

## 9. Responsive website development: bootstrap

### 9.1 What is Bootstrap?

- **Definition**

- It is an HTML, CSS, and JS library developed by Twitter for developing responsive, mobile friendly or mobile first projects.
  - By library, we mean that Bootstrap is a collection of CSS and JavaScript files. When you connect these files into your HTML page, various Bootstrap classes and ready-made components become available for design layout.

- **Purpose**

- Mainly to simplify responsive web development
- Also, to provide some pre-defined styling for some HTML elements
- To speed up the creation of layouts by providing a great number of templates and ready-made solutions

- **Why use bootstrap?**

- Easy to be used by anyone with basic HTML and CSS knowledge
- To make use of its responsive features

- **Version**

- Bootstrap 5 is the newest version. The major changes in Bootstrap 5 are
  - removal of jQuery dependency for dynamic features and favoring vanilla JavaScript dependency. This is the main difference between Bootstrap 5 and all the other older versions
  - Inclusion of custom designed form controls that are built using semantic standards with no need to add extra HTML
  - Enhanced grid system by adding new classes (new xxl grid tier, new gutter classes and vertical spacing classes added)

## 9.2 How do we use Bootstrap?

- **You can use Bootstrap on your website in 2 ways**
  - 1. By downloading the required Bootstrap files (like the compiled and minified CSS and JS files) from <https://getbootstrap.com/> , saving them into your website folder and linking the files with your HTML
  - 2. By including Bootstrap by copying a CDN (Content Delivery Network) and adding the link into your HTML.
- **Steps to including Bootstrap in your website:**
  - **Step 1:** Create a project folder containing your HTML file
  - **Step 2:** Download Bootstrap's CSS and JS files from the Bootstrap website from <https://getbootstrap.com/> and save these files in your project.
  - **Step 3:** link both the CSS and JS files (you downloaded and saved in your project) to your HTML's <head> section. Similarly, if you plan to use the Bootstrap CDN, you copy and paste the CSS CDN links into your HTML's <head> before any of your custom CSS files.
    - Linking the Bootstrap .css file is just like linking your external CSS file
    - Make sure your Bootstrap CSS is linked before your custom CSS file.
    - There might come a case when you want to override the default Bootstrap CSS styling. In such cases, do not modify the bootstrap.css file. Just link your custom CSS file below the bootstrap.css file and apply the modification there
    - You can now start using the predefined Bootstrap CSS and JS properties!
      - Ex: Practice to link the Bootstrap files correctly

## 9.3 Bootstrap containers

- Containers are HTML elements, mostly divs, where we normally put every other HTML content of the site we build. Bootstrap containers wrap or pad contents of our website within them.
- **Bootstrap uses the 12-column grid system**
  - This means that there are only 12 columns available per each row. The full width of containers is divided into 12 equal columns. Remember, if you have more than 12 columns in a row, the columns will stack regardless of the screen size of your device.

- Bootstrap's grid system is responsive, and the columns will re-arrange depending on the screen size: On a big screen it might look better with the content organized in three columns, but on a small screen it would be better if the content items were stacked on top of each other.
- **In Bootstrap, there are 2 container classes available:**

#### **1. Fixed container (the .container class)**

- The **.container** class provides a fixed width container that is specifically defined to be a certain number for different screen sizes.
- To define a fixed container size on your HTML, you use the class `".container"`
- This class is defined starting from line 556 on bootstrap.css (Bootstrap 4) or line 592 of Bootstrap 5
- For example, look at line 556 of bootstrap.css (bootstrap 4) or 606 of Bootstrap 5: the width of a container that has a screen size above 576px is not more than 540px

#### **2. Fluid container (the .container-fluid class)**

- Fluid container means, the width of the container is defined to take 100% of the available screen size. There is no fixed amount specified for width, rather, the width of the container will always fill the entire width of the screen
- To define a flexible container size on your HTML, you use the class `".container-fluid"`
- `.container-fluid` is Defined on line 588 of bootstrap.css (Bootstrap 4) and line 593 of bootstrap.css (Bootstrap 5)

## 9.4 Bootstrap 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.
  - You can use media queries to conditionally apply styles your CSS based on breakpoints.
- Bootstrap is developed with the mobile first approach. Mobile first, responsive design is the goal. This means, every CSS that is defined for smaller screen size applies on bigger screen sizes unless it is overridden for larger devices. For example, the following styling of our container size applies to any screen size that is 576px and above

```
@media (min-width: 576px) {  
  
    .container {  
  
        max-width: 540px;  
  
    }  
  
}
```

- **Bootstrap 5 uses 6 break points to handle responsiveness:** please note that each breakpoint (also known as grid tiers) was chosen to comfortably hold containers whose widths are multiple of 12.
  - **Extra-small** < 576px (Mini phones)
  - **Small (sm)** ≥ 576px (Regular phones)
  - **Medium (md)** ≥ 768px (Tablets)
  - **Large (lg)** ≥ 992px (Laptops)
  - **Extra-large (xl)** 1200px (Monitors)
  - **Extra-extra-large (xxl)** 1400px (Any other bigger screen device)
  - Look at this link to understand the breakpoints for Bootstrap 4 and 5:
    - <https://getbootstrap.com/docs/4.4/layout/grid/#grid-options> (Bootstrap 4)
    - <https://getbootstrap.com/docs/5.0/layout/grid/#grid-options> (Bootstrap 5)

## 9.5 Bootstrap grid system (rows)

- The Bootstrap grid system is built using two main components: rows and columns. It's built with flexbox and is fully responsive.
  - Notes: - Every column is defined within a row and row is divided into 12 equal columns - We use the class ".row" to define a row. There are multiple classes we use to define columns. We will see them later.
  - Ex: How to define a row    `<div class="row"></div>`

## 9.6 Bootstrap grid system (columns)

- **Columns**
  - **Class prefix:** we use Bootstrap class prefixes to define the width of specific columns on different screen sizes. There are multiple class prefixes used to define the width of a column. Note: As Bootstrap uses the mobile first approach, anything defined on smaller size screen also applies for bigger screen sizes if it is not overridden. Every column class is defined to contain 100% of the container when it is outside of its defined screen size. We will see this in example later.
  - Below, you will find the 5 Bootstrap grid classes (column class prefix) to define width of a column for different devices. What follows the prefixes is normally a number from 1 - 12 determining how many columns an element should contain. The width that corresponds to the specific class is predefined in the CSS file.
    - **.col-** : this prefix is used to define the width of a column on any screen size. Example: `<div class="col-4"> Four columns </div>`
    - **.col-sm-** : this prefix is used to define the width of a column on mobile devices. Example: `<div class="col-sm-4"> Four columns </div>`
    - **.col-md-** : this prefix is used to define the width of a column on Tablets. Example: `<div class="col-md-4"> Four columns </div>`
    - **.col-lg-** : this prefix is used to define the width of a column on Laptops. Example: `<div class="col-lg-4"> Four columns </div>`
    - **.col-xl-** : This prefix is used to define the width of a column on larger screens. Example: `<div class="col-xl-4"> Four columns </div>`
    - **.col-xxl-** : This prefix is used to define the width of a column on extra-large screens. Example: `<div class="col-xxl-4"> Four columns </div>`

- **Refer these links to see how Bootstrap breakpoints and container width work across multiple devices:**
  - <https://getbootstrap.com/docs/4.4/layout/grid/#grid-options> (Bootstrap 4)
  - <https://getbootstrap.com/docs/5.0/layout/grid/#grid-options> (Bootstrap 5)
- **Responsive columns:** responsiveness of Bootstrap is controlled by the combination of container width and column width
  - **Container width:** for fluid containers, the width is always at 100%. For fixed containers, the corresponding width of the containers is defined in the Bootstrap CSS. Example: Look at how the width of the containers respond when resizing the width of the browser
  - **Column width:** since column width is defined in percentage, it automatically adjusts when the container width changes. We can also use different column width for multiple screen sizes. The way to achieve this is to include multiple column classes that are going to redefine the width of the container based on the device size. Example: `<div class="col-sm-12 col-md-6 col-lg-3"> Responsive columns </div>`
- **Auto-layout columns**
  - **Equal width:** here, you can use the “.col” class without explicit number/unit to make the size of each column to be of the equal width. When you use just the “.col” class, bootstrap will automatically try to use equal space for each column that apply to every device width. In the example below, you can see that the 2 divs inside the row have equal width for all screen sizes.

```
<div class="row" >
```

```
    <div class="col"> Equal column size </div>
```

```
    <div class="col"> Equal column size </div>
```

```
</div>
```

- **Column breaks:** Breaking columns to a new line in flexbox requires a trick: adding an element with width: 100% wherever you want to wrap your columns to a new line. Please note that this can also be accomplished by using multiple row classes.

```
<div class="row">

    <div class="col-6 col-sm-3">.col-6 .col-sm-3</div>

    <div class="col-6 col-sm-3">.col-6 .col-sm-3</div>

    <!-- Force next columns to break to new line by adding extra div with
    width of 100% -->

    <div class="w-100"></div>

    <div class="col-6 col-sm-3">.col-6 .col-sm-3</div>

    <div class="col-6 col-sm-3">.col-6 .col-sm-3</div>

</div>
```

- **Variable width:** you can also set the column width/value (for any breakpoint size) to "auto" to size columns based on the natural width of their content.

```
<div class="row ">

    <div class="col "> Column 1 </div>

    <div class="col-md-auto"> Variable width content</div>

    <div class="col "> Column 3</div>

</div>
```

- **Refer these links for detailed explanation on additional Bootstrap classes:**
  - **Gutters:** <https://getbootstrap.com/docs/4.4/layout/grid/#gutters>
  - **Alignment:** <https://getbootstrap.com/docs/4.4/layout/grid/#alignment>
  - **Order:** <https://getbootstrap.com/docs/4.4/layout/grid/#reordering>
  - **Jumbotron:** <https://getbootstrap.com/docs/4.0/components/jumbotron/>
  - **Nav:** <https://getbootstrap.com/docs/4.4/components/navs/>
  - **Btn:** <https://getbootstrap.com/docs/4.4/components/buttons/#button-tags>

- **Form:** <https://getbootstrap.com/docs/4.4/components/forms/>