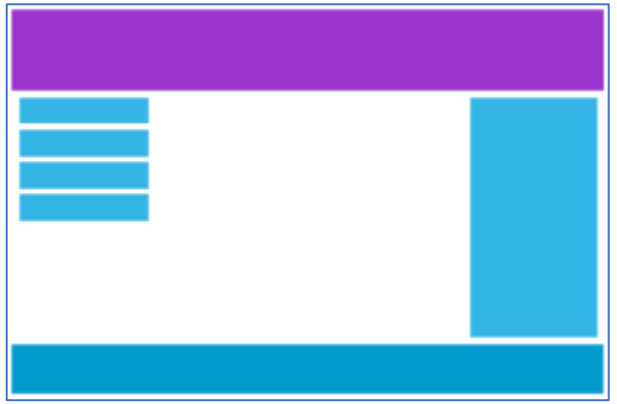
Responsive Web Design(RWD)

- **Responsive Web Design(RWD)** makes your web page **look good on all devices.**
- Designing For **The Best Experience For All Users**

- Responsive web design is not a program or a JavaScript.

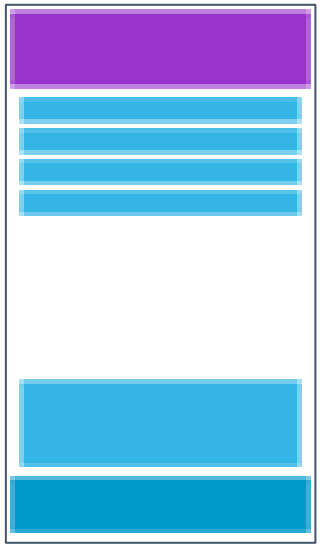
- Web pages can be viewed using many different devices: **desktops, tablets, and phones**. Your web page should look good, and be easy to use, regardless of the device.

- Web pages **should not leave out information to fit smaller devices**, but rather adapt its content to fit any device



Desktop

Tablet



Phone

Responsive Web Design - Media Queries

- **Media query** is a CSS3 technique uses the **@media rule** to include a **block of CSS properties only if a certain condition is true.**
- Ex. Given below **sets the background color of the body to light blue when the screen width is 600 pixels or less**

Media Queries – changind BG COLOR

```
<html> <head>
<style>
body { /* for desktop screen */
  background-color: lightgreen;
}
@media only screen and (max-width: 600px) {
  body {
      /* for tablet screen *?
    background-color: lightblue;
  }
}
</style>
</head>
```

```
@media only screen and (max-width: 300px)
{
  body {
      /* for mobile screen */
    background-color: lightblue;
  }
```

```
<body>
<p>Resize the browser window. When the
width of this document is 600 pixels or
less, the background-color is
"lightblue", otherwise it is
"lightgreen".</p>
</body>
</html>
```

Resize the browser window. When the width of this document is 600 pixels or less, the background-color is "lightblue", otherwise it is "lightgreen".

Media Queries - Variable Font Size.

- We can also use media queries **to change the font size of an element** on different screen sizes

```
<html>
<head>
<style>
div{
  background-color: lightgrey;
  padding: 20px;
}
@media screen and (min-width: 600px) {
  div{
    font-size: 80px;
  }
}
@media screen and (max-width: 600px) {
  div{
    font-size: 25px;
  }
}
</style>
```

```
</head>
<body>
<div>Example DIV.</div>
</body>
</html>
```

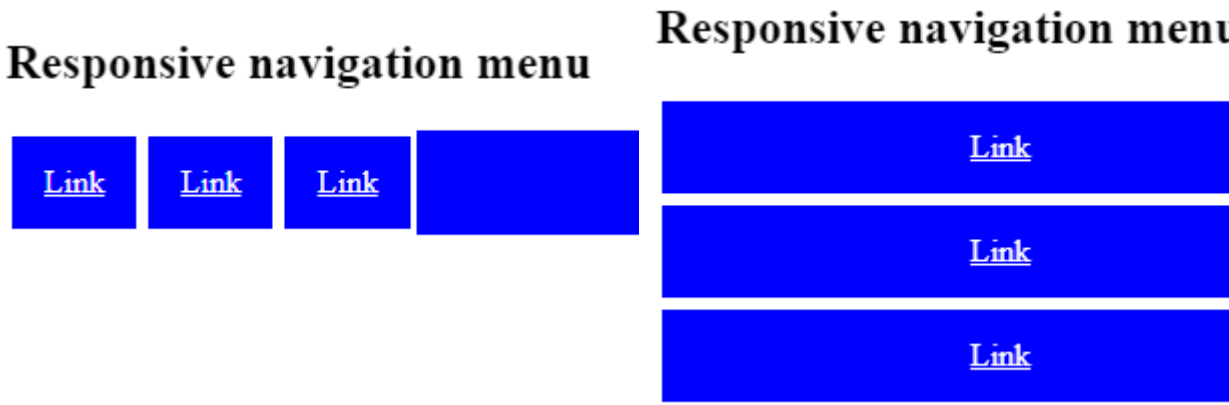
Example DIV.

Media Queries For Menus

- Media queries to create a responsive **navigation menu**, that varies in design on **different screen sizes.**

```
<html>
<head>
<style>
/* Style the top navigation bar */
.topnav {
  overflow: hidden;
  background-color:blue;
}
/* Style the topnav links */
.topnav a {
  float: left;   display: block;
  color: white;   text-align: center;
  padding: 14px 16px;   border-style: solid;
}
/* Change color on hover */
.topnav a:hover {
  background-color: #ddd;
  color: black;
}
```

```
/* On screens that are 600px wide or less,
make the menu links stack on top of each other
instead of next to each other */
@media screen and (max-width: 600px) {
  .topnav a {
    float: none; width: 100%;
  } }
</style> </head> <body>
<h2>Responsive navigation menu</h2>
<div class="topnav">
  <a href="#">Link</a> <a href="#">Link</a>
  <a href="#">Link</a> </div> </body>
</html>
```



Responsive Web Design - Images

Using The **width Property**

- If the **width property** is **set to a percentage** and the **height property** is set to "**auto**", the **image will be responsive and scale up and down**

Resize the browser window to see how the image will scale.

```
<html>
<head>
<style>
img{
  width: 100%;
  height: auto;
}
</style>
</head>
<body>

<img src="flower1.jpg" width="460"
height="345">
<p>Resize the browser window to see how the
image will scale.</p>

</body>
</html>
```

Responsive Web Design - Images

max-width Property

- If the **max-width property is set to 100%,** the image will scale down if it has to, **but never scale up to be larger than its original size**:

Resize the browser window to see how the image will scale when the width is less than 460px.

```
<html>
<head>
<style>
img {
  max-width: 100%;
  height: auto;
}
</style>
</head>
<body>

<img src="flower1.jpg" width="460"
height="345">
<p>Resize the browser window to see how the
image will scale when the width is less than
460px.</p>

</body>
</html>
```

Responsive Web Design - Images

Different Images for Different Devices

- **media queries used** to display **different images on different devices.**

- Here is one large image and one smaller image that will be displayed on different devices:

Resize the browser width and the background image will change at 400px.

```
<html>
<head>
<style>
/* For width smaller than 400px: */
 body{
  background-repeat: no-repeat;
  background-image: url('img_smallflower.jpg');
 }
/* For width 400px and larger: */
@media only screen and (min-width: 400px) {
  body {
     background-image: url("flower1.jpg");
  }
}
</style>
</head>
<body>
<p style="margin-top:360px;">Resize the browser width and the background image will change at 400px.</p>
</body></html>
```

How to Set the width-range for a Media Query

The method we just discussed of creating media queries by applying just one **width** property solves just one problem

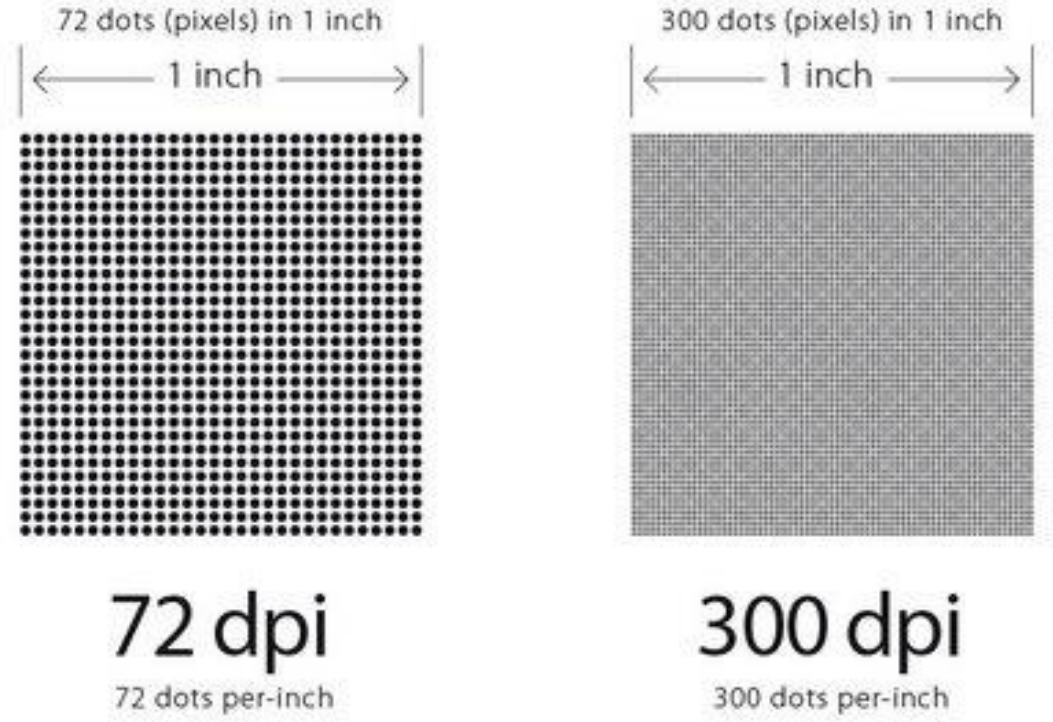
Resize the browser width and the background image will change at 400px.

```
<html>
  <head>
    <title>Example Media Query</title>
    <style>
      /* styles for all screens */
      .image {
        width: 100%;
        height: auto;
      }
/* styles for screens with a maximum width of 768px */
      @media screen and (max-width: 768px) {
        .image {
          max-width: 100%;
        }
      }
    </style>
  </head>
  <body>
    <img src="flower1.jpg" alt="Example Image"
class="image">
  </body>
</html>
```

Dots Per Inch (DPI)

dpi

- Represents the **number of dots per inch**.
- Screens typically contains 72 or 96 dots per inch, but the dpi for printed documents is usually much greater.




```
<html>
<head>
    <title>Image DPI Example</title>
    <style>
        img {
            width: 400px;
            height: 300px;
            dpi:300;
        }
    </style>
</head>
<body>
    <h1>Image DPI Example</h1>
    <p>This image will have a DPI of 300:</p>
    <img src="flower1.jpg" alt="A beautiful
image">
</body>
</html>
```

“and” Operator

- When using media queries, we can use logical operators such as “**and**" to **combine multiple conditions**

```
<html>
  <head>
    <style>
      /* Styles for all devices */
      body {
        font-size: 16px;
      }
/* Styles for devices with screens between 768px and 1024px wide */
   @media (min-width: 768px) and (max-width: 1024px) {
        body {
          font-size: 20px;
        }
      }
    </style>
  </head>
  <body>
    <p>Welcome to my responsive web page!</p>
  </body> </html>
```

Media Queries - Comma-separated lists

```
<html>
  <head>
        <style>
 /* Default styles for all heading elements */
      h1, h2, h3, h4, h5, h6 {
        font-family: sans-serif;
        font-weight: bold;
      }
/* Styles  heading elements on small screens */
      @media (max-width: 767px) {
        h1, h2, h3, h4, h5, h6 {
          font-size: 20px;
        }
      }
/* Styles for medium and large screens */
      @media (min-width: 768px) {
        h1, h2, h3, h4, h5, h6 {
          font-size: 24px;
        }
      }     </style>
```

- **Comma-separated lists** can be useful in RWD to **apply the same style rules to multiple selectors**, but **only under certain conditions.**

- For example, **apply different styles** to heading elements **(h1, h2, h3, etc.) depending on the screen size**.

```
  </head>
  <body>
    <h1>Heading 1</h1>
    <h2>Heading 2</h2>
    <h3>Heading 3</h3>
    <h4>Heading 4</h4>
    <h5>Heading 5</h5>
    <h6>Heading 6</h6>
  </body>
</html>
```

Breakpoints

- In CSS, breakpoints are used to **apply different styles** to a web page **based on the screen size of the device.**
- **The layout of the page changes to fit different screen sizes**, making it more accessible and user-friendly.

```
<html> <head> <style>
.example {
  padding: 20px;
  color: white;
}
/* Extra small devices (phones, 600px and down) */
@media only screen and (max-width: 600px) {
  .example {background: red;}
}
/* Small devices (portrait tablets and large
phones, 600px and up) */
@media only screen and (min-width: 600px) {
  .example {background: green;}
} /* Medium devices (landscape tablets, 768px and up) */
@media only screen and (min-width: 768px) {
  .example {background: blue;}
}
/* Large devices (laptops/desktops, 992px and up) */
@media only screen and (min-width: 992px) {
  .example {background: orange;}
}
/* Extra large devices (large laptops and desktops, 1200px and
up) */
@media only screen and (min-width: 1200px) {
  .example {background: pink;}
}
</style> </head>
<body>
<h2>Typical Breakpoints</h2>
<p class="example">Resize the browser window
to see how the background color of this
paragraph changes on different screen
sizes.</p>
</body>
</html>
```

Breakpoints Sizing elements

- In RWD, breakpoints are used to **define different screen sizes** at which the layout of a webpage will change.
- Sizing elements in RWD often involves using **breakpoints to set different sizes for different screen sizes.**
- For example, Adjusting the font size of a paragraph element (<p>) depending on the screen size.

```
<html>   <head>
    <style>
  /* Default font size for all <p> elements */
      p {
        font-size: 16px;
      }
/* Font size <p> elements on small screens */
      @media (max-width: 767px) {
        p {
          font-size: 14px;
        }
      }
```

```
/* Font size for <p> elements on medium and large screens */
      @media (min-width: 768px) {
        p {
          font-size: 18px;
        }
      }
    </style>
  </head>
  <body>
    <p>Sizing elements in RWD often involves using breakpoints to set different sizes for different screen sizes.
</p>
  </body>
</html>
```

Sizing elements - Relative Measurements Em, Rem

- In RWD design, **em** and **rem** are commonly used **units for sizing elements** relative to the font size of the parent element or the root element, respectively.

```
<html>
  <head>
    <meta charset="utf-8">
    <title>Responsive Font Sizes with em and
rem</title>
    <style>
/* Default font size for all elements */
      body {
        font-size: 16px;
      }

/* Font size for headings on small screens */
      @media (max-width: 767px) {
        h1, h2, h3 {
          font-size: 2em; /* 32px */
        }
      }
```

```
/* Font size for headings on medium and large screens */
      @media (min-width: 768px) {
        h1, h2, h3 {
          font-size: 2.5rem; /* 40px */
        }
      }
    </style>
  </head>
  <body>
```

- In CSS, the **rem** unit is only relative to the document's **root element**, while the **em** unit is only relative to the **immediate parent of the targeted element**

Percentages (Height,Width, Padding , Margin) in RWD

- **Percentages** are commonly used in **RWD** for sizing elements such as **width, height, padding, and margin.**

- When used appropriately, **percentages can allow these elements to adjust their size** based **on the size of the viewport or parent container**, resulting in a more flexible and responsive design.

1) Width and height: → **sets width or height of an element to a percentage value** will make it relative to the size of its parent element.

- Ex> setting the width of an **image to 50%** will make it take up **half of the width of its parent element**. Similarly, setting the **height of a div to 75%** will make it **take up 75% of the height** of its parent element.

2) Padding: → sets the padding of an element to a percentage value will make it relative to the width of the element itself.

For example, setting the padding of a **div to 10% will make the padding area equal to 10% of the width of the div.**

3) Margin: **Sets the margin of an element to a percentage** value will make it relative to the width of its parent element.

For example, setting the margin of a **div to 5% will create a margin equal to 5% of the width of its parent element.**

Percentages (Height,Width, Padding , Margin) in responsive web design

```
<html>
  <head>
    <meta charset="utf-8">
    <title>Using Percentages for Sizing Elements in Responsive Web Design</title>
    <style>
/* Set width of div to 50% its parent elnt */
      div {
        width: 50%;
        border: 1px solid black;
      }

     /* Set the height of an image to 75% of its parent element */
      img {
        height: 75%;
        border: 1px solid black;
      }
      /* Set the padding of a div to 10% of its own width */
      .box {
        padding: 10%;
        border: 1px solid black;
      }
      /* Set the margin of a div to 5% of its parent element */
      .spacer {
        margin: 5%;
        height: 100px;
        background-color: gray;
      }
    </style>
  </head>
  <body>
    <div>
      <img src="flower1.jpg" alt="Placeholder image">
    </div>
    <div class="box">
      <p>Percentages are commonly used in RWD for sizing elements such as width, height, padding, and margin.
</p>
    </div>
    <div class="spacer"></div>
  </body>
</html>
```

Responsive Web Design - Videos

- To add a video in CSS RWD (Responsive Web Design), we can use **<video>** tag and then apply CSS styles to it to make it responsive

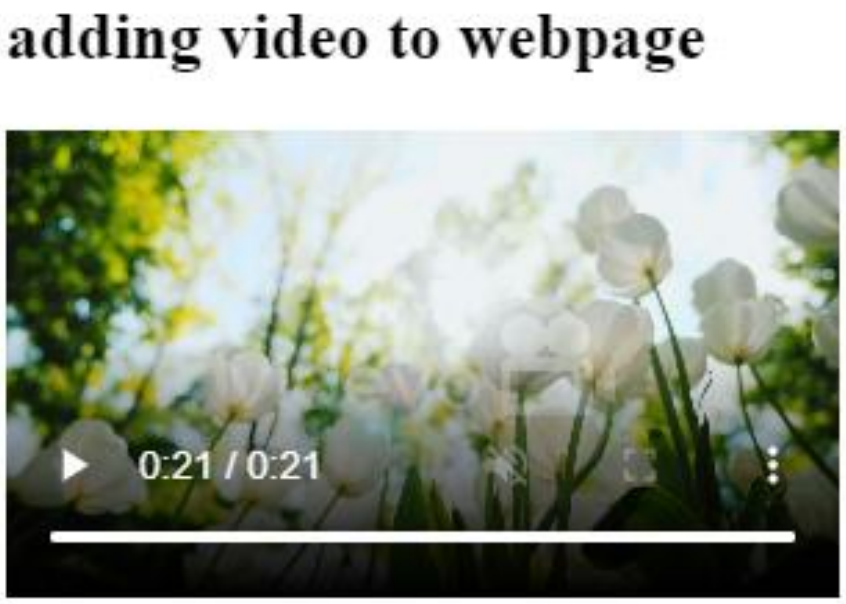
adding video to webpage


```
<html>
  <head>
    <meta charset="UTF-8">
    <title>Video Example</title>
    <style>
        video {
          width: 80%;
         height: 60%;
      }
    </style>
  </head>
  <body>
   <h2> adding video to webpage</h2>
    <video controls>
    <source src="flower.mp4" type="video/mp4">
      </video>

  </body>
</html>
```

Using The max-width Property

- If the **max-width property** is set to 100%, the video player will scale down if it has to, but never scale up to be larger than its original size:




```
<html>
<head>
<meta name="viewport" content="width=device-
width, initial-scale=1.0">
<style>
video {
  max-width: 100%;
  height: auto;
}
</style>
</head>
<body
<video width="400px" controls>
  <source src="flower.mp4" type="video/mp4">
</video>
<p>Resize the browser window to see how the size
of the video player will scale when the width is
less than 400px.</p>
</body>
</html>
```