Lesson:

Introduction to Responsive Web Design

PW SKILLS

Topics Covered

- What is Responsive web design(RWD)?
- Advantages of RWD
- Ways to implement Responsive web design

What is Responsive Web Design(RWD)?

Responsive Web Design (RWD) is an approach to web design and development that aims to create websites that provide an optimal viewing and user experience across a wide range of devices and screen sizes. For smartphones, tablets, and various other devices, it is important for websites to be adaptable and accessible on different platforms.

The primary goal of responsive web design is to ensure that a website's layout and content can automatically adjust and adapt to fit the screen size and orientation of the device being used. This means that regardless of whether someone is viewing a website on a desktop computer, a smartphone, or a tablet, the website should be visually appealing, easy to navigate, and provide a seamless user experience.

Advantages of RWD

1. **Improved User Experience:** RWD ensures that your website looks and functions well across different devices, providing a seamless and user-friendly experience for visitors.

2. **Increased Mobile Traffic:** With the rising use of mobile devices, RWD allows your website to cater to a larger audience, as it adapts to various screen sizes and resolutions.

3. **Cost and Time Efficiency:** Instead of creating separate websites or apps for different devices, RWD allows for a single codebase, reducing development time and costs associated with maintaining multiple versions.

4. **Search Engine Optimization (SEO) Benefits:** Having a responsive website is favoured by search engines like Google, as it avoids duplicate content and provides a consistent URL structure. This can contribute to better search rankings and increased visibility.

Ways to implement Responsive Web Design

We have several ways to implement RWD, we will discuss a few common techniques to make your website Responsive.

1. **Responsive Layouts-** By default, several layout methods such as Multiple-column layout, Flexbox, and Grid possess inherent responsiveness.

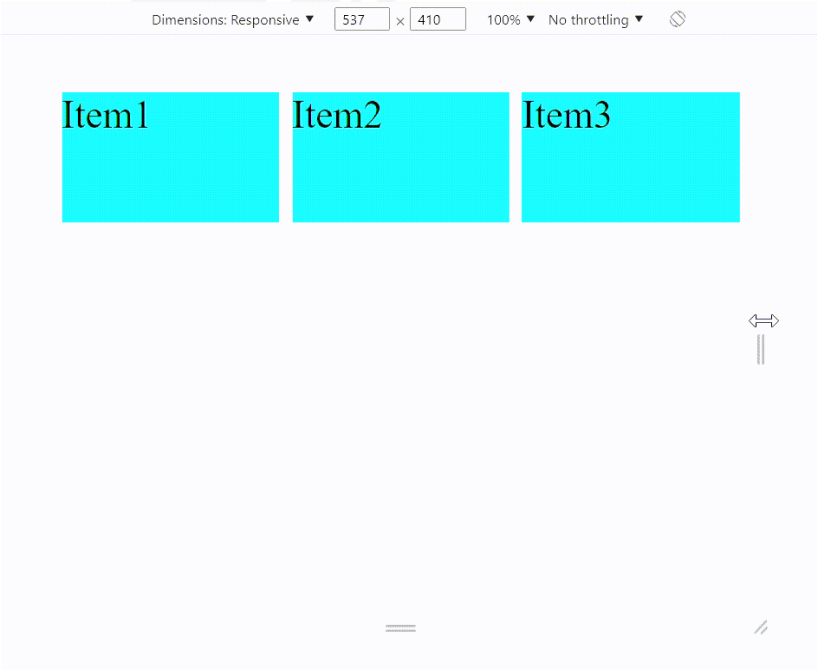
Although we have already studied flex and grid, let's take an example to recall it with respect to responsive design.

Example:- Check below responsive design using flex layout.

```
<style>
   .container {
     display: flex;
     justify-content: space-around;
     gap: 10px;
   }

   .container div {
     width: 100%;
     height: 100px;
     background-color: aqua;
     font-size: 30px;
   }
 </style>
 <body>
   <div class="container">
     <div>Item1</div>
     <div>Item2</div>
     <div>Item3</div>
   </div>
 </body>
```

BROWSER OUTPUT-



In the above example, we have 3 three flex items, and if you observe they are by default responsive because of **display:flex**

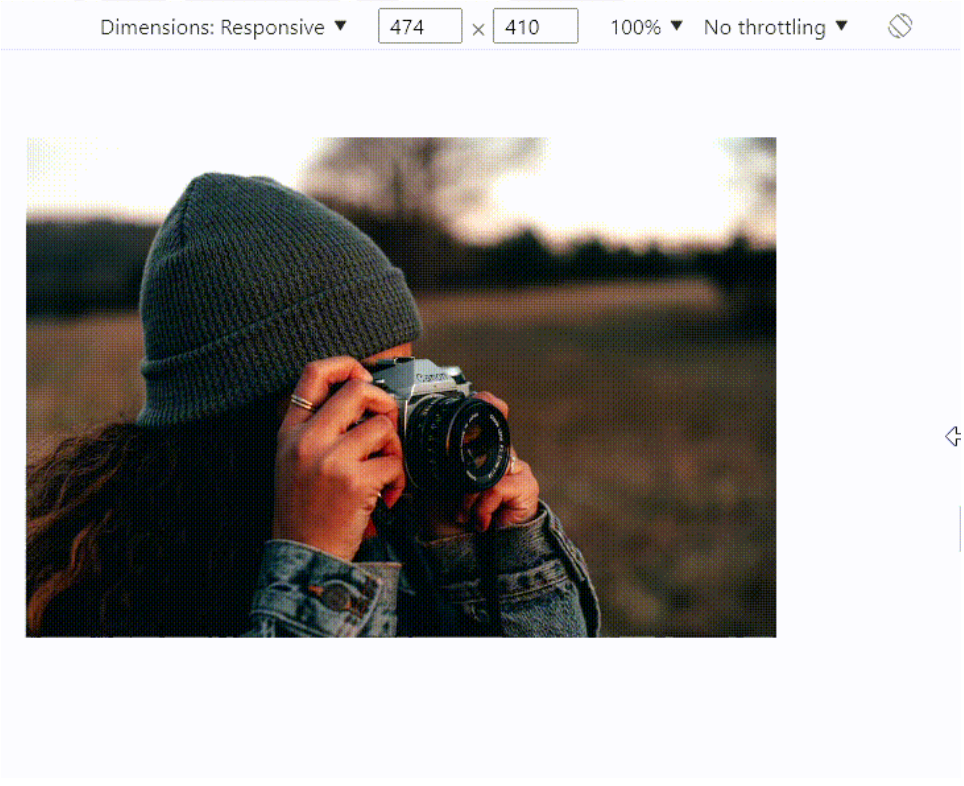
2. **Responsive images-** To make the image size responsive, we can use **max-width:100%**, so that image never overflows its container and adjusts itself as the container size increases or decreases.

Example:

```
<style>
  .container{
    width: 80vw;
  }

  img {
    max-width: 100%;
  }
</style>

<body>
   <div class="container">
      <img src="./assets/photo.jpg" alt="responsive image"></div>
   </div>
</body>
```



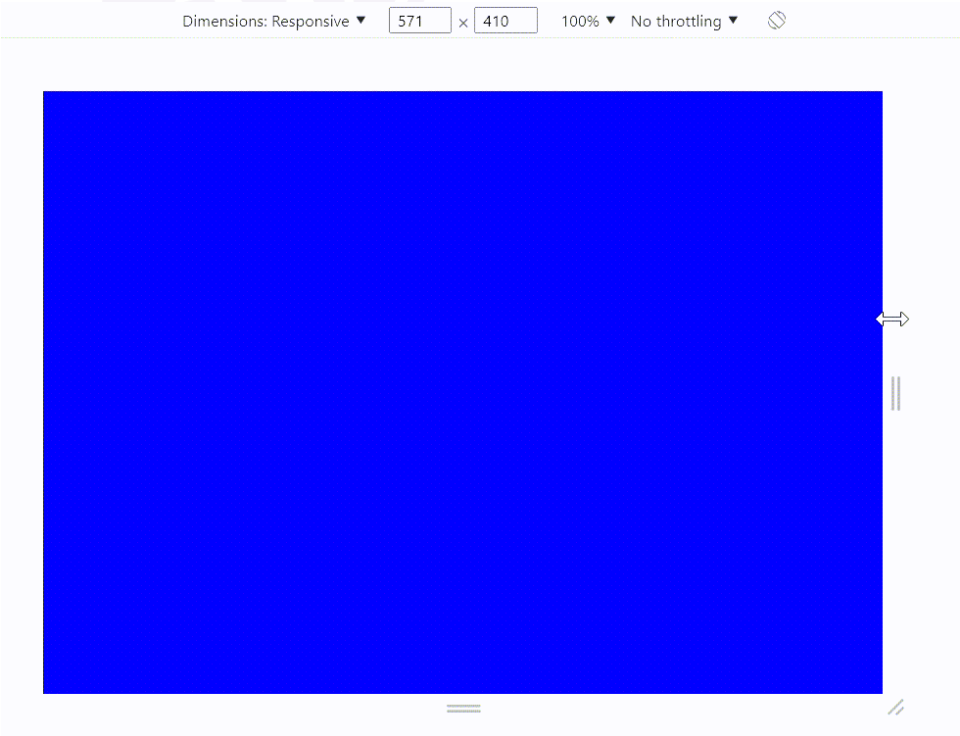
To analyse the output please open inspect mode**(CTRL + SHIFT + I).**

3. **Media Query-** Media query is the most prominent method for creating mobile responsive web designs. Media queries enable us to perform a set of evaluations, such as determining if the user's screen exceeds a specific width, and selectively applying CSS to appropriately style the webpage based on the user's requirements.

Example

```
<style>
  body {
     background-color: aqua;
  }

/* Apply below CSS when screen width reaches ⩾ 600px */
  @media (max-width: 600px) {
    body {
      background-color: blue;
    }
  }
</style>
<body>
    <div class="container"></div>
</body>
```



In the above example, if you observe background colour changes, when it reaches screen width more than or equal to **600px**. To analyse the output please open inspect mode**(CTRL + SHIFT + I)**. We will study more above media queries in upcoming lessons.

4. **Responsive Typography-** Responsive typography refers to the technique of adjusting font sizes based on either media queries or viewport units, depending on the available screen space.

Example

```
<style>
   div {
     font-size: 10vw;
  }
</style>
<body>
    <div>Responsive Text</div>
</body>
```

Here, we have set the font size of text as **10vw** (10% of viewport width). To analyse the output please open inspect mode**(CTRL + SHIFT + I)** and try changing viewport width and observe text size.

5. **The Viewport meta tag-** In HTML source code of a responsive webpage, it is common to observe the presence of the following <meta> tag within the <head> section of the document.

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

Let's break down the attributes and their meanings:

- **name="viewport":** This attribute specifies that the <meta> tag is defining the viewport settings for the web page.
- **content="width=device-width, initial-scale=1":** This attribute defines the content, or the actual instructions, for the viewport settings. It consists of two parts:
- **width=device-width:** This instructs the browser to set the width of the viewport to the width of the device's screen. It ensures that the webpage's layout adapts to the screen size of the device being used, allowing content to fit properly.
- **initial-scale=1:** This sets the initial zoom level of the webpage to 100%. It ensures that the webpage is displayed at a 1:1 scale, without any initial zooming or scaling applied.