HTML Responsive Web Design

Responsive web design means creating HTML webpages that work well on different devices and browsers. The pages automatically adjust to fit various screen sizes and viewports.

What is a Responsive Web Design?

Responsive web design is an approach that allows the creation of the webpages compatible with different screen sizes, resolutions, and devices, such as desktops, laptops, tablets, and smartphones. It helps in improving the user experience, accessibility, and performance of a website.

When a responsive website, such as **TutorialsPoint.com**, is accessed on any device of different dimensions, the content is displayed optimally without any overflow or overlap with other elements. The content remains consistent across devices, but its arrangement adapts to ensure readability.

Best Practices for Making Webpages Responsive

There are several ways by which we can make our webpage responsive. A general practice is to make use of inbuilt CSS or bootstrap frameworks that provide pre-designed components and grid systems. Here are some general steps to ensure the responsiveness of a webpage.

- **Use Responsive Grid Layouts**
  Always design your layout with a flexible grid structure, so when the size of the screen increases, the grid will extend accordingly.
- **Use Responsive Images and Media**
  Ensure that the images on the webpage are properly scaled within their containers. You can use CSS properties like **"max-width: 100%;"** and **"height: auto."**.
- **Use Media Queries**
  You can use CSS media queries to apply different styles for different screen sizes. This allows you to adjust the layout, font sizes, and other design elements based on the device's width.
- **Use a Meta Tag with Viewport**
  The viewport meta tag inside the HTML <head> tag ensures proper scaling and rendering on mobile devices.
- **Use Relative Units**
  Use relative units like em, rem, or percentages for font sizes, padding, and margins to ensure that text scales appropriately.

Using the Viewport for Responsive Design

The viewport represents the visible area of a user's device. The content that is outside of the viewport can be seen if scrolled. It helps web pages' render well on different devices by controlling the width and scale of the page.

To control the viewport, add the following `<meta>` tag in the **`<head>`** section −

```
<meta name = "viewport" content = "width=device-width, initial-scale = 1.0">
```

The above tag tells the browser to match the page width to the screen's width and set the initial zoom level when the page is first loaded by the user.

We can adjust viewport settings for width, **`initial-scale`**, **`maximum-scale`**, **`minimum-scale`**, and **`user-scalable`**. These adjustments can make our website more accessible and user-friendly.

Attributes of the Meta Viewport Tag for Responsiveness

The following attributes of the **`<meta>`** viewport tag are used for responsiveness −

S.No.	Attribute & Description
1	**`width`** It controls the width of the virtual viewport.
2	**`height`** It controls the height of the virtual viewport.
3	**`initial-scale`** It specifies the initial zoom level of the viewport when a web page is first loaded.
4	**`minimum-scale`** It specifies the minimum zoom level to which the user can zoom the page.
5	**`maximum-scale`** It defines the maximum zoom level to which the user can zoom the page.
6	**`user-scalable`** It specifies whether the user can zoom in or out.
7	**`interactive-widget`** It defines how the interactive UI widgets affect the viewport.

Example

The following example illustrates how to use the **`<meta>`** viewport tag to make a web page responsive:

</> Open Compiler

```
<html>
<head>
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <style>
      .container {
         background-color: #f1f1f1;
         display: flex;
         flex-direction: row;
      }
      .col {
         width: 47%;
         margin: auto 1%;
         background-color: #4CAF50;
         color: white;
         text-align: center;
         line-height: 100px;
         font-size: 10px;
      }
   </style>
</head>
<body>
   <h2>Setting up dimensions in percentage for the HTML element </h2>
   <div class="container">
      <div class="col"> Column 1 </div>
      <div class="col"> Column 2 </div>
   </div>
</body>
</html>
```

On executing the above HTML code, a two-column responsive layout will be displayed.

Creating Responsive Text

In **HTML**, to make **responsive text** that adjusts its font size automatically based on the viewport, we need to use the `font-size` property of CSS along with the **"vw"** length unit. The **vw** is an abbreviation that stands for **view width** or **viewport width**, which is a relative length unit of CSS.

The relative length units are always calculated with respect to another element's size. Note that **1vw** is equal to 1% of the width of the viewport.

Example

In this example, we are using the "vw" length unit to make the text responsive:

</> Open Compiler

```
<html>
<head>
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
   <h1 style="font-size:6vw;">Example of Responsive Typography</h1>
   <h2 style="font-size:5vw;">Responsive text by Using the vw length unit.</h2>
   <p style="font-size:3vw;"> The text will adapt the font size according to the
device's width. </p>
</body>
</html>
```

The above HTML code will generate a responsive Text.

Creating Responsive Images

In **HTML**, we can create images that are responsive, meaning they can adjust their size to fit the viewport. To do so, we can either set the image's **width** property to 100% or the **max-width** property to 100%. The **width** property can scale the image larger than its original size, on the other hand, the max-width property ensures the image does not scale beyond its original size.

Example

The following example shows how to create responsive images. For the first image, we are using the **width** property, and for the second one max-width property:

</> Open Compiler

```
<html>
<head>
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
   <h2>Setting the width property to 100% </h2>
   <img src="https://www.tutorialspoint.com/images/logo.png" alt="logo"
style="width:100%; ">
   <h2> Creating the responsive image by setting the max-width property to 100% </h2>
   <img src="https://www.tutorialspoint.com/images/logo.png" alt="logo" style="max-
width:100%; height:auto; ">
</body>
</html>
```

On running the above code, we will get a responsive image.

Responsive Design Using Media Queries

The CSS **media query** serves as a filter that enables us to style the web page selectively for different devices. A single web page can have multiple media queries, each for a specific screen size. To define these screen sizes, we use the media query breakpoints and style the HTML elements accordingly. Here are the common breakpoints –

```
Mobile: 360 x 640 px
Tablet: 768 x 1024 px
Laptop: 1366 x 768 px
HDdesktop: 1920 x 1080 px
```

Example

The following HTML code demonstrates the use of media queries in designing a responsive layout.

</> Open Compiler

```
<html>
<head>
   <style>
      .main {
         width: 50%;
         height: 50vh;
         display: flex;
         align-items: center;
         text-align: center;
         margin: 10px auto;
         flex-direction: column;
      }
      img {
         width: 100%;
         height: 100%;
      }
      .description {
         width: 100%;
         height: 100%;
         font-size: 30px;
         color: red;
         margin-top: 20px;
      }
      /* using media query */
      @media screen and (max-width: 600px) {
         .main {
            width: 100%;
            height: 100vh;
            margin: 5px auto;
         }

         .description {
            font-size: 20px;
            color: blue;
            margin-top: 10px;
         }
      }
   </style>
</head>
<body>
   <div class="main">
      <img src="https://www.tutorialspoint.com/images/logo.png" alt="logo" width="100"
height="200">
      <div class="description"> The above is a logo of Tutorilaspoint. The logo is
responsive, and it will be displayed in the centre of the screen. </div>
   </div>
</body>
</html>
```

In addition to the above-mentioned techniques, other methods such as **Flexbox** and **CSS Grid** can also be used to design responsive web pages. We have explored these techniques in detail in the previous chapters.