Fundamentals of Web Development

Module 5 - Responsive Web Design

Learning Objectives

- ➔ Introduction to Responsive Web Design (RWD)
- ➔ Media queries and breakpoints
- ➔ Introduction to Bootstrap
- ➔ Containers
- ➔ Grid
- ➔ Typography
- ➔ Spacing
- ➔ Navbar
- ➔ Combinators
- ➔ Specificity

Introduction to RWD

- ➔ Responsive web design (RWD) is a web design approach to make web pages render well on all screen sizes and resolutions while ensuring good usability.
- ➔ RWD must be thought and implemented from the very start of designing the web page.
- ➔ HTML is fundamentally responsive, or fluid (i.e.)if a web page containing only HTML, with no CSS, and if resized the window, the browser will automatically reflow the text to fit the viewport.
- ➔ The viewable area of a browser is known technically as the viewport. The viewport is seldom equivalent to the screen size of a device, especially in instances where a user can resize a browser window.
- ➔ This viewport meta tag tells mobile browsers that they should set the width of the viewport to the device width, and scale the document to 100% of its intended size.

<meta name="viewport" content="width=device-width,initial-scale=1" />

Media Queries

➔ Uses the @media rule to include a block of CSS properties only if a certain condition is true.

```
@media only screen and (max-width: 600px) {
  body {
      background-color: lightblue;
  }
}
```

Click Here

Introduction to Bootstrap

- ➔ Bootstrap is a free, open source and is the most popular HTML,CSS and JavaScript framework developed by twitter for creating responsive web application.
- ➔ Bootstrap is the most popular framework for quickly styling your website.
  - ◆ **Grid**
  - ◆ **Typography**
  - ◆ **Accordions**
  - ◆ **Alerts**
  - ◆ **Badge**
  - ◆ **Button**
  - ◆ **Breadcrumb etc.**

Containers

- Containers are the most basic layout element in Bootstrap and are required when using the grid system.
- Containers are basically used to wrap content with some padding.
- **Three main types:**
  - .container → Fixed-width (changes with breakpoints)
  - .container-fluid → 100% width at all breakpoints
  - .container-{breakpoint} → Fixed width until that breakpoint (.container-sm, .container-md, etc.)

Containers

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	X-Large ≥1200px	XX-Large ≥1400px
.container	100%	540px	720px	960px	1140px	1320px
.container-sm	100%	540px	720px	960px	1140px	1320px
.container-md	100%	100%	720px	960px	1140px	1320px
.container-lg	100%	100%	100%	960px	1140px	1320px
.container-xl	100%	100%	100%	100%	1140px	1320px
.container-xxl	100%	100%	100%	100%	100%	1320px
.container-fluid	100%	100%	100%	100%	100%	100%

Click Here

Grid

➔ Used to create page layouts through a series of rows and columns. The Grid system consists of 12 columns.

Click Here

	xs <576px	sm ≥576px	md ≥768px	lg ≥992px	xl ≥1200px	xxl ≥1400px
Container `max-width`	None (auto)	540px	720px	960px	1140px	1320px
Class prefix	`.col-`	`.col-sm-`	`.col-md-`	`.col-lg-`	`.col-xl-`	`.col-xxl-`
# of columns	12					

Click Here

Typography

➔ Typography makes it easy to create headings, paragraphs, lists etc. in a way that would be appealing to the users.

Click Here

Spacing

- ➔ Bootstrap Spacing Utilities — one of the most used and easiest features to make your layout neat and balanced.
- ➔ Bootstrap provides **margin (m)** and **padding (p)** utility classes that you can apply directly to any element

Click Here

Navbar

➔ The Bootstrap Navbar helps you create responsive menus that automatically adapt to different screen sizes

Click Here

Combinators

- ➔ In CSS, combinators define the relationship between selectors and how elements in a document are selected based on their relationships with other elements.
- ➔ Combinators allow you to target elements that are related to other elements in various ways, whether it's by proximity, hierarchy, or siblings.

Combinators

- **Descendant Combinator**
  - The descendant combinator selects all elements that are descendants of a specific element. A descendant can be a child, grandchild, or any deeper level of nested element.

    Syntax: ancestor descendant

- **Child Combinator (>)**
  - The child combinator selects elements that are direct children of a specified element, meaning there are no intermediate elements between the parent and the child.

    Syntax: parent > child

Click Here

Combinators

- **Sibling Combinator(~)**
  - Selects all elements B that are siblings of A and come after it, not just the immediate one.

    Syntax: A ~ B

- **Adjacent Sibling Combinator (+)**
  - Selects the element B that is immediately next to element A (they share the same parent).

    Syntax: A + B

Click Here

Specificity

- Specificity determines which CSS rule is applied when multiple rules target the same HTML element.

Priority	Selector type
1	Inline Style
2	ID Selector or pseudo class selector
3	Class Selector
4	Element selector

Click Here

End of Module 5

Queries?