≥768px	Medium devices
.col-lg-	≥992px	Large devices
.col-xl-	≥1200px	Extra large
.col-xxl-	≥1400px	Extra extra large

```
<div class="row">
  <div class="col-12 col-md-6">Column 1</div>
  <div class="col-12 col-md-6">Column 2</div>
</div>
```

2. Responsive Utilities

- Show/hide elements using .d-none, .d-md-block
- Adjust spacing per device using m-, p- classes with breakpoints

```
<div class="d-none d-lg-block">Visible only on large screens</div>
```

3. Responsive Images

- Use .img-fluid → makes image scale with parent

```

```

4. Navbar & Components

- Navbar → .navbar-expand-lg → collapses on small screens
- Cards, modals, buttons → adapt with flex & grid utilities

Real-Life Examples

- E-commerce website → Products stack on mobile, grid on desktop
- Portfolio website → Navbar collapses into hamburger menu
- Dashboard → Cards resize automatically

Interview-Ready Points

- Bootstrap = mobile-first framework
- Grid + Breakpoints + Utilities = responsive layout
- Responsive images → .img-fluid
- Navbar, buttons, cards → automatically adapt using Bootstrap classes

Bootstrap: Customization & Extensions

Definition:

Customization → Modifying Bootstrap's default styles to fit your project design.

Extensions → Adding extra functionality or components beyond Bootstrap's default library.

Why Customization & Extensions Are Needed

- Maintain brand identity with colors, fonts, and layout
- Avoid repetitive overrides in CSS
- Extend Bootstrap for advanced components without rewriting everything

Ways to Customize Bootstrap

1. Override CSS

- Add a custom CSS file after Bootstrap CSS
- Change variables like colors, fonts, spacing

```
<link href="bootstrap.min.css" rel="stylesheet">
<link href="custom.css" rel="stylesheet">
/* custom.css */
.btn-primary {
    background-color: #ff6600; /* Custom orange */
    border-radius: 12px;
}
```

2. Use Sass Variables

- Bootstrap is built with Sass, allowing theme customization before compilation
- Modify variables in _variables.scss

```
$primary: #ff6600; // Change primary color
$font-family-base: 'Arial, sans-serif';
• Compile Sass → generate customized Bootstrap CSS
```

3. Utility Classes & Mixins

- Use Bootstrap utility classes with custom values
- Create custom utilities for spacing, color, or flex

```
@mixin custom-margin($size) {
    margin: $size;
}
.my-div {
    @include custom-margin(20px);
}
```

Extensions & Plugins

Popular Bootstrap Extensions

Extension	Purpose
-----------	---------

Bootstrap Icons	Ready-to-use icon library
-----------------	---------------------------

Bootstrap Select	Enhanced dropdowns
------------------	--------------------

Bootstrap Datepicker	Advanced date picking UI
----------------------	--------------------------

Bootstrap Table	Responsive tables with sorting & filtering
-----------------	--

MDBootstrap	Material Design + Bootstrap components
-------------	--

Example: Bootstrap Icons

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.5/font/bootstrap-icons.css" rel="stylesheet">
<i class="bi bi-heart-fill"></i>
```

Real-Life Example

- **Brand-specific theme** → override primary color, font
 - **Enhanced UI** → date picker, icon libraries, advanced tables
 - **Custom cards** → unique style for products or portfolios
-

Interview-Ready Points

- Customize via **CSS overrides, Sass variables, and mixins**
 - Extensions add **extra functionality** (icons, tables, date pickers)
 - Enables **professional, branded, and scalable UI**
-