CSS3 Media Queries

- Media query is one of the most important way to create Responsive Web Design.
- Media queries allow us to customize the presentation of our web pages for a specific range of devices like mobile phones, tablets, desktops, etc. without any change in markups.
- Media query is used to create responsive web design. It means that the view of web page differs from system to system based on screen or media types.
- A media query consists of a media type that can contain one or more expression which can be either true or false.
- Media queries are an excellent way to create responsive layouts. Using media queries, we can customize our website differently for users browsing on devices like smart phones or tablets without changing the actual content of the page.
- Media queries are case-insensitive.

Advantages of Media Queries

There are some advantages of media queries that includes :-

1) Only Valid CSS will be loaded.
2) Web Responsive Design.
3) Supported by all major browsers.
4) Improves SEO.

Media Queries Syntax

We can use media queries by following two ways :-

a) External Media Queries

a.1) By Using HTML <link> tag

We can apply media queries by using HTML <link> tag.

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <meta http-equiv="X-UA-Compatible" content="ie=edge">
        <title>Document</title>
        <link rel="stylesheet" href="css/mobile.css" media="screen and (max-width:480px)"/>
        <link rel="stylesheet" href="css/tablet.css" media="screen and (max-width:780px)"/>
        <link rel="stylesheet" href="css/desktop.css" media="screen and (max-width:10200px)"/>
</head>
<body>
        <h1>CSS Media Queries</h1>
</body>
</html>
```

a.2) By Using @import rule

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <meta http-equiv="X-UA-Compatible" content="ie=edge">
        <title>Document</title>
        <style>
                @import "smallscreen.css" only screen and (max-width: 680px);
        </style>
</head>
<body>

</body>
</html>
```

b) Internal Media Queries – By using CSS

We can apply media queries by using CSS also.

b.1) with "only" keyword

```
@media only screen and (max-width:420px)
{
        /* CSS Styles To Apply */
}
```

b.2) without "only" keyword

```
@media screen and (max-width:420px)
{
        /* CSS Styles To Apply */
}
```

Understanding Breakpoints & Expression

a) Breakpoints

- It is the point at which different styles are applied to a webpage depending on the viewport.
- Set breakpoints as determined by the content on the page.
- It is set to understand the code & syntax of how media query detects viewport size.

a.1) Choosing Breakpoints

- Don't define breakpoints based on device classes.
- Defining breakpoints based on specific devices, products, brand names, or operating systems that are in use today can result in a maintenance nightmare.
- Instead, the content itself should determine how the layout adjusts to its container.

a.2) Few recommendations are used in order to apply media queries: -

1) Create breakpoints based on content, not based on specific devices, products, or brands.
2) Design for the smallest mobile device first; then progressively enhance the experience as more screen real estate becomes available.
3) Keep lines of text to a maximum of around 70 or 80 characters.

Design the content to fit on a small screen size first, then expand the screen until a breakpoint becomes necessary. This allows you to optimize breakpoints based on content and maintain the least number of breakpoints possible.

We have different breakpoints according to website :-

1) Desktops & Laptops -- +960px
2) Tablets -- 960px - 760px
3) Mobile Phones -- 480px - 240px

Example

```
@media screen and (max-width:420px){
        /* CSS Styles To Apply */
}
```

Here, (max-width : 420px) is a breakpoint.

b) Expressions

- A logical expression is used to test whether a viewport has reached a particular breakpoint.
- The logical expression includes the name of a media query feature, a characteristic of the environment, and a breakpoint value to be tested.
- If the logical expression evaluates to “true”, the media query applies those styles followed by it.

Example

```
@media screen and (max-width:420px) {
        /* CSS Styles to Apply */
}
```

Here, @media screen and (max-width:420px) is an Expression.

Media Types for Media Queries

- Media types describe the general category of a device. Except when using the "not" or "only" logical operators.
- Below there are several media types that we can use with media queries: -
  1) all -- Intended for speech synthesizers.
  2) print -- Intended for paged material and documents viewed on a screen in print preview mode.
  3) screen -- Use this for Color computer monitor screens.
  4) speech -- Intended for speech synthesizers.

Targeting Media Types for Media Queries

- Media types describe the general category of a device.
- We can target media types by the following ways: -

```
1) @media screen { /*CSS Styles*/ }
2) @media print, screen { /*CSS Styles*/ }
```

Media Features for Media Queries

- The Media features are used to check the capabilities of the device (dimensions, resolutions etc.) using an expression and then accordingly apply the CSS style rules.
- Media features describe specific characteristics of the user agent, output device, or environment

There are many features of media query which are listed below :-

1) any-hover

The any-hover CSS media feature can be used to test whether any available input mechanism can hover over elements.

Syntax

```
@media(any-hover : value)
{
        /*CSS Styles*/
}
```

where,
value = none, hover

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
        <title>Document</title>
        <style>
                @media (any-hover: hover) {
                        a:hover {
                                background: yellow;
                        }
                }
        </style>
</head>
<body>
        <a href="#">Hover Me</a>
</body>
</html>
```

2) any-pointer

The any-pointer CSS media feature tests whether the user has any pointing device (such as a mouse).

Syntax

```
@media(any-pointer : value)
{
        /*CSS Styles*/
}
```

where,
value = none, coarse(less accurate), fine(more accurate).

3) aspect-ratio

The aspect-ratio CSS media feature can be used to test the aspect ratio of the viewport.

Syntax

```
@media(aspect-ratio : value)
{
        /*CSS Styles*/
}
```

Example

```
<!DOCTYPE html>
<html>
  <head>
  <meta charset="UTF-8">
    <style>
     input[type="checkbox"]:checked {
          background: yellowgreen;
          }
          @media (any-pointer: fine) {
                    input[type="checkbox"] {
                              -moz-appearance: none;
                              -webkit-appearance: none;
                              appearance: none;
                              width: 15px;
                              height: 15px;
                              border: 2px solid blue;
                    }
          }
          @media (any-pointer: coarse) {
                    input[type="checkbox"] {
                              -moz-appearance: none;
                              -webkit-appearance: none;
                              appearance: none;
                              width: 30px;
                              height: 30px;
                              border: 2px solid red;
                    }
          }
    </style>
  </head>
  <body>
    <input id="test" type="checkbox" />
          <label for="test">Click me</label>
  </body>
</html>
```

4) color

- The "color" media feature is used for colored screens.
- The color CSS media feature can be used to test the number of bits per color component (red, green, blue) of the output device.
- If the device is not a color device, the value is zero.

Syntax

```
                    @media(color)
                    {
                              // CSS Styles
                    }
```

Example

```
<!DOCTYPE html>
<html>
  <head>
  <meta charset="UTF-8">
    <style>
                    p {
                              color: black;
                    }
                    @media (color) {
                    p {
                              color: #24ba13;
                    }
           }
    </style>
  </head>
  <body>
    <p>This text should be black on non-color devices, red on devices with a low number
of colors, and greenish on devices with a high number of colors.</p>
  </body>
</html>
```

5) **height**

The height CSS media feature can be used to apply styles based on the height of the viewport (or the page box, for paged media).

Syntax

```
@media(height : value)
{
        // CSS Styles
}
```

where,
value = length(in px / % etc).

Example

```
<!DOCTYPE html>
<html>
 <head>
 <meta charset="UTF-8">
  <style>
           body       {
                margin: 0px;
           }
           div        {
                border: 4px solid black;
                height: 100vh;
           }

  @media (height: 360px) {
           div        {
                background-color: tomato;
                padding: 10px;
                font-size: 25px;
                color: white;
           }
     }
     @media (height: 500px) {
           div
           {
                background-color: teal;
                padding: 10px;
                font-size: 35px;
                color: white;
           }
     }
  </style>
 </head>
 <body>
     <div>Resize Browser</div>
 </body>
</html>
```

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
        <title>Document</title>
        <style>
                p
                {
                        color: red;
                }
                @media screen and (max-height: 400px) {
                        p
                        {
                                color: blue;
                                font-size: 40px;
                        }
                }
        </style>
</head>
<body>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ut veritatis fugiat
quos beatae repellendus ducimus quod culpa, cum qui incidunt inventore temporibus
ipsum magnam dolorum. Cupiditate, id nemo, minus magnam rem culpa perspiciatis
iusto facere excepturi officia ipsam soluta aliquid saepe dolore ea? Eum saepe obcaecati
quae unde, non natus fugiat dolorem suscipit nostrum rerum dolores, aspernatur
praesentium, consequuntur cupiditate? Minima dolore aliquam aliquid quaerat
laboriosam enim quia quam incidunt laborum doloribus veritatis placeat exercitationem,
nostrum maxime! Modi similique id in ducimus veritatis harum reiciendis aut. Velit,
necessitatibus sequi magni quasi id sapiente consectetur magnam perferendis
architecto ut, aliquid consequuntur?</p>
</body>
</html>
```

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
        <title>Document</title>
        <style>
                p
                {
                        color: red;
                }
                @media screen and (max-width: 800px) {
                        p
                        {
                                color: blue;
                                font-size: 40px;
                        }
                }
        </style>
</head>
<body>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ut veritatis fugiat quos beatae repellendus ducimus quod culpa, cum qui incidunt inventore temporibus ipsum magnam dolorum. Cupiditate, id nemo, minus magnam rem culpa perspiciatis iusto facere excepturi officia ipsam soluta aliquid saepe dolore ea? Eum saepe obcaecati quae unde, non natus fugiat dolorem suscipit nostrum rerum dolores, aspernatur praesentium, consequuntur cupiditate? Minima dolore aliquam aliquid quaerat laboriosam enim quia quam incidunt laborum doloribus veritatis placeat exercitationem, nostrum maxime! Modi similique id in ducimus veritatis harum reiciendis aut. Velit, necessitatibus sequi magni quasi id sapiente consectetur magnam perferendis architecto ut, aliquid consequuntur?</p>
</body>
</html>
```

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
        <title>Document</title>
        <style>
                p
                {
                        color: red;
                }
                @media screen and (min-height: 600px) {
                        p
                        {
                                color: blue;
                                font-size: 40px;
                        }
                }
        </style>
</head>
<body>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ut veritatis fugiat
quos beatae repellendus ducimus quod culpa, cum qui incidunt inventore temporibus
ipsum magnam dolorum. Cupiditate, id nemo, minus magnam rem culpa perspiciatis
iusto facere excepturi officia ipsam soluta aliquid saepe dolore ea? Eum saepe obcaecati
quae unde, non natus fugiat dolorem suscipit nostrum rerum dolores, aspernatur
praesentium, consequuntur cupiditate? Minima dolore aliquam aliquid quaerat
laboriosam enim quia quam incidunt laborum doloribus veritatis placeat exercitationem,
nostrum maxime! Modi similique id in ducimus veritatis harum reiciendis aut. Velit,
necessitatibus sequi magni quasi id sapiente consectetur magnam perferendis
architecto ut, aliquid consequuntur?</p>
</body>
</html>
```

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
        <title>Document</title>
        <style>
                p
                {
                        color: red;
                }
                @media screen and (min-width: 600px) {
                        p
                        {
                                color: blue;
                                font-size: 40px;
                        }
                }
        </style>
</head>
<body>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ut veritatis fugiat
quos beatae repellendus ducimus quod culpa, cum qui incidunt inventore temporibus
ipsum magnam dolorum. Cupiditate, id nemo, minus magnam rem culpa perspiciatis
iusto facere excepturi officia ipsam soluta aliquid saepe dolore ea? Eum saepe obcaecati
quae unde, non natus fugiat dolorem suscipit nostrum rerum dolores, aspernatur
praesentium, consequuntur cupiditate? Minima dolore aliquam aliquid quaerat
laboriosam enim quia quam incidunt laborum doloribus veritatis placeat exercitationem,
nostrum maxime! Modi similique id in ducimus veritatis harum reiciendis aut. Velit,
necessitatibus sequi magni quasi id sapiente consectetur magnam perferendis
architecto ut, aliquid consequuntur?</p>
</body>
</html>
```

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
        <title>Document</title>
        <style>
                p
                {
                        color: red;
                }
                @media screen and (width: 600px) {
                        p
                        {
                                color: blue;
                                font-size: 40px;
                        }
                }
        </style>
</head>
<body>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ut veritatis fugiat
quos beatae repellendus ducimus quod culpa, cum qui incidunt inventore temporibus
ipsum magnam dolorum. Cupiditate, id nemo, minus magnam rem culpa perspiciatis
iusto facere excepturi officia ipsam soluta aliquid saepe dolore ea? Eum saepe obcaecati
quae unde, non natus fugiat dolorem suscipit nostrum rerum dolores, aspernatur
praesentium, consequuntur cupiditate? Minima dolore aliquam aliquid quaerat
laboriosam enim quia quam incidunt laborum doloribus veritatis placeat exercitationem,
nostrum maxime! Modi similique id in ducimus veritatis harum reiciendis aut. Velit,
necessitatibus sequi magni quasi id sapiente consectetur magnam perferendis
architecto ut, aliquid consequuntur?</p>
</body>
</html>
```

Targeting media features for Media Queries

Media features describe the specific characteristics of a given user agent, output device, or environment.

We can target media features by the following ways :-

1) @media (hover: hover) { // CSS Code }
2) @media (max-width: 12450px) { // CSS Code }

Logical operators in Media Queries

The logical operators are used to compose a complex media query.

Below are the list of logical operators that are used with media queries :-

1. and
2. not
3. only
4. comma

1) and

- The 'and' operator combines one media type with other media types.
- The 'and' operator combines one media type with other media properties(media features) like width, heights etc.
- The 'and' operator combines one media property(media features) with multiple media properties(media features)
- The 'and' operator applies both 'media types' / 'media features'.

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
        <title>Document</title>
        <style>
                p {
                        padding:10px;
                        border:2px solid brown;
                        font-size:2em;
                }

                @media (min-width: 600px) and (max-width: 1000px) {
                        p {
                                background:orangered;
                                color: white;
                        }
                }
        </style>
</head>
<body>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolorum quo
adipisci consectetur cupiditate odio quas esse nulla similique error illum voluptates,
quod officiis tempora autem vitae molestiae vel rem deleniti.</p>
</body>
</html>
```

2) not

- The 'not' operator is the negation operator.
- The not operator is used to negate a media query, returning true if the query would otherwise return false.
- If present in a comma-separated list of queries, it will only negate the specific query to which it is applied.
- If we use the not operator, we must also specify a media type.
- When the entire media query is true, this operator reverts the expression to false. Similarly, if the media query is false, then this operator reverts the expression to true.

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
        <title>Document</title>
        <style>
                p {
                        padding:10px;
                        border:2px solid brown;
                        font-size:2em;
                }

                @media not all and (max-width: 700px){
                        p {
                                background:orangered;
                                color: white;
                        }
                }
        </style>
</head>
<body>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolorum quo
adipisci consectetur cupiditate odio quas esse nulla similique error illum voluptates,
quod officiis tempora autem vitae molestiae vel rem deleniti.</p>
</body>
</html>
```

3) only

- In order to specify the media type then we have to use the 'only' operator.
- The only operator is used to apply a style only if an entire query matches, and is useful for preventing older browsers from applying selected styles.
- When not using only, older browsers would interpret the query screen and (max-width: 500px) simply as screen, ignoring the remainder of the query, and applying its styles on all screens.

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Document</title>
    <style>
        p {
            padding:10px;
            border:2px solid brown;
            font-size:2em;
        }

        @media only screen and (max-width: 700px){
            p {
                background:orangered;
                color: white;
            }
        }
    </style>
</head>
<body>
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolorum quo
adipisci consectetur cupiditate odio quas esse nulla similique error illum voluptates,
quod officiis tempora autem vitae molestiae vel rem deleniti.</p>
</body>
</html>
```

4) comma

- The comma(,) operator is like logical OR operator.
- The comma(,) operator separates each media query from other media queries separated by a Comma(,).
- Commas are used to combine multiple media queries into a single rule.
- By comma(,) operator if any one of the media query which is separated by comma is true, then the entire media query is true and the same rule will be applicable to all the comma-separated media queries.

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
        <title>Document</title>
        <style>
                p {
                        padding:10px;
                        border:2px solid brown;
                        font-size:2em;
                }

                @media (max-width: 600px),(min-width: 1000px) {
                        p {
                                background:orangered;
                                color: white;
                        }
                }
        </style>
</head>
<body>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolorum quo
adipisci consectetur cupiditate odio quas esse nulla similique error illum voluptates,
quod officiis tempora autem vitae molestiae vel rem deleniti.</p>
</body>
</html>
```

Creating complex Media Queries using logical operators

Complex media queries can be created with the combination of :-

1) and
2) not
3) comma(,)
4) only

a) Combining two breakpoints

a.1) By Using 'and' Operator

- Example 1

```
@media (min-width: 600px) and (max-width: 1000px) {
        // CSS Code
}
```

- Example 2

```
@media (min-width: 30em) and (orientation: landscape) {
        // CSS Code
}
```

- Example 3

```
@media screen and (min-width: 30em) and (orientation: landscape) {
        // CSS Code
}
```

a.2) By Using 'comma' Operator

```
@media (min-width: 600px) , (max-width: 1000px) {
        // CSS Code
}
```

b) Combining multiple types

```
@media screen , print and (min-width: 30em) and (orientation: landscape) {
        // CSS Code
}
```

c) Negating a feature with "not"

```
@media not all and (max-width: 700px){
        // CSS Code
}
```

CSS3 Media Query Orientation

- The orientation CSS media feature can be used to test the orientation of the viewport.
- In Portrait mode, the height size is greater than the width size.
- The page or image in Landscape mode has greater width size than the height.

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
        <title>Document</title>
        <style>
                p {
                        padding:10px;
                        border:2px solid brown;
                        font-size:2em;
                }
                @media screen and (orientation: landscape){
                        p                       {
                                color: blue;
                                font-size: 40px;
                        }
                }
                @media screen and (orientation: portrait){
                        p                       {
                                color: orange;
                                font-size: 50px;
                        }
                }
        </style>
</head>
<body>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolorum quo
adipisci consectetur cupiditate odio quas esse nulla similique error illum voluptates,
quod officiis tempora autem vitae molestiae vel rem deleniti.</p>
</body>
</html>
```

Difference between "screen" & "only screen"

1) "screen"

- It is used to set the screen size of media query.
- The screen size can be set by using max-width and min-width.
- The screen size is differ from screen to screen.

2) "only"

The only keyword is used to prevent older browsers that do not support media queries with media features from applying the specified styles.

Media Queries for mobile, tablets & desktops

- Media queries are an excellent way to create responsive layouts.
- Using media queries you can customize your website differently for users browsing on devices like smart phones or tablets without changing the actual content of the page.

Below there is a list of media queries for standard devices :-

1) For Smartphones

1.1) For Portrait & Landscape

Examples

```
@media screen and (min-width: 320px) and (max-width: 480px){
   /* CSS styles */
}
Or,

@media screen and (device-width: 360px) and(device-height: 640px) {
        /* HTC One X / LG / Samsung Galaxy S6 */
}
Or,

@media only screen and (min-device-width: 320px) and(max-device-width:
480px) and (orientation:landscape/portrait) { /*For Iphone 4*/}
Or,

@media only screen and (min-device-width: 320px) and(max-device-width:
568px) and (orientation:landscape/portrait) { /*For Iphone 5*/}
Or,

@media only screen and (min-device-width: 375px) and(max-device-width:
667px) and (orientation:landscape/portrait) { /*For Iphone 7*/}
Or,

@media only screen and (min-device-width: 414px) and(max-device-width:
736px) and (orientation:landscape/portrait) { /*For Iphone 7plus*/}
```

1.2) For Portrait

Examples

```
@media screen and (max-width: 320px){
    /* styles */
}
or,

@media only screen and (max-width: 480px){
    /* styles */
}
or,

@media (max-width: 480px) {
    /* styles */
}
```

1.3) For Landscape

Example

```
@media screen and (min-width: 321px){
    /* styles */
}
```

1.4) For Large Screen Mobiles

Example

```
@media only screen and (max-width: 767px){
    /* styles */
}
```

2) For Tablets & Ipads

2.1) For Portrait & Landscape

```
@media screen and (min-width: 768px) and (max-width: 1024px){
   /* styles */
}
Or,

@media only screen and (max-width: 1024px){
   /* styles */
}
```

2.2) For Portrait

```
@media screen and (min-width: 768px){
   /* styles */
}
```

2.3) For Landscape

```
@media screen and (min-width: 1024px){
   /* styles */
}
```

2.4) Miscellaneous

```
/* Nexus 10 - Landscape and Portrait */
@media screen and (device-width: 800px) and(device-height: 1280px) and
(orientation:landscape/portrait) { /*CSS Style*/}

/* Nexus 7 - Landscape and Portrait */
@media screen and (device-width: 600px) and(device-height: 960px) and
(orientation:landscape/portrait) { /*CSS Style*/}

/* Surface Pro 3  - Landscape and Portrait */
@media screen and (device-width: 960px) and(device-height: 1440px) and
(orientation:landscape/portrait) { /*CSS Style*/}
```

```
/* BlackBerry Z10 - Landscape and Portrait */
@media only screen and (min-device-width: 384px) and(max-device-width: 640px) and (orientation:landscape/portrait) { /*CSS Style*/}

/* Kindle Fire - Landscape and Portrait */
@media only screen and (min-device-width: 600px) and(max-device-width: 1024px) and (orientation:landscape/portrait) { /*CSS Style*/}

/* iPad  - Landscape and Portrait */
@media only screen and (min-device-width: 768px) and(max-device-width: 1024px) and (orientation:landscape/portrait) { /*CSS Style*/}

/* iPad Mini  - Landscape and Portrait */
@media only screen and (min-device-width: 768px) and(max-device-width: 1024px) and (orientation:landscape/portrait) { /*CSS Style*/}

/* iPad Mini 2-4  - Landscape and Portrait */
@media only screen and (min-device-width: 1536px) and(max-device-width: 2048px) and (orientation:landscape/portrait) { /*CSS Style*/}

/* iPad Pro - Landscape and Portrait */
@media only screen and (min-device-width: 2048px) and(max-device-width: 2732px) and (orientation:landscape/portrait) { /*CSS Style*/}
```

3) For Desktops & Laptops

```
@media screen and (min-width: 1224px){
   /* styles */
}
```

4) For Larger Screens

```
@media screen and (min-width: 1824px){
   /* styles */
}
```