

CSS Tutorial: em, rem, vh and vw units + Responsive design Explained | Web Development Tutorials #29

 codewithharry.com/videos/web-development-in-hindi-29

In this tutorial, we will discuss the responsive design and CSS units.

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.

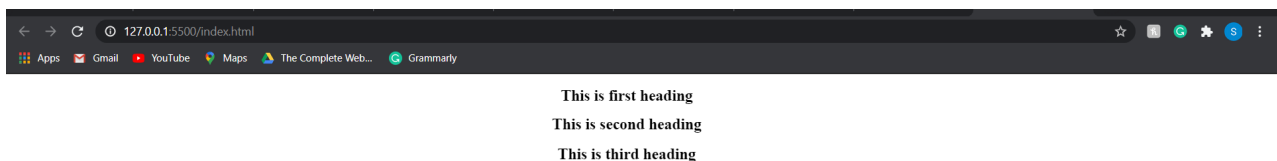
This tutorial will cover the use of CSS size units to make a responsive website. We will see other ways of responsive design in further tutorials. So, let's start our discussion on CSS size units. First of all, we are using the below CSS to create three headings:

```

<!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>

```

Output:



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>
```

Output:



This is first heading

This is second heading
This is third 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 $10 \times 10 = 100\text{px}$.

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 :



This is first heading

This is second heading

This is third heading

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 $(1/100) \times 1000 = 10\text{px}$.

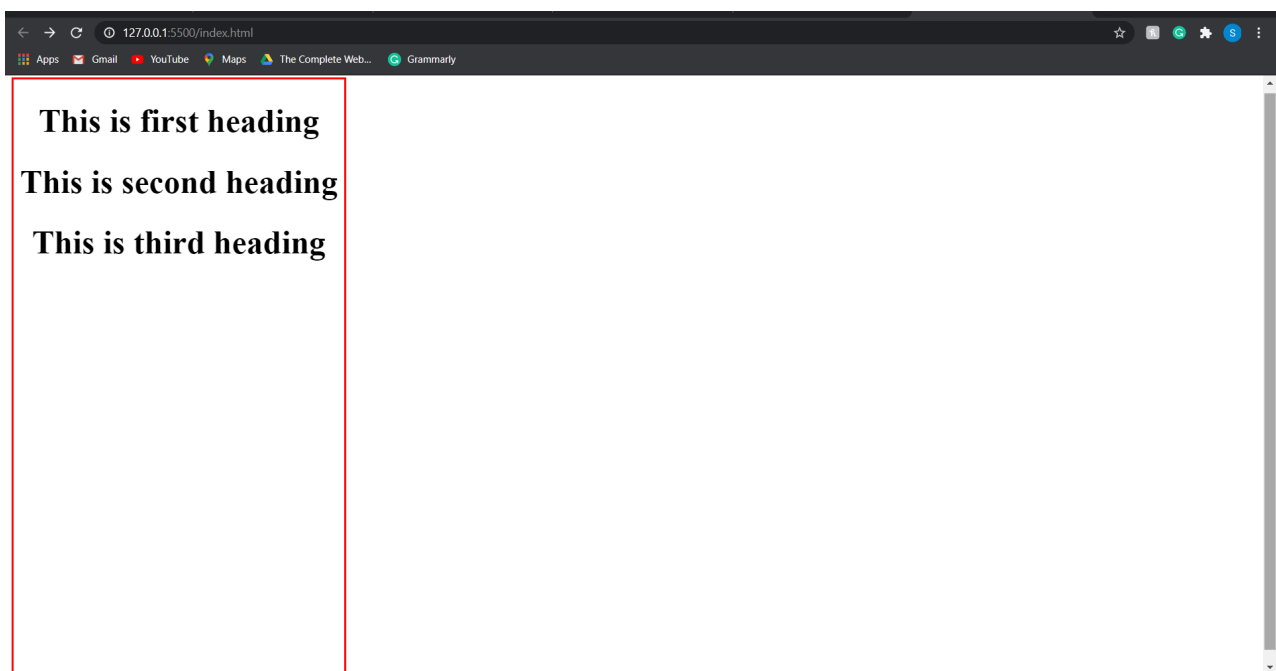
Use the CSS given below :

```

<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.

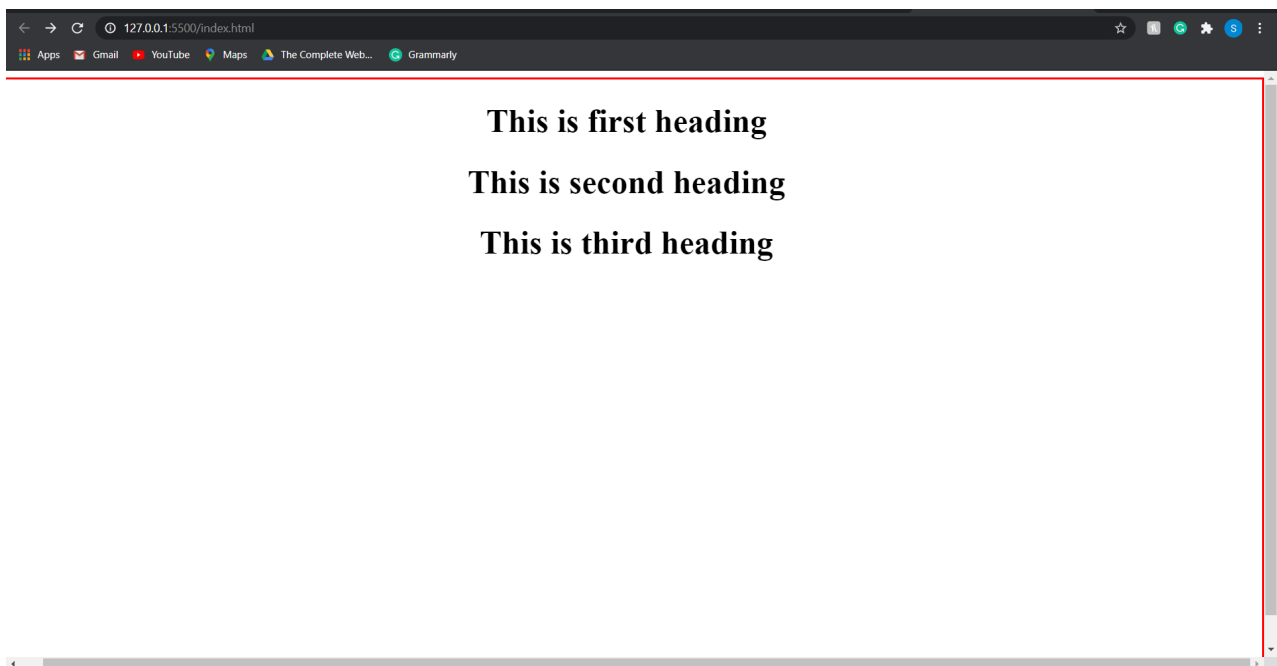
VW-

- 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:



In the above image, you can see that the border's width is changed to 100% of the viewport. This tutorial ends here, and I believe that all the concepts discussed in this tutorial are crystal clear to you.

Code as described/written in the video

```

<!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>Size units</title>
  <style>
    html{
      font-size: 25px;
    }
    .container{
      width:400px;
      /* height: 344px; */
      height: 100vh;
      width: 100vw;
      font-size: 10px;
      border :2px solid red;
    }
    h1{
      text-align: center;
    }
    #first{
      /* font-size: 3em;
      padding: 3em; */

    }
    #second{
      /* font-size: 3rem;
      padding: 3rem; */

    }

  </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>

```