BOOTSTRAP

By-
Harshit Khandelwal IT35
Khushi Jaiswal IT40
Mahak Chauhan IT47

WHAT IS BOOTSTRAP?

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains HTML, CSS and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

FEATURES OF BOOTSTRAP

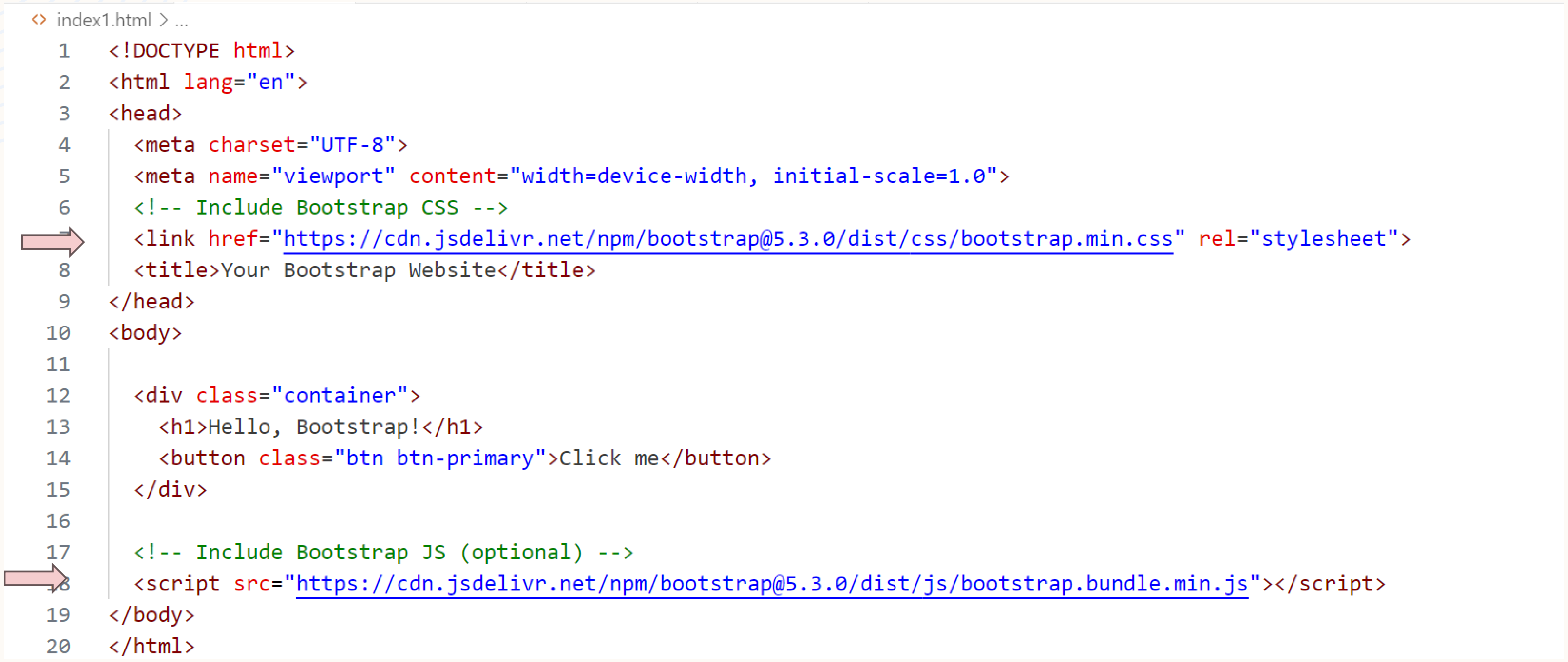
Front-end framework: bootstrap is primarily A front-end framework, providing A set of pre-designed and pre-built html, css, and javascript components. These components can be easily integrated into web projects, allowing developers to create responsive and visually appealing websites or web applications.

Responsive design: one of the main features of bootstrap is its emphasis on responsive design. The framework includes A responsive grid system and components that adapt to different screen sizes, making it easier to create websites that look good on various devices, from desktops to tablets and smartphones.

Ease of use: bootstrap is known for its ease of use and quick development capabilities. By using predefined classes and components, developers can save time and effort in creating common design elements, allowing them to focus more on the functionality and unique aspects of their projects.

Pre-designed components: bootstrap comes with A variety of pre-designed ui components such as navigation bars, buttons, forms, modals, carousels, and more. These components can be customized and combined to build consistent and aesthetically pleasing user interfaces.

Here's a simple example of how you might include Bootstrap in an HTML file:



index1.html > ...

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- Include Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
  <title>Your Bootstrap Website</title>
</head>
<body>

  <div class="container">
    <h1>Hello, Bootstrap!</h1>
    <button class="btn btn-primary">Click me</button>
  </div>

  <!-- Include Bootstrap JS (optional) -->
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

ADVANTAGES OF BOOTSTRAP

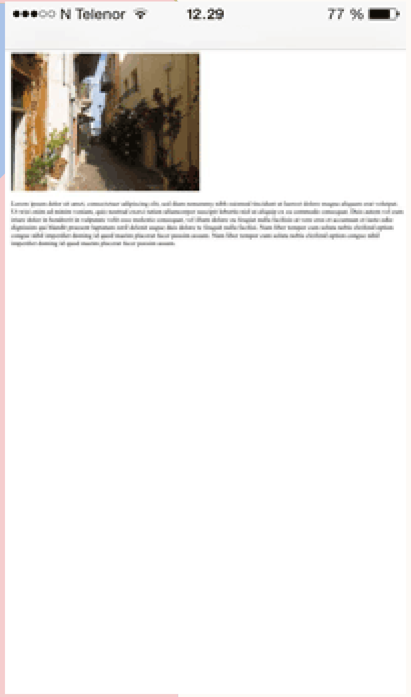
1. Fewer Cross browser bugs
2. A consistent framework that supports major of all browsers and CSS compatibility fixes
3. Lightweight and customizable
4. Responsive structures and styles
5. Several JavaScript plugins using the jQuery
6. Good documentation and community support
7. Loads of free and professional templates, WordPress themes and plugins
8. Great grid system

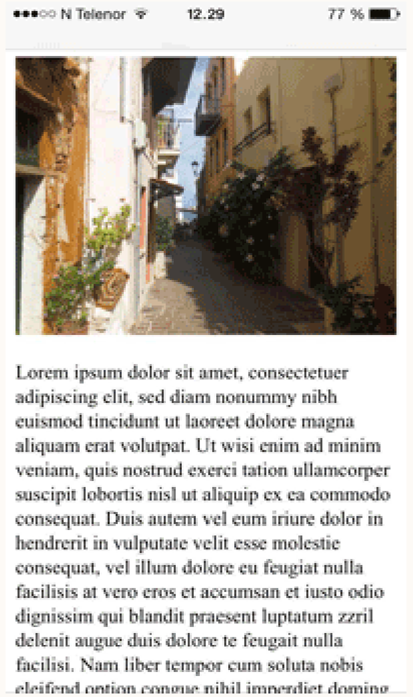
VIEWPORT

The viewport is the user's visible area of a web page.

The viewport varies with the device, and will be smaller on a mobile phone than on a computer screen.

```
<meta name="viewport" content="width=device-width, initial scale=1.0">
```





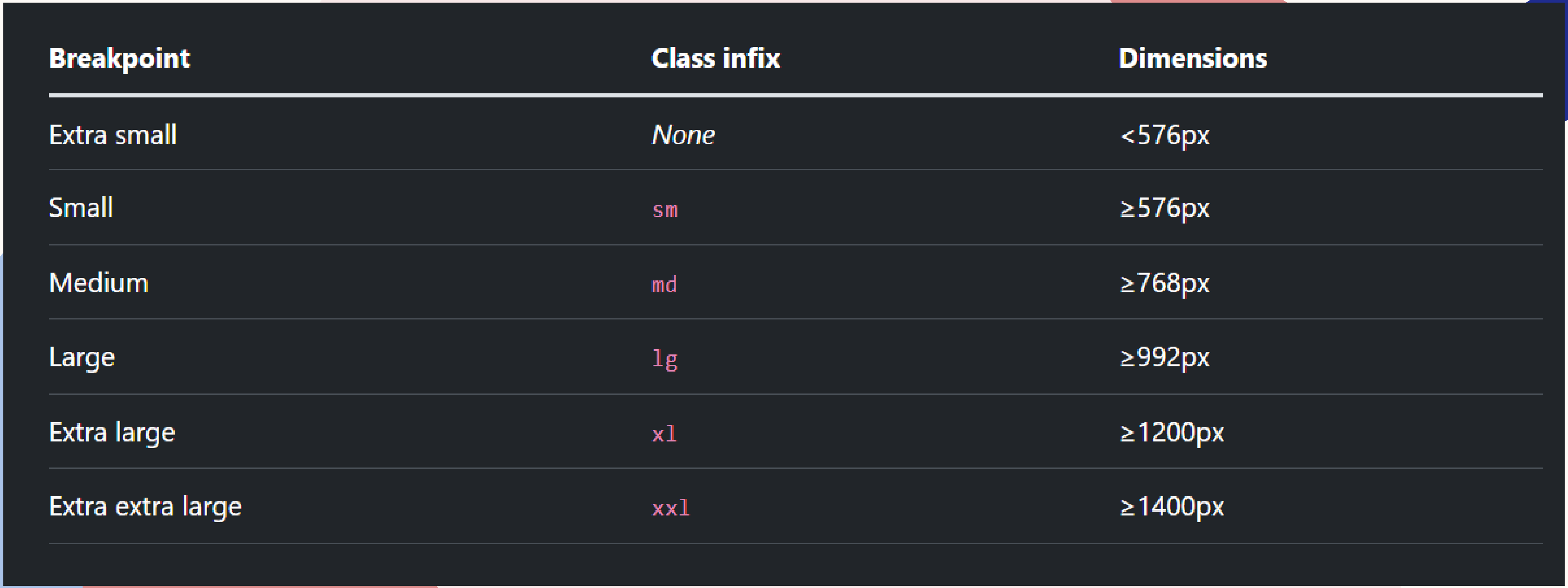
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming

BREAKPOINTS

Breakpoints are customizable widths that determine how your responsive layout behaves across device or viewport sizes in Bootstrap.

- **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.
- **Use media queries to architect your CSS by breakpoint**. Media queries are a feature of CSS that allow you to conditionally apply styles based on a set of browser and operating system parameters. We most commonly use min-width in our media queries.
- **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.

AVAILABLE BREAKPOINTS



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

CONTAINERS

Containers are a fundamental building block of Bootstrap that contain, pad, and align your content within a given device or viewport.

Containers are the most basic layout element in Bootstrap and are **required when using our default grid system**. Containers are used to contain, pad, and (sometimes) center the content within them. While containers can be nested, most layouts do not require a nested container.

Bootstrap comes with three different containers:

- **.container**, which sets a max-width at each responsive breakpoint
- **.container-{breakpoint},** which is width: 100% until the specified breakpoint
- .**container-fluid**, which is width: 100% at all breakpoints

DEFAULT CONTAINER

Our default .container class is a responsive, fixed-width container, meaning its max-width changes at each breakpoint.

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

RESPONSIVE CONTAINERS

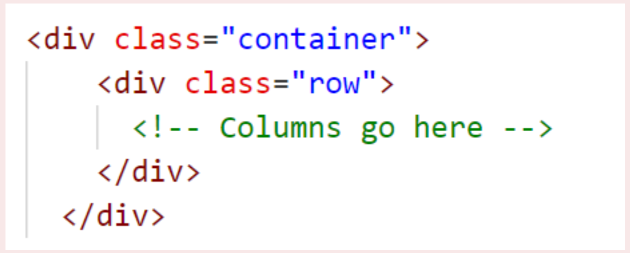
Responsive containers allow you to specify a class that is 100% wide until the specified breakpoint is reached, after which we apply max-widths for each of the higher breakpoints. For example, .container-sm is 100% wide to start until the sm breakpoint is reached, where it will scale up with md, lg, xl, and xxl.

```
<div class="container-sm">100% wide until small breakpoint</div>
<div class="container-md">100% wide until medium breakpoint</div>
<div class="container-lg">100% wide until large breakpoint</div>
<div class="container-xl">100% wide until extra large breakpoint</div>
<div class="container-xxl">100% wide until extra extra large breakpoint</div>
```

GRID SYSTEM

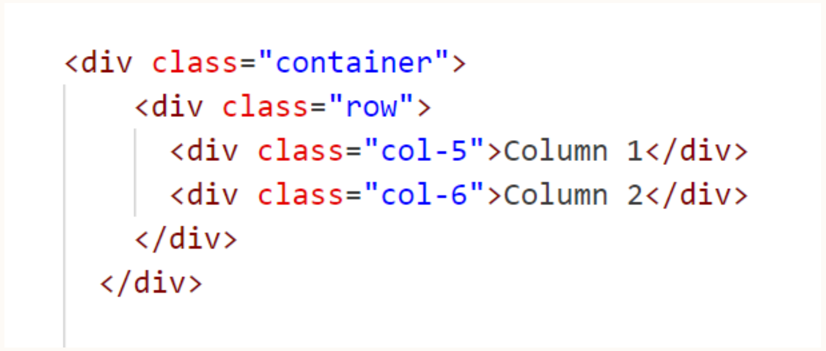
Bootstrap's grid system is a powerful layout system that allows developers to cre
responsive and flexible designs. The grid system is based on a 12-column la
used to organize content on a web page. Here are some key concepts of B
grid system:

Rows: Inside the container, you create rows using the .row class. Rows are horizontal containers for columns.

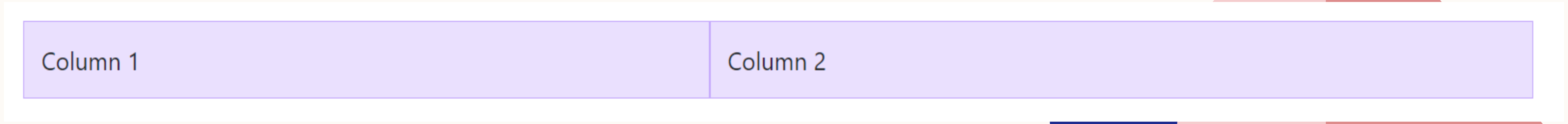


```
<div class="container">
    <div class="row">
      <!-- Columns go here -->
    </div>
  </div>
```

Columns: Columns are the building blocks of the grid system. You define the number of columns each element should span using classes like .col-, where the number represents the number of columns to span.

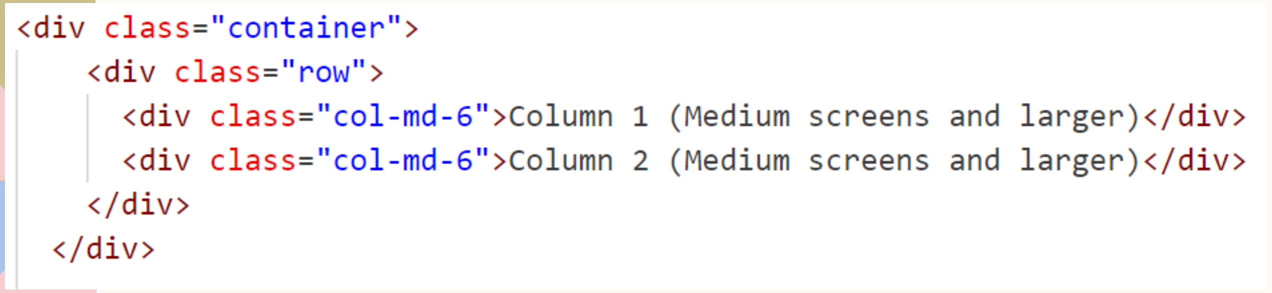


```
<div class="container">
  <div class="row">
    <div class="col-5">Column 1</div>
    <div class="col-6">Column 2</div>
  </div>
</div>
```

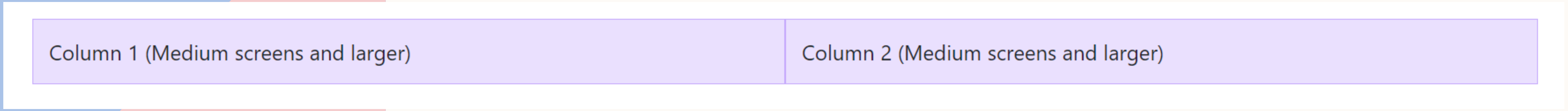


Responsive Classes:

- Bootstrap provides responsive classes to control the layout at different screen sizes.
- Classes like .col-sm-, .col-md-, .col-lg-, and .col-xl- can be used to define different column widths for small, medium, large, and extra-large screens.



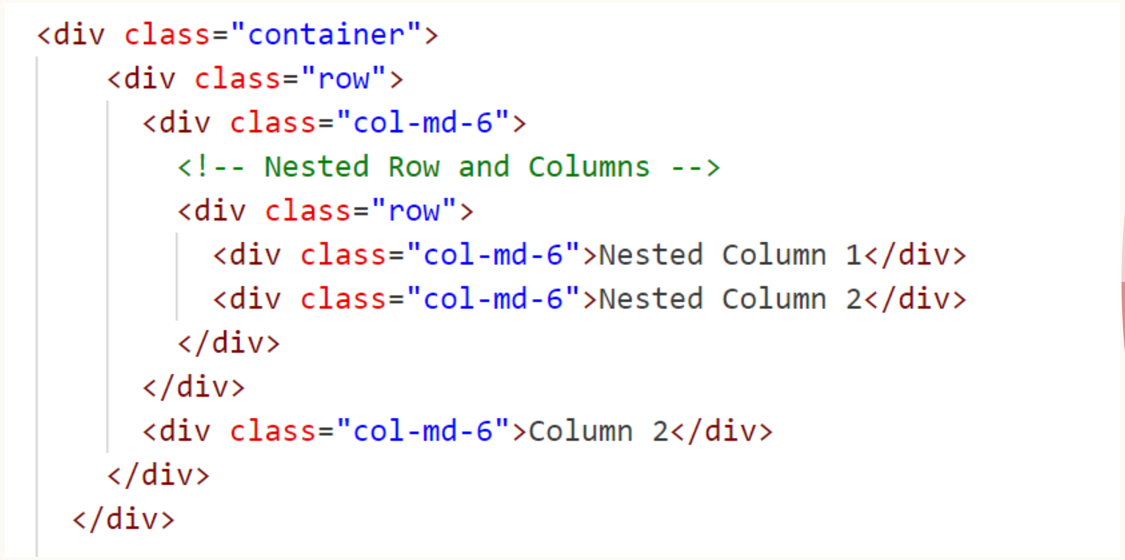
```html
<div class="container">
    <div class="row">
      <div class="col-md-6">Column 1 (Medium screens and larger)</div>
      <div class="col-md-6">Column 2 (Medium screens and larger)</div>
    </div>
  </div>
```



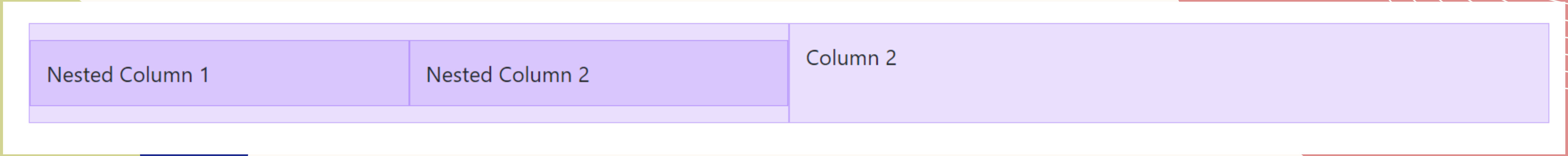
Column 1 (Medium screens and larger) | Column 2 (Medium screens and larger)

Nesting Columns:

Columns can be nested inside other columns to create more complex layouts.



```html
<div class="container">
    <div class="row">
      <div class="col-md-6">
        <!-- Nested Row and Columns -->
        <div class="row">
          <div class="col-md-6">Nested Column 1</div>
          <div class="col-md-6">Nested Column 2</div>
        </div>
      </div>
      <div class="col-md-6">Column 2</div>
    </div>
  </div>
```



FLOAT

In Bootstrap, the ‘.float’ class was used to set the alignment of element by floating it to the left or right within its containi element. However, as of Bootstrap 4, the ‘.float’ class has b replaced with the ‘.float-{breakpoint}-{direction}’ utility classes.

.float-{breakpoint}-{direction}: This class defines the float behavior at different breakpoints. ‘{breakpoint}’ can be ‘sm’, ‘md’, ‘lg’, or ‘xl’, and ‘{direction}’ can be ‘left’ or ‘right’.

Responsive variations also exist for each float value.

{breakpoint} is the minimum viewport width at which the class takes effect. **{direction}** is the direction the element should float.

```html
<div class="float-sm-left">Float left on viewports sized SM (small) or wider</div><br>
<div class="float-md-left">Float left on viewports sized MD (medium) or wider</div><br>
<div class="float-lg-left">Float left on viewports sized LG (large) or wider</div><br>
<div class="float-xl-left">Float left on viewports sized XL (extra-large) or wider</div><br>
```

All the support classes of float are:

❑ .float-left	❑ .float-sm-left	❑ .float-md-left	❑ .float-lg-left	❑ .float-xl-left
❑ .float-right	❑ .float-sm-right	❑ .float-md-right	❑ .float-lg-right	❑ .float-xl-right
❑ .float-none	❑ .float-sm-none	❑ .float-md-none	❑ .float-lg-none	❑ .float-xl-none

INTERACTIONS

Interactions is a utility class that controls how elements of a website are interacted with (how the user interacts with them). Utility classes like text selection and pointer events help to interact with the website.

TEXT SELECTION : Text selection helps us to change the way when a user tries to select the content displayed.

- **user-select-all:** When this class is active the entire text will get selected.
- **user-select-auto:** When this class is active user can select any amount of text.
- **user-select-none:** When this class is active user cannot select any text.

POINTER EVENTS : Pointer events in Interactions come with pe-none and pe-auto classes, to prevent or add element interactions.

- **pe-none:** The pe-none class is used to prevent element interaction. If you use the pe-none class inside the anchor tag, the link will be not accessible.
- **pe-auto:** This class helps to add the interactions with the pointer. Basically, this shows the default behavior. For example, if you use the pe-auto class inside the anchor tag, the link can be easily accessible.

Syntax:

```
<p>
    <a href="" class="pe-*">...</a>
    ....
</p>
```

POSITIONS

Bootstrap facilitates different Position values, i.e. the top, right, bottom, and property values which can be used to position an element. These define t separation between an HTML element and the viewport's edge.

1. **position-static:** This class when added to an element sets its position to be static.
2. **position-relative:** This class when added to an element sets its position to be relative with respect to the elements on top of it.
3. **position-absolute:** This class when added to an element sets its position relative to the closest parent and it set its position absolute with that.
4. **position-fixed:** This class when added to an element sets its position to be relative with respect to the viewport and it will stay in the same place if the page is scrolled.
5. **position-sticky:** This class when added to an element sets its position either relative or fixed with respect to the scroll position.

FLEXBOX

Bootstrap utilizes Flexbox, a CSS layout model, to create responsive and efficient designs. Flexbox provides a more efficient way to lay out, align, and distribute space among items in a container, even when their size is unknown or dynamic.

How is Flexbox integrated and used in Bootstrap ?

1. **Enabled by Default:** Flexbox is enabled by default. This means that many of the Bootstrap components and grid system are built with Flexbox.
2. **Responsive Grid System:** Bootstrap's grid system uses Flexbox to provide a series of responsive, mobile-first flexbox containers, rows, and columns for layout purposes.
3. **Utility Classes:** Bootstrap provides a wide range of utility classes that leverage Flexbox.

- **Flex Containers:** To define a flex container (d-flex, d-inline-flex).
- **Direction:** To set the direction of flex items (flex-row, flex-row-reverse, flex-column, flex-column-reverse).
- **Justify Content:** To align items horizontally (justify-content-start, justify-content-end, justify-content-center, justify-content-between, justify-content-around).
- **Align Items:** To align items vertically (align-items-start, align-items-end, align-items-center, align-items-baseline, align-items-stretch).
- **Align Self:** To individually align items (align-self-start, align-self-end, align-self-center, align-self-baseline, align-self-stretch).
- **Fill and Grow:** To control the flex item sizing (flex-fill, flex-grow-0, flex-grow-1).
- **Wrap:** To control the wrapping of items (flex-wrap, flex-nowrap, flex-wrap-reverse).

THANK YOU

QnA