5 : Introduction to bootstrap

IT2406 - Web Application Development 1

Level I - Semester 2

Bootstrap Grid System

Introduction to Grids in Bootstrap

- Bootstrap includes a responsive, mobile first fluid grid system that appropriately scales up to 12 columns as the device or viewport size increases
- It includes predefined classes for easy layout options
- It includes powerful mixins for generating more semantic layouts
- Grid systems are used for creating page layouts through a series of rows and columns which hosts the content of a web page
- The Grid is made up of groupings of Rows & Columns inside one or more containers

How Bootstrap grid system works

- Rows must be placed within a ".container" class for proper alignment and padding
- Use rows to create horizontal groups of columns
- Content should be placed within columns
- Columns may be immediate children of rows
- Predefined grid classes like ".row" and ".col-xs-4" are available for quickly making grid layouts

How Bootstrap grid system works

- Columns create gutters (gaps between column content) via padding
- Padding is offset in rows for the first and last column via negative margin on “.rows”.
- Grid columns are created by specifying the number of twelve available columns
- For example:
  - Three equal columns would use three “.col-xs-4”
  - Two equal columns would use three “.col-xs-6”

Bootstrap Grids

- Bootstrap grid system allows up to 12 individual columns
- It allows to group the columns together to create wider columns too
- These columns will re-arrange automatically depending on the screen size

You can use any combinations to prepare the layout

```
<div class="col-md-12">Span 12 columns</div>
<div class="col-md-6"> 6 units</div><div class="col-md-6"> 6 units </div>
<div class="col-md-4"> 4 units </div><div class="col-md-8"> 8 units </div>
<div class="col-md-4"> 4 units </div><div class="col-md-4"> 4 units </div> <div class="col-md-4"> 4 units </div>
```

Grid Classes

- The Bootstrap grid system has four classes
  - xs - focuses on mobile phones
  - sm - focuses on tablets
  - md - focuses on desktops
  - lg - focuses on larger desktops
- These classes can be used to create more dynamic, flexible and complex layout structures.

Basic Structure of a Bootstrap Grid

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

Create a row ( )

Add the desired number of columns (tags with appropriate .col-*-*classes)

Note that numbers in .col-*-* should always add up to 12 for each row

GRID Options

	Extra small devices Phones (<768px)	**Small devices Tablets (≥768px)**	**Medium devices Desktops (≥992px)**	**Large devices Desktops (≥1200px)**
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints
Max container width	None (auto)	750px	970px	1170px
Class prefix	**.col-xs-**	**.col-sm-**	**.col-md-**	**.col-lg-**
# of columns	12	12	12	12
Max column width	Auto	60px	78px	95px
Gutter width	30px (15px on each side of a column)	30px (15px on each side of a column)	30px (15px on each side of a column)	30px (15px on each side of a column)
Nestable	Yes	Yes	Yes	Yes
Offsets	Yes	Yes	Yes	Yes
Column ordering	Yes	Yes	Yes	Yes

Media Queries

- Media queries are conditional CSS rules.
- It simply applies some CSS based on certain conditions
- When the specified conditions are met, the style is applied
- Media Queries in Bootstrap allows to move, show and hide content based on viewport size

Example media queries to create key break points

- Extra small devices (phones, less than 768px)
- No media query since this is the default in Bootstrap
- Small devices (tablets, 768px and up)

  @media (min-width: @screen-sm-min) { ... }
- Medium devices (desktops, 992px and up)

  @media (min-width: @screen-md-min) { ... }
- Large devices (large desktops, 1200px and up)

  @media (min-width: @screen-lg-min) { ... }

Basic example of single column

```
<div class="container">
    <div class="row">
        <div class="col">Single column grid</div>
    </div>
</div>
```

Basic example of two column structure

```
<div class="container">
    <div class="row">
        <div class="col">column one</div>
        <div class="col">column two</div>
    </div>
</div>
```

Basic example of three columns

```
<div class="container">
    <div class="row">
        <div class="col"> left column </div>
         <div class="col">center column</div>
        <div class="col">right column</div>
    </div>
</div>
```

The fundamental Rules of the Grid creation

Commonly three abstract level rules which are very useful

- Columns must be the immediate child of a Row
- Rows are only used to contain Columns, nothing else
- Rows should be placed inside a Container

IMPORTANT

The Rows & Columns always work together

.container class

- The Container can be used to hold any elements and content
- It is not only used for holding the Grid Rows & Columns
- It is very important to control width of the layout
- The Container is also used to evenly align the left and right edges of the layout within the browser's viewport
- The Container is used to counteract the negative margins of the Row

Viewport: *The visible area inside the browser window*

Types of containers in Bootstrap 4

- **Fixed-width** container to center your layout in the middle:

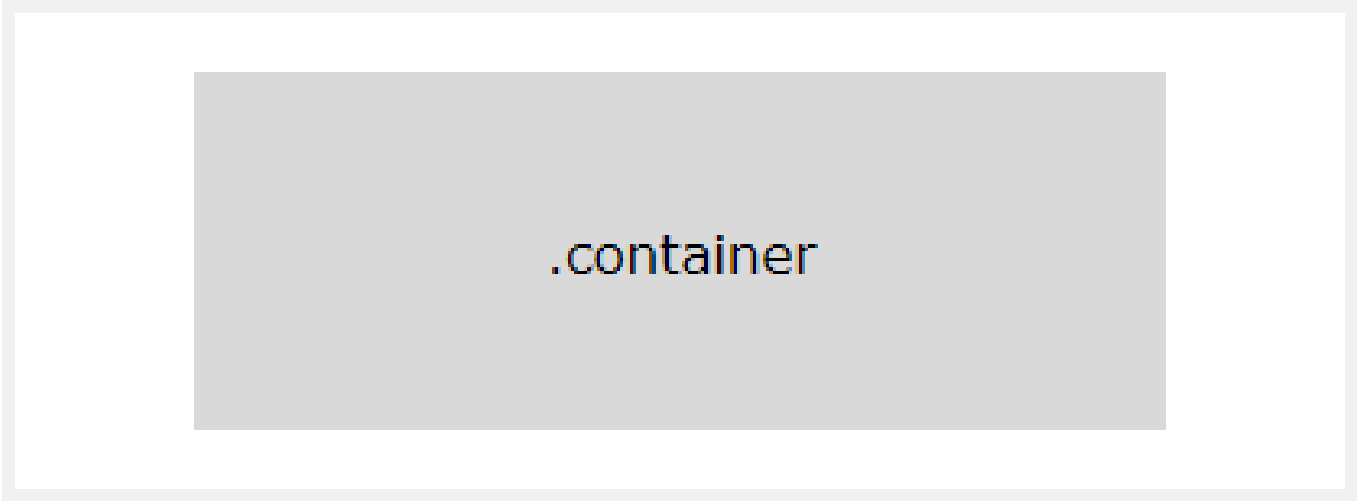
```
<div class="container"></div>
```

- **Full-width** container for a layout the spans the entire width:

```
<div class="container-fluid"></div>
```

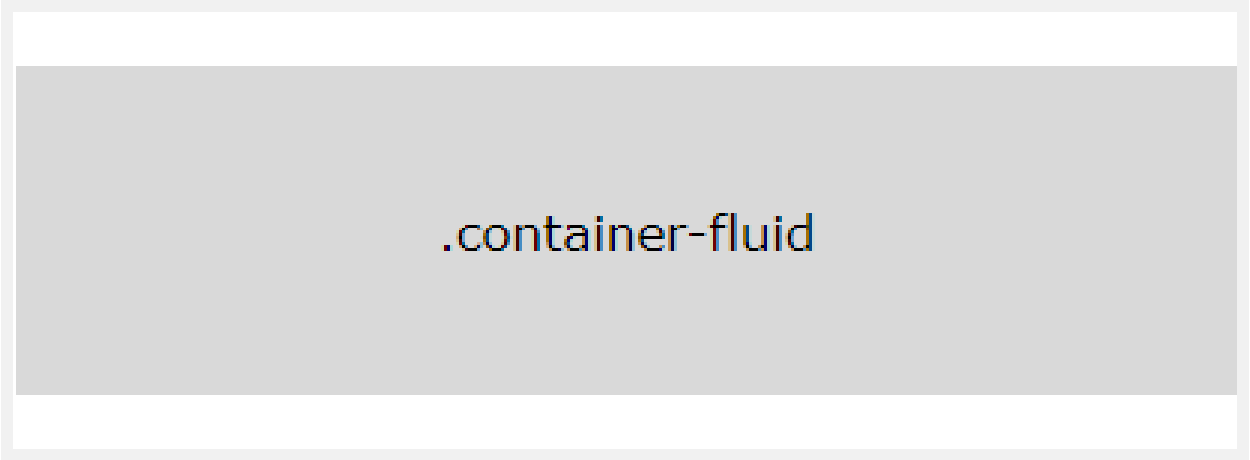
Container fixed

- The .container class provides a responsive fixed width container



Container fluid

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



Responsive containers

- .container-sm|md|lg|xl classes can be used to create responsive containers
- The max-width of the container will change on different screen sizes/viewports depending on the device

```
<div class="container-sm">.container-sm</div>
<div class="container-md">.container-md</div>
<div class="container-lg">.container-lg</div>
<div class="container-xl">.container-xl</div>
```

Maximum width change according to the viewport

Class	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
`.container-sm`	100%	540px	720px	960px	1140px
`.container-md`	100%	100%	720px	960px	1140px
`.container-lg`	100%	100%	100%	960px	1140px
`.container-xl`	100%	100%	100%	100%	1140px

Grouping columns to create wider columns