**BOOTSTRAP LESSON PLAN**

**Lesson 1: What is Bootstrap**

**Bootstrap** is a free and open-source front-end framework used for building responsive and mobile-first websites and web applications. It was originally developed by Twitter and is now maintained by a community of developers.

**Installation** of Bootstrap can be done using CDN (Content Delivery Network) or downloading source file. 2

**UX vs UI**

UI – Refers to the user interface that focuses on the visual or graphic design of the website or web app.

UX – Refers to the user experience which focus on how user-friendly the website or web app is.

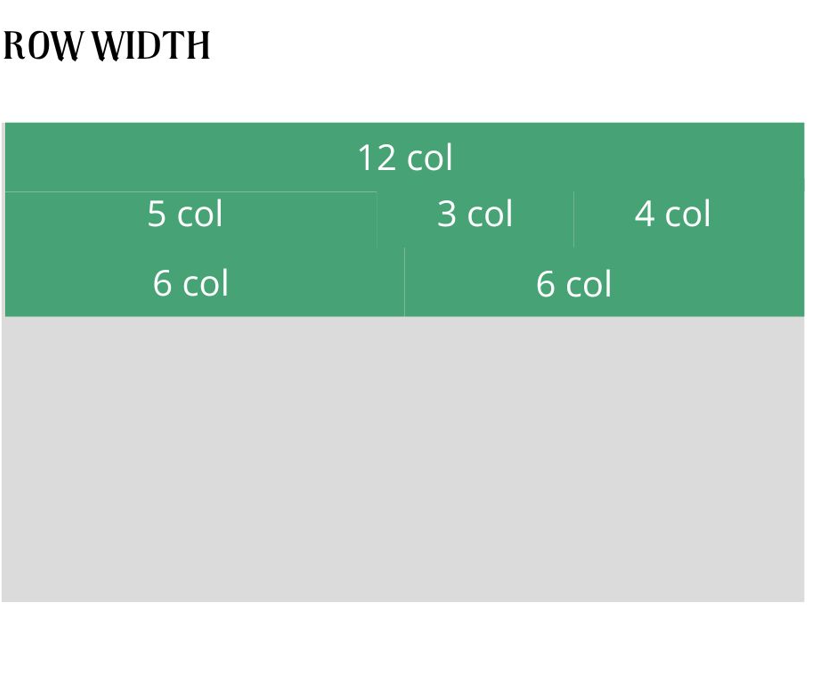
**Lesson 2: Containers and Grid System**

**Grid System**

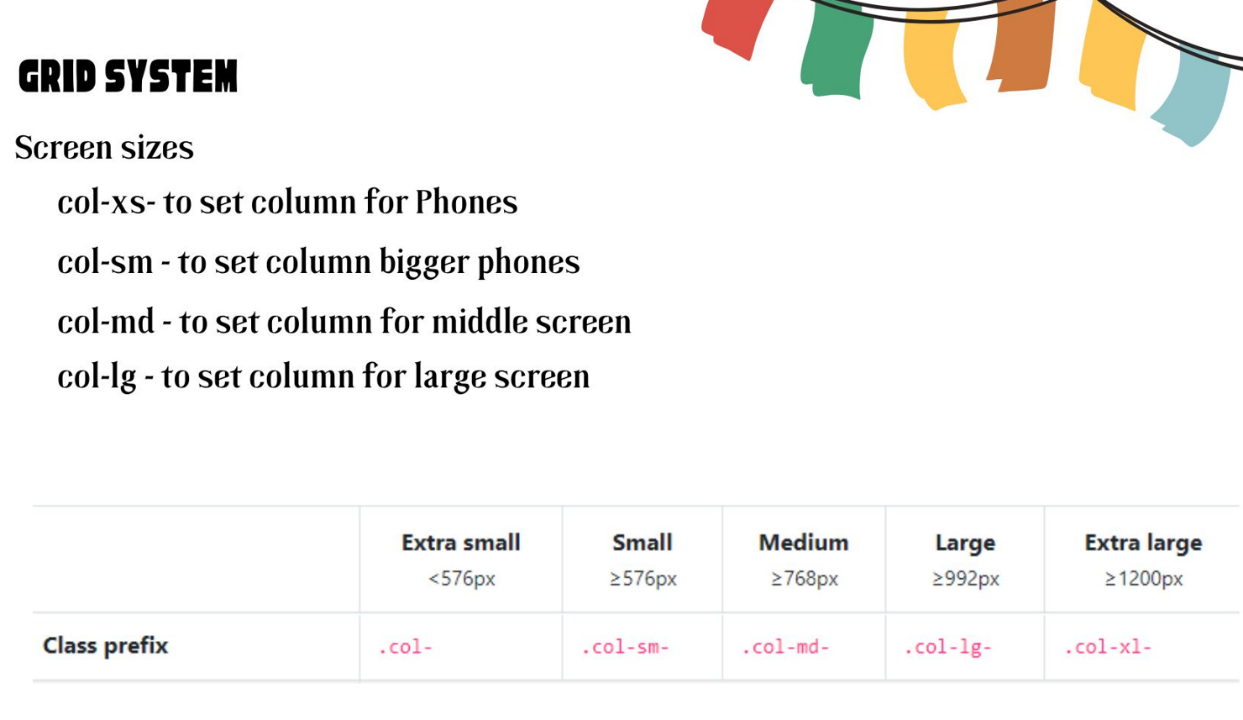
Row – Specify that a container in Bootstrap is a row and it should contain columns.

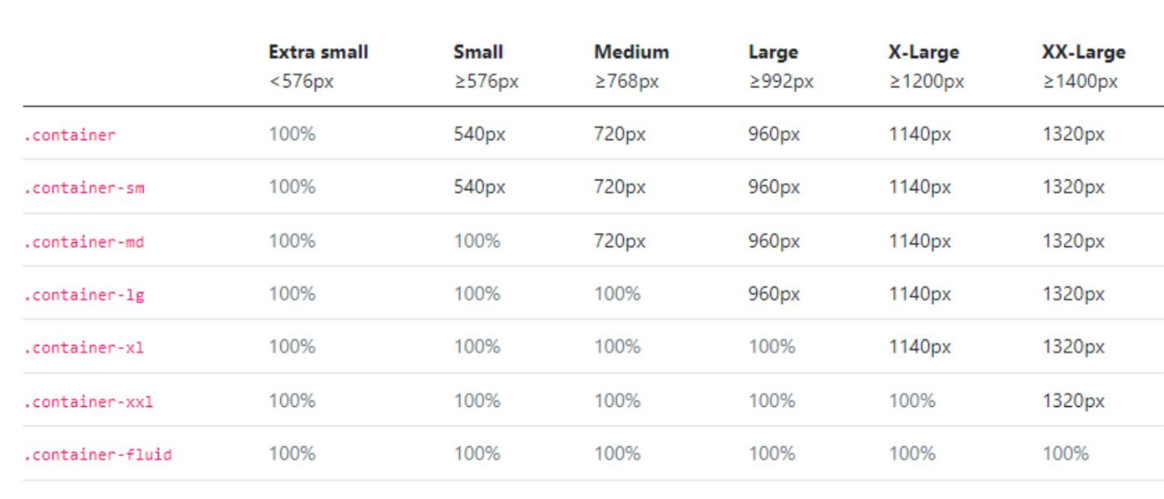
Col – To specify that the container is a column

**The Row width** is always a combination of 12 columns.



**Screen Sizes –** You could declare the column width based from the screen size





**Containers** – Use to make a maximum breakpoint for organizing content

**Lesson 3: Form Control**

**Form Control** – The design of Bootstrap for input elements

Select, checks and Radios, Range, Input Group, Floating labels, Color Picker

**Lesson 4: Utilities**

**Colors** – Boostrap makes it easier for us to apply colors

**The default colors –** Primary(blue), Danger(red), light(white), dark(black), warning(yellow-orange), info (blue-green) and success (Green)

**Backgrounds and Text** – You could easily apply the default colors to background and text using class bg for backgrounds and text to text

**Border** – Bootstrap has a default border with the class border

**Shadow** – Shadow is use to add aesthetic into an element, best used in white background elements.

**Display** – Bootsrap also offers a much easier approach in applying common CSS functionalities.

**Flex** – In addition to the grid system, Bootstrap also includes a powerful CSS flexbox utility that allows you to create flexible and responsive layouts. The flexbox utility classes in Bootstrap are used to control the alignment, positioning, and behavior of elements within a flex container. Here are some key concepts and classes related to flexbox in Bootstrap:

**Flex Container**

To create a flex container, you can use the **.d-flex** class. This class can be applied to any parent element to establish a flex context for its child elements.

By default, the flex container arranges its child elements in a row. To change the direction, you can use the**, .flex-row-reverse** or .**flex-column-reverse** classes.

**Flex Alignment**

Bootstrap offers alignment utility classes to control the alignment of flex items along the main axis and cross axis.

Classes like **.justify-content-**\* are used to align items along the main axis (horizontally for a row or vertically for a column).

<div class="d-flex justify-content-end">...</div>

<div class="d-flex justify-content-center">...</div>

<div class="d-flex justify-content-between">...</div>

<div class="d-flex justify-content-around">...</div>

<div class="d-flex justify-content-evenly">...</div>

Classes like **.align-items-\*** are used to align items along the cross axis.

<div class="d-flex align-items-end">...</div>

<div class="d-flex align-items-center">...</div>

<div class="d-flex align-items-baseline">...</div>

<div class="d-flex align-items-stretch">...</div>

Additionally, the **.align-content-\*** class can be used to align multiple lines of flex items when they wrap.

**align-content-end**: Aligns multiple rows of flex items at the end of the cross-axis.

**align-content-center**: Aligns multiple rows of flex items at the center of the cross-axis.

**align-content-between**: Aligns multiple rows of flex items with equal spacing between them.

**align-content-around**: Aligns multiple rows of flex items with equal spacing around them.

**align-content-stretch**: Stretches multiple rows of flex items to fill the cross-axis (default).

**Float** - Toggle floats on any element, across any breakpoint.

<div class="float-end">The EXAMPLE Content </div>

**Overflow** – Applies the overflow capability of Bootstrap

<div class="overflow-auto">...</div>

<div class="overflow-hidden">...</div>

**Position** – Quick approach for the css position property

<div class="position-static">...</div>

<div class="position-relative">...</div>

<div class="position-absolute">...</div>

<div class="position-fixed">...</div>

**Margin – Padding** - Use the margin and padding  to control how elements and components are spaced and sized. (1-5)

**Sizing** - Easily make an element as wide or as tall with our width and height utilities

<div class="w-25 p-3" style="background-color: #eee;">Width 25%</div>

<div class="w-50 p-3" style="background-color: #eee;">Width 50%</div>

<div class="w-75 p-3" style="background-color: #eee;">Width 75%</div>

<div class="w-100 p-3" style="background-color: #eee;">Width 100%</div>

<div class="w-auto p-3" style="background-color: #eee;">Width auto</div>

**Text alignment – wrapping** -  text alignment, wrapping, text break, text-transform and weight.

**Display property** – Configuration for display properties such as inline, block and hidden.

**Vertical align** - Easily change the vertical alignment of **inline, inline-block**, **inline-table**, and **table cell elements**.

<span class="align-baseline">baseline</span>

<span class="align-top">top</span>

<span class="align-middle">middle</span>

<span class="align-bottom">bottom</span>

<span class="align-text-top">text-top</span>

<span class="align-text-bottom">text-bottom</span>

**Lesson 5: Components**

**Buttons - Buttons, button group, Dropdown button, btng group vertical**

**Organizers – Accordion, Card, List Group**

**Navbar, Carousel, modal**

**Useful elements – badge, Progress, spinners, tool tip, Table**

**Bootstrap Icons**

**Activity: Make a portfolio Landing Page**