Week 2:Advanced CSS Topics

Outline:

- **CSS Units**
- **Media Queries**
- **CSS Selectors**
- **Animation & Key Frames**
- **Creating Transitions in CSS**
- **Transform property in CSS**
- **CSS Grid**

1.CSS Units: em, rem, vh and vw units + Responsive design Explained

What Is Responsive Design?

Have you ever noticed that websites nowadays adjust themselves according to the resolution of your device(Smartphone, tablet, or desktop computer)? Isn't it amazing that a website is automatically changing its height and width according to the size of the user's screen? This is possible because of the responsive design. Let's dive deep into responsive design.

- Responsive design is a way for a web developer to make a website adapt to all the devices and resolutions.
- Endless new resolutions and devices are challenging to support separately for a web developer; therefore, responsive design is the best approach to develop a website that can adjust itself according to the screen size.
- Responsive design is a necessity and not a luxury anymore!

Various Ways To Achieve Responsive Design :

1. By using rem/vh/vw units over pixels.
2. By using max-width/max-height.
3. Using CSS Media Queries.
4. Setting up the viewport.

First of all, we are using the below CSS to create three headings:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Size Units</title>
    <style>
        h1{
            text-align: center;
            font-size: 10px;
        }
        .container{
            font-size: 10px;
        }
    </style>
</head>
<body>
    <div class="container">
   <h1 id="first">This is first heading</h1>
   <h1 id="second">This is second heading </h1>
   <h1 id="third">This is third heading</h1>
</div>
</body>
</html>
```

Here, we have just created three headings and aligned them to the center. Now, we will use these heading to understand the concept of em, rem,vh, and vw.

em-

- Font-sizes are inherited relative to the parent element.

- 10em means ten times of the parent element.

Use the CSS given below :

```
<style>
    h1{
        text-align: center;
        font-size: 10px;
    }
    .container{
        font-size: 10px;
    }
    #first{
        font-size: 10em;
    }
</style>
```

This is first heading

In the above image, you can see that the font-size of the first heading is changed. Earlier the font-size was 10px because <h1> inherited this size from its parent element, i.e., container. Now, the font-size of <h1> has changed to 100px because 10 em means ten times of the parent element so 10*10=100px.

rem-

- Font-size gets set according to the font-size of the root element.
- In general, <html> is the root element.
- In rem, "r" stands for "root."

Use the CSS given below:

```
<style>
    html{
        font-size: 7px;
    }
    h1{
        text-align: center;
        font-size: 10px;
    }
    .container{
        font-size: 10px;
    }
    #first{
        font-size: 10em;
    }
    #second{
        font-size: 10rem;
    }
</style>
```

In the above code, we have given the font-size of 7px to the <html>. Then, we have applied the font-size of 10rem to the second heading. Here is the output :

In the above image, you can see that the font size of the second heading has changed from 7px to 70px because 10rem is equal to 10 times the root element. You can verify the font-size with the help inspect element functionality of the chrome browser.

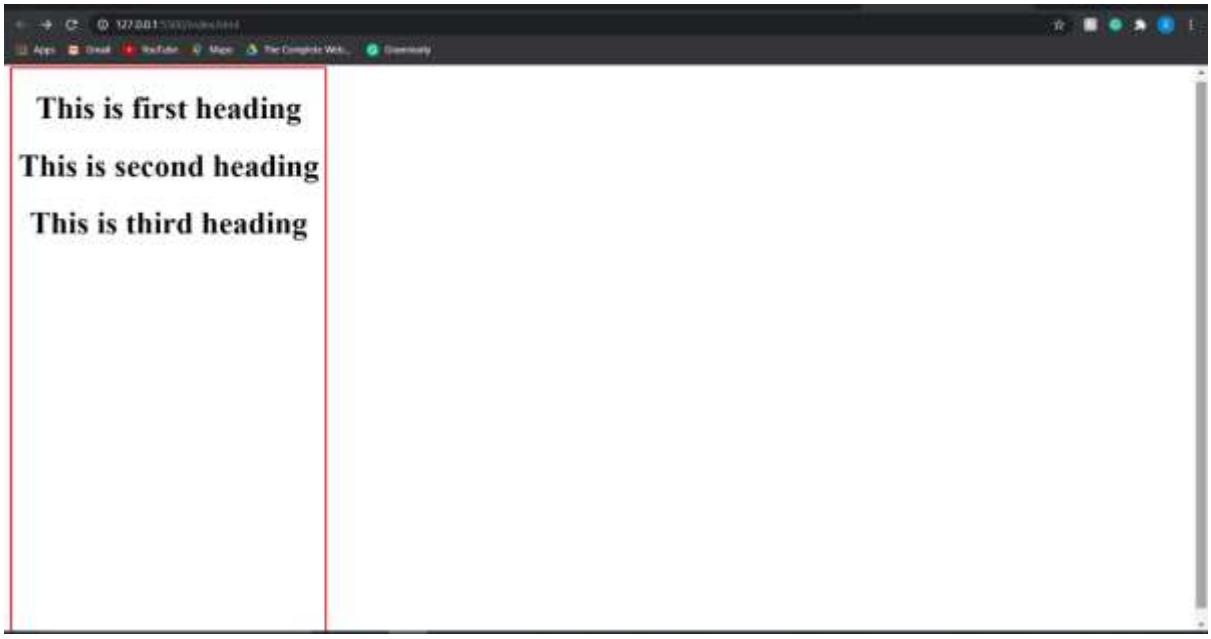
vh-

- It stands for viewport height.
- vh is equal to the 1/100 times of the viewport height.
- Example: Suppose height of the browser is 1000px, so 1vh is equaled to (1/100)*1000= 10px.

Use the CSS given below :

```
<style>
        html{
            font-size: 7px;
        }
        h1{
            text-align: center;
            font-size: 40px;
        }
        .container{
          border: 2px solid red;
          height: 100vh;
          width: 400px;
        }
        /* #first{
            font-size: 10em;
        }
        #second{
            font-size: 10rem;
        } */
    </style>
```

In the above code, we have made a border and assigned a height of 100vh to it. Here are the results:



In the above image, you can see that the border's height is changed to 100% of the viewport.

vw-

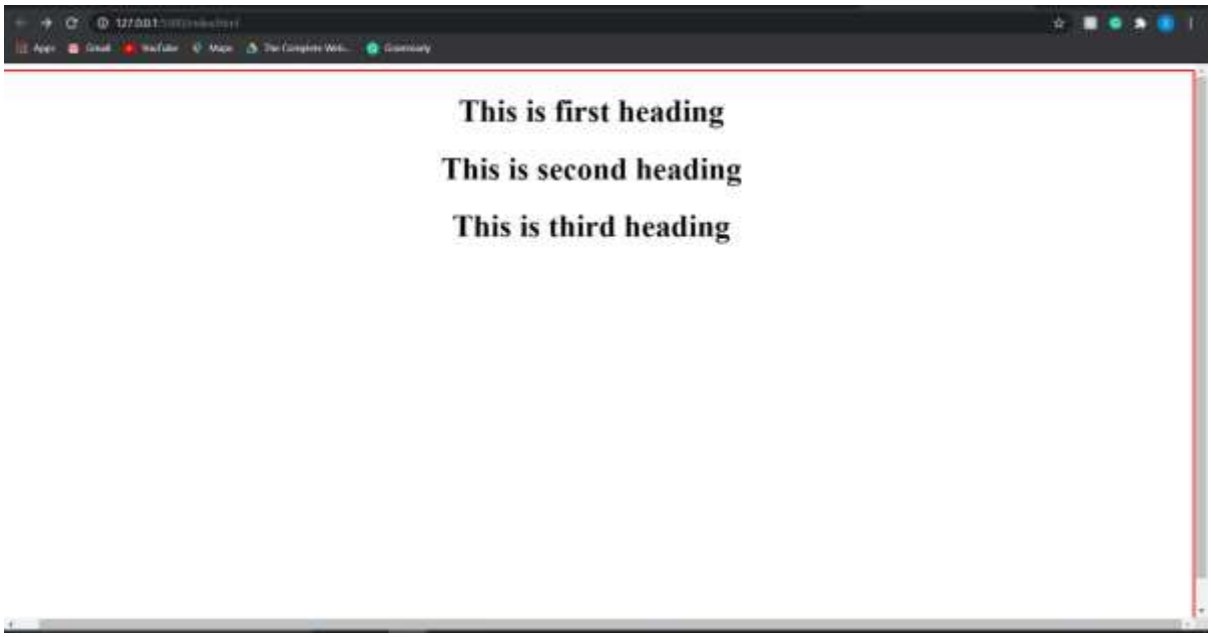
- It stands for viewport width.
- Similar to vh, vw is equal to the 1/100 times of the viewport width.
- Example: Suppose width of the browser is 1000px, so 1vw is equaled to (1/100)*1000= 10px.

Use the CSS given below:

```
<style>
        html{
            font-size: 7px;
        }
        h1{
            text-align: center;
            font-size: 40px;
        }
        .container{
          border: 2px solid red;
          height: 100vh;
          width: 100vw;
        }
```

```
    /* #first{
        font-size: 10em;
    }
    #second{
        font-size: 10rem;
    } */
</style>
```

In the above code, we have assigned a width of 100vw to the border. Here are the results:



https://webdesign.tutsplus.com/tutorials/comprehensive-guide-when-to-use-em-vs-rem--cms-23984

2. CSS Tutorial: Media Queries Explained

What Is Media Query?

- Media queries are used when we want to customize our website's presentation according to the user's screen size. With the help of media queries, you can display different markups based upon the device's general type(mobile, desktop, tablet).
- A media query is a logical operation. Whenever a media query becomes true, then the related CSS is applied to the target element.

Syntax Of Media Query :

A media query is composed of two things: media type and expression. A media query can contain one or more expressions.

Syntax :

```
@media media-type and (media-feature)
{
/* CSS Rules to be applies*/
}
```

Example:

```
@media screen and (max-width: 800px)
{
#contents{width:90%}
}
```

Let's understand this example :

- @media: A media query always starts with @media.
- Screen: It is the applicable media type.
- max-width**:** It is the media feature.
- #contents{width:90%} : It is the CSS to be applied when the conditions are met.

Now, we will spend some time understanding how to use media queries on a website.

First of all, we have used the below CSS to design four divs.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>CSS Media Query</title>
    <style>
        .box {
            text-align: center;
            background-color: rgba(255, 196, 0, 0.959);
            color: white;
            font-size: 50px;
        }
 </style>
</head>
<body>
    <div class="box" id="box1"> Windows</div>
```

```
    <div class="box" id="box2"> MacOS</div>
    <div class="box" id="box3"> Linux</div>
    <div class="box" id="box4">Android</div>
</body>
</html>
```

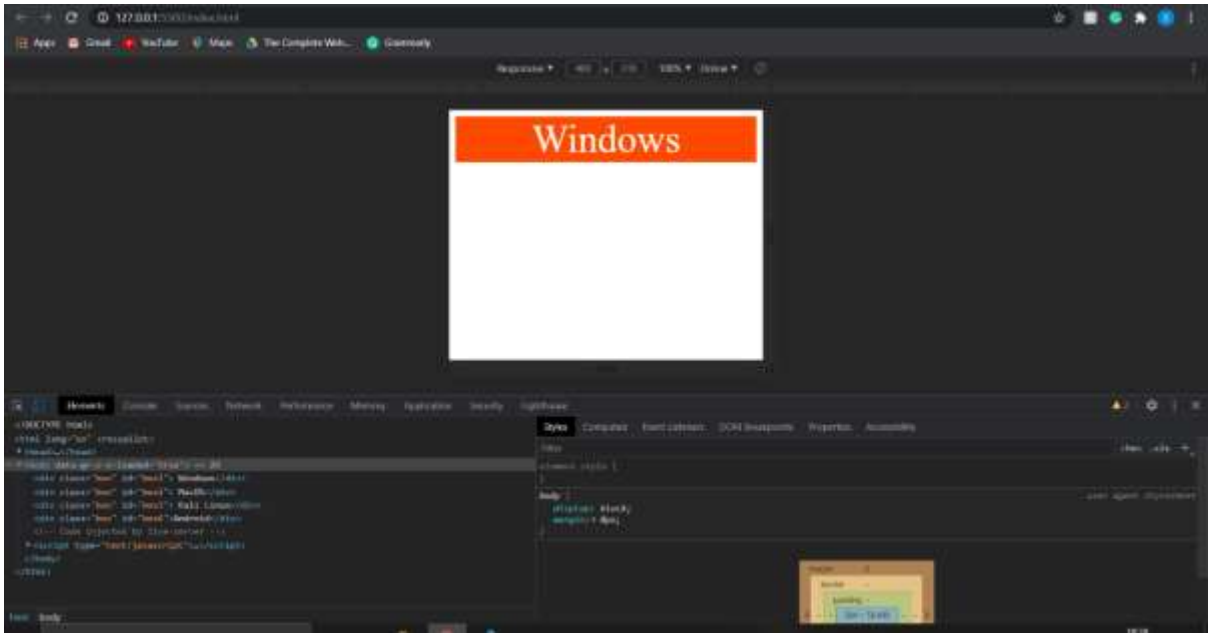
Now, we are changing the display value to display:none; and then we will apply the media queries. Here is the CSS :

```
<style>
        .box {
            text-align: center;
            background-color: rgba(255, 196, 0, 0.959);
            color: white;
            font-size: 50px;
             display: none;
        }
 </style>
```

Now, it's time to apply our very first media query. Here is the CSS used :

```
@media (min-width: 400px){
            #box1{
                display: block;
            }
        }
```

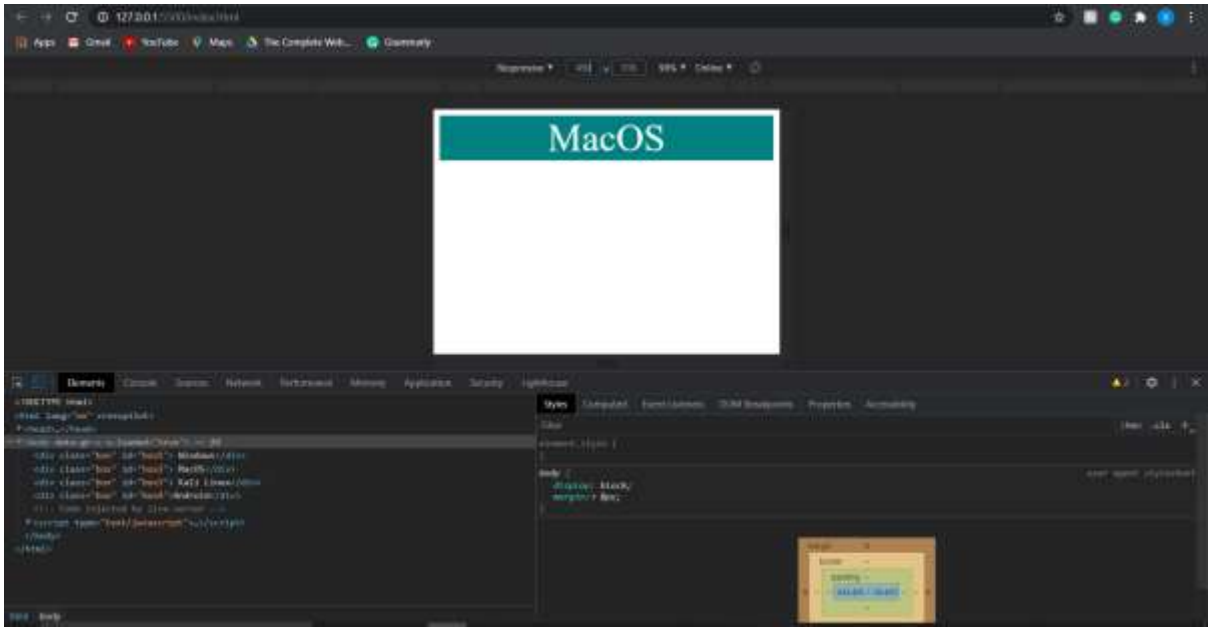
In the above code, we have applied the media query to id=box1. So, whenever the page's width is 400px or more than 400px, then the display value of the box1 will be changed from display:none; to display:block; and applied CSS will be visible to the user.



In the above image, I have used the chrome developer tools to change the page's width. You can see that the media query is applied when the page's width is set to 400px. So whenever the page width is 400px or more than 400px, then the media query for the box1 will be applied because we have set the minimum width to 400px. Similarly, we have applied a media query for the box2. Here is the CSS used:

```
/* @media (min-width: 400px){
        #box1{
            display: block;
        }
    } */
     @media (min-width: 450px) and (max-width: 600px){
        #box2{
            display: block;
            background-color: teal;
        }
    }
```

In the above code, we have set minimum width to 450px and maximum width to 600px. So the media query for the box2 will be applied whenever the screen's width is between 450px and 600px.



So, that's how you can easily include different media queries depending upon your requirements. Note that we have used and operator to combine two queries to be true. You can also use other operators such as not, only, and comma(,). So, this is all for this tutorial, and I hope you have got the basic knowledge of media queries in CSS.

CSS Tutorial: More on CSS Selectors

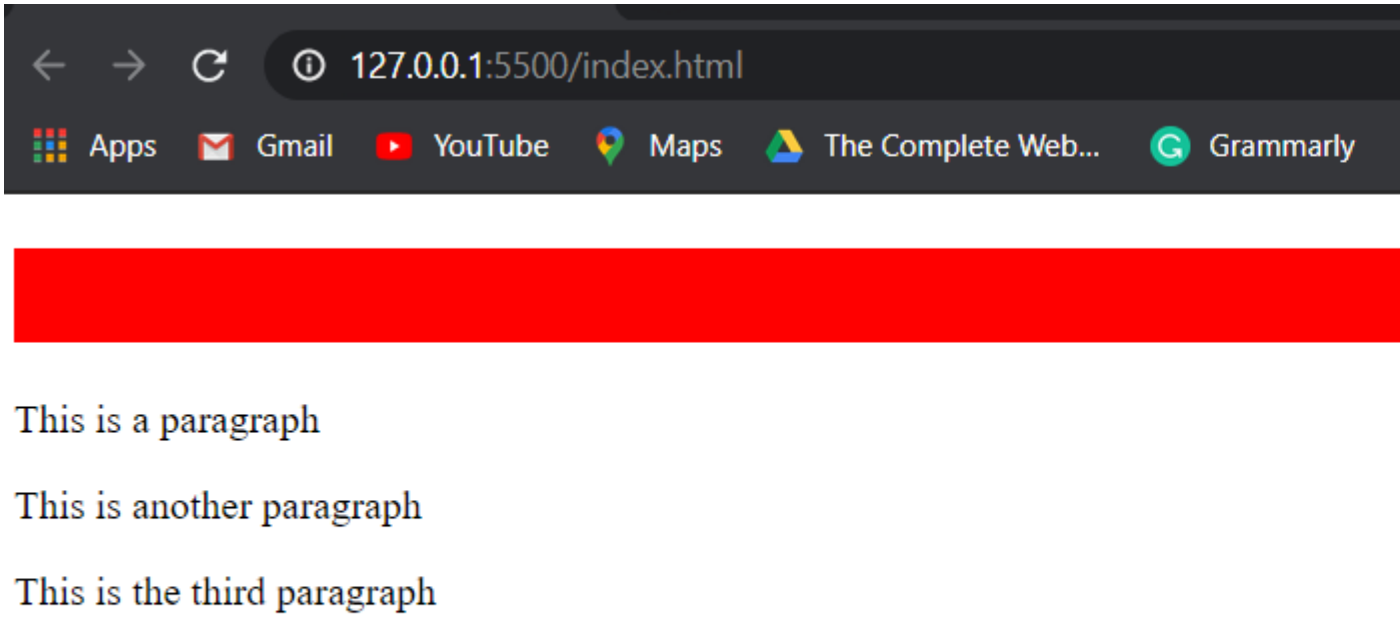
First of all, we have used the CSS given below to design some divs and paragraph tags.

```
<!DOCTYPE html>

<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Advanced CSS Selectors</title>
    <style>
        h1 {
            background-color: red;
            color: black;
            font-weight: bold;
            text-align: center;
```

```
        }
        div p {
            color: rgb(0, 0, 128);
            background-color: orange;
            font-weight: bold;
        }
    </style>
</head>
<body>
    <h1> This is more on selectors</h1>
    <div class="container">
        <div class="row">
            <p> This is a paragraph</p>
        </div>
        <p> This is another paragraph</p>
    </div>
    <p> This is the third paragraph</p>
</body>
</html>
```

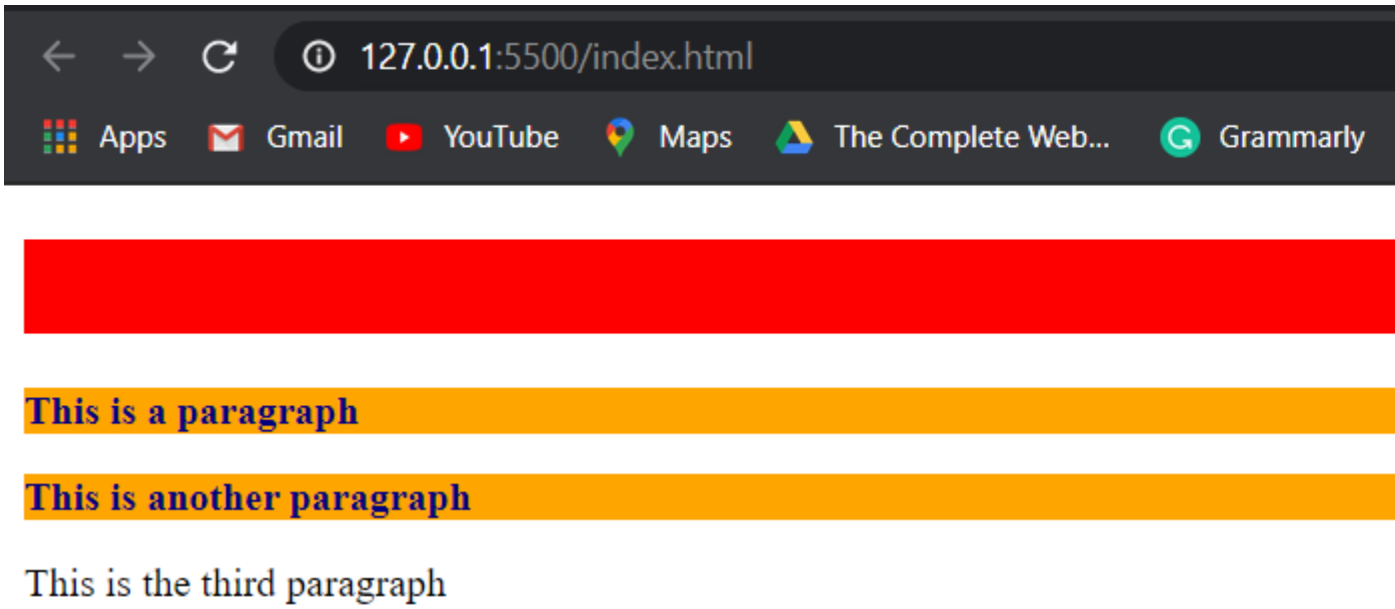
Output :



Now, let's suppose you want to add styling to all the paragraphs inside the div. What will you do? You can use the following CSS :

```
div p{
        color: rgb(0, 0, 128);
        background-color:orange;
        font-weight: bold;
    }
```

Here are the results :



In the above image, you can see that the CSS is applied to the two paragraphs, but it is not applied to the third paragraph. Why? Let me answer this simple question for you. The CSS is not applied to the third

paragraph because we have applied CSS only on the <p> tags inside div, and the third paragraph is not inside a div element.

This was very simple. Now let's move on to the next situation. What will you do if you want to target the <p> tags, which are the direct child of the div? So, whenever we want to target a direct child element, we use the following syntax :

element> element;

Example :

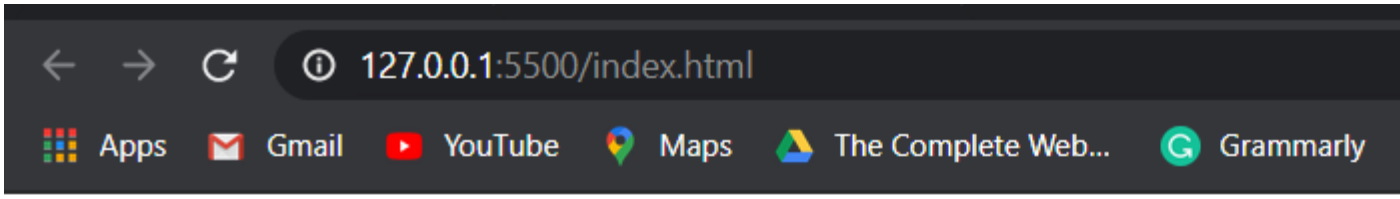
div>p

In the above example, the styling will be applied to all <p> elements which are the direct child of any div element. Let's understand this with the help of paragraph tags that we created at the starting of this tutorial. Here is the CSS used :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Advanced CSS Selectors</title>
    <style>
        h1 {
            background-color: red;
            color: black;
            font-weight: bold;
            text-align: center;
        }
        div p {
            color: rgb(0, 0, 128);
            background-color: orange;
            font-weight: bold;
        }
        div>p{
            background-color: lightblue;
            color:white;
        }
    </style>
</head>
<body>
    <h1> This is more on selectors</h1>
    <div class="container">
```

```
        <div class="row">
            <ul>
                <li class="item">
                    <p>This is a paragraph inside li and this is not a direct child of div. It will
not get affected.</p>
                </li>
            </ul>
            <p> This is the second paragraph and it will get affected because it is the
direct child of div.</p>
        </div>
        <p> This is the third pargraph and it will get affected because it is also a direct
child</p>
    </div>
    <p> This is the outer most paragraph and it will not get affected. </p>
</body>
</html>
```

Here are the results :



- **This is a paragraph inside li and this is not a direct child of div. It will not g**

This is the second paragraph and it will get affected because it is the direct child

This will get affected because it is also a direct child

This is the outer most paragraph and it will not get affected.

In the above image, you can see that the CSS is applied to the second and third paragraphs because they are the direct child of div. Paragraph inside <li> is the direct child of <li> and not of . Also, in the case of the outermost paragraph, the parent element is the body and not div; therefore, no styling is applied to it. Now, we will talk about one more type of CSS selectors, i.e., div+p.

div+p :

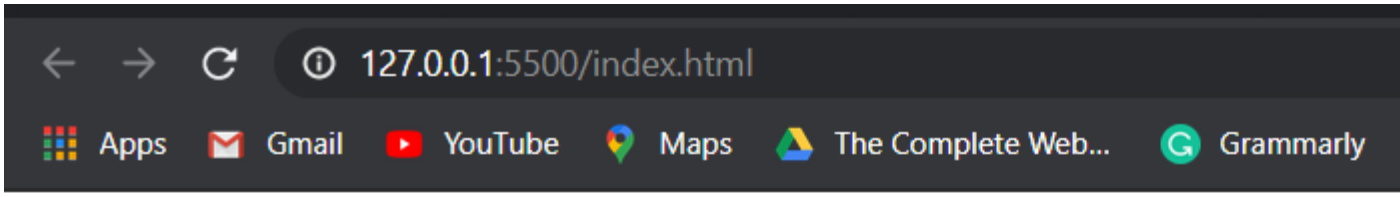
There might be situations where you want to select the <p> tags that are immediately after the elements. In such cases, we use the div+p selectors. Let's understand this with the help of example given below:

Html and CSS used:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Advanced CSS Selectors</title>
    <style>
        h1 {
            background-color: red;
            color: black;
            font-weight: bold;
            text-align: center;
        }
        /* div p {
            color: rgb(0, 0, 128);
            background-color: orange;
            font-weight: bold;
        } */
        /* div>p{
            background-color: lightblue;
            color:white;
        } */
       div+p{
          color: white;
          background-color:#D2302C;
       }
    </style>
</head>
<body>
    <h1> This is more on selectors</h1>
```

```
    <div class="container">
        <div class="row">
            <ul>
                <li class="item">
                    <p>This is a paragraph and it is not immediately after the div element
so no CSS will be applied to it.</p>
                </li>
            </ul>
            <p> This is the second paragraph and it is not immediately after the div
element so no CSS will be applied to it.</p>
        </div>
        <p> This is the third pargraph and it will get affected because it is immediately
afetr the div tag.</p>
    </div>
    <p> This is the outer most paragraph and it will also get affected because it is
immediately after the div tag. </p>
</body>
</html>
```

Output :



- This is a paragraph and it is not immediately after the div element so no CSS wi

This is the second paragraph and it is not immediately after the div element so no CSS

This is the third pargraph and it will get affected because it is immediately afetr the di

This is the outer most paragraph and it will also get affected because it is immediately

In the above image, you can see that the CSS is applied only to the third and outermost paragraphs because they are next to the element.

Week 2 - Day two

Homework

Read about browser compatibility for css attributes, take a look at the website https://caniuse.com/ and add a css property to see which browsers support it

m/css/css_font.asp
And web fonts https://www.w3schools.com/css/css3_fonts.asp

Fonts and a brief look at @font-face and importing fonts