

# Full Stack Software Development

**Course:** Introduction to Web Development

**Lecture On:** Introduction to CSS

**Instructor:** Siddhesh Prabhugaonkar

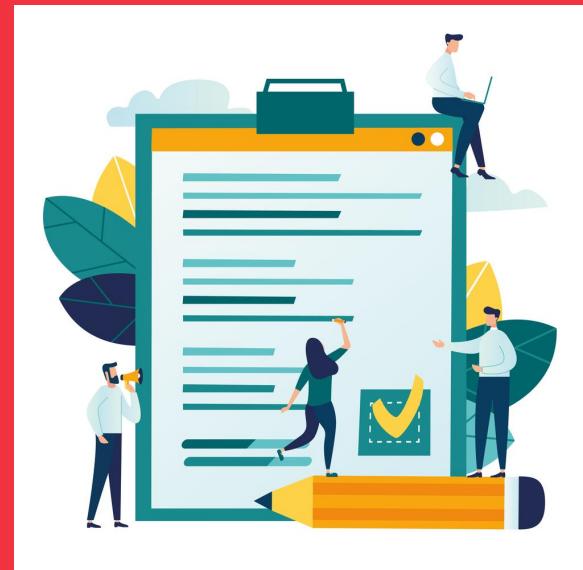


# In the previous class, we discussed...

- How forms are created in HTML
- HTML5

# Today's Agenda

- Introduction to CSS
- How to add CSS and style our website
- CSS height, width and unit
- CSS Colors, Text and Fonts
- CSS Backgrounds



# Introduction to CSS



- CSS stands for **Cascading Style Sheets**.
- It makes an HTML website presentable.
- It adds style to various HTML elements.
- It helps you to define how the elements should look, where they should be placed and whether they should be displayed or not.
- It also helps you to define how elements should look on different devices.

## class vs id



**class**

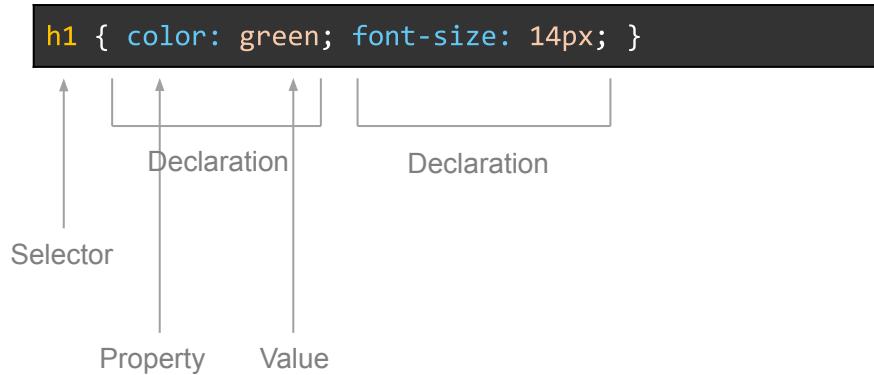
A group of people exhibiting the same properties



**id**

An individual with his/her own identity

## Syntax of CSS



- A selector can be an element itself such as **p**, **h1** or the **class** or the **id**.
- A property signifies the property of an element, such as the **color** of a text element.
- A value signifies the value that should be assigned to a property, such as **red** for the **color** property.

## Comment CSS Code

```
h1 {  
  color: green;  
  /*font-size: 14px;*/ ← Single-line comment  
}  
  
/*  
h2 {  
  color: red;  
  text-align: center; ← Multi-line comment  
}  
*/
```

**A CSS code can be commented out by adding /\* and \*/ to the code.**

## CSS Selectors

```
<p class="paragraph">This is a paragraph.</p>  
  
<h1 id="header1">This is a header.</h1>  
  
<span>This is a span.</span>
```

```
span { color: green; } /*Element selector*/  
  
.paragraph { color: green; } /* Class selector */  
  
#header1 { color: green; } /* ID Selector */  
  
* { color: green; } /* Universal Selector */
```

- **Element selectors** only require the element name.
- **Class selectors** require ‘.’ before the class name given to an element.
- **ID selectors** require ‘#’ before the ID name given to an element.
- **Universal selector** require ‘\*’. It signifies all the elements in a given file.

# Poll 1 (15 Sec)

What does CSS stand for?

1. Creative Styling Sheets
2. Colourful Style Sheets
3. Cascading Style Sheets
4. Class Style Sheet



# Poll 1 (Answer)

What does CSS stand for?

1. Creative Styling Sheets
2. Colourful Style Sheets
- 3. Cascading Style Sheets**
4. Class Style Sheet

## Poll 2 (15 Sec)

Which of the following is true about the priority of a CSS selector?

1. ID > Class > Element
2. Element > ID > Class
3. Element > Class > ID
4. None of the above

# Poll 2 (Answer)

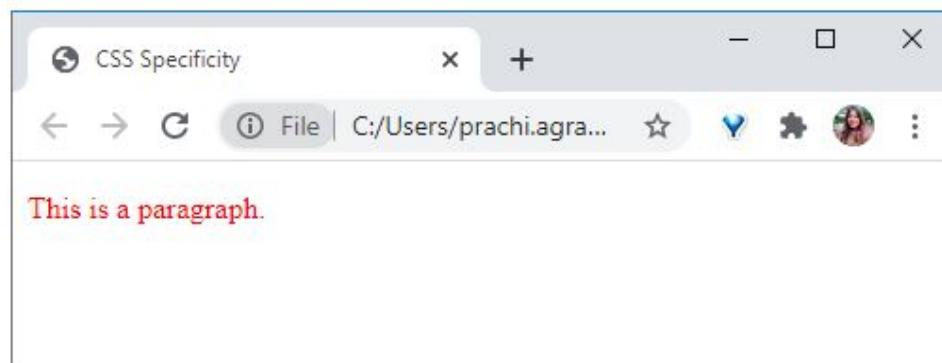
Which of the following is true about the priority of CSS selector?

1. **ID > Class > Element**
2. Element > ID > Class
3. Element > Class > ID
4. None of the above

## Priority CSS Selectors

```
<p id="para" class="paragraph">  
    This is a paragraph.  
</p>
```

```
p { color: blue; } /*Element selector*/  
  
.paragraph { color: green; } /* Class selector */  
  
#para { color: red; } /* ID Selector */  
  
p.paragraph { color: purple; }
```



# Poll 3 (15 Sec)

What will be the output of the following CSS code?

1. Red
2. Green
3. No colour

```
p {  
    color: red  
}  
p {  
    color: green  
}
```

# Poll 3 (Answer)

What will be the output of the following?

1. Red
2. Green
3. No colour

\*\* You can read more about specificity [here](#).

```
p {  
    color: red  
}  
  
p {  
    color: green  
}
```

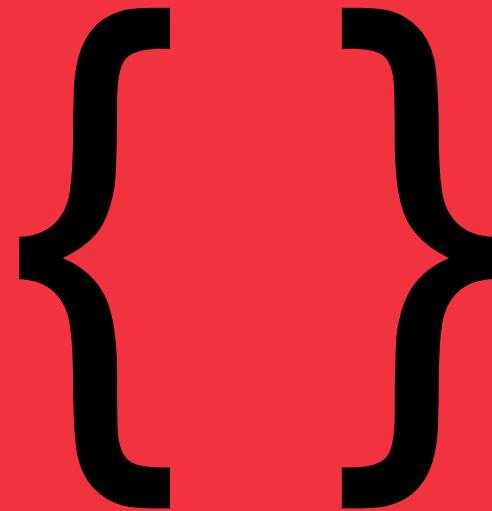
## Group CSS Selectors

```
<h1>
    This is a heading.
</h1>
<p id="para1">
    This is a paragraph 1.
</p>
<p id="para2">
    This is a paragraph 2.
</p>
<p id="para3">
    This is a paragraph 3.
</p>
```

```
h1, p { color: blue; } /*Group the Element selectors*/

#para1, #para2, #para3 { /* Group ID Selector */
    color: red;
}
```

# CSS: How to Use



## Inline vs Internal vs External

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Inline CSS</title>
  </head>
  <body>
    <span style="color: red">
      This is a span.
    </span>
  </body>
</html>
```

Inline CSS

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Internal CSS</title>
    <style>
      .text1 { color: red; }
    </style>
  </head>
  <body>
    <span class="text1">This is a
span.</span>
  </body>
</html>
```

Internal CSS

## Inline vs Internal vs External

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>External CSS</title>
    <link href="index.css"
rel="stylesheet" />
  </head>
  <body>
    <span class="text1">This is a
span.</span>
  </body>
</html>
```

index.html

```
.text1 {
  color: red;
}
```

index.css

## External CSS

## Inline vs Internal vs External

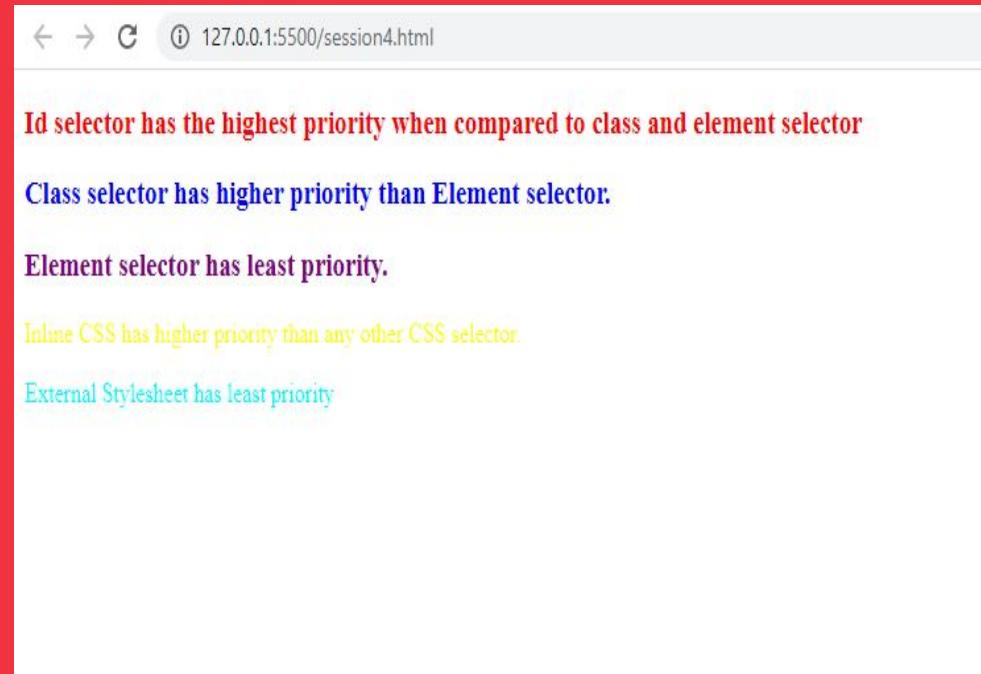
- **Inline stylesheet:** This is a CSS that is directly applied to an element inside HTML code.
- **Internal stylesheet:** This is a CSS that is applied inside a file but not inside an element. It is generally added inside the `<style>` tag, which is contained within the `<head>` tag.
- **External stylesheet:** This is a CSS that is applied from an external file. This external file is referred to inside the HTML code using the `<link>` tag.
- The following is the order of priority: Inline > Internal > External.

# Hands-On Exercise 1 (3 min)

Write a program for the given stub code such that the output is as shown in the adjoining screenshot.

The stub code is provided [here](#).

The solution is provided [here](#).



## Poll 4 (15 Sec)

Which of the following is the correct order of inline, external and internal CSS?

1. Inline > External > Internal
2. Inline > Internal > External
3. External > Internal > Inline
4. Internal > External > Inline

# Poll 4 (Answer)

Which of the following is the correct order of inline, external and internal CSS?

1. Inline > External > Internal
- 2. Inline > Internal > External**
3. External > Internal > Inline
4. Internal > External > Inline

# Poll 5 (15 Sec)

Where do we need to declare the reference to an external style sheet  
in our HTML document?

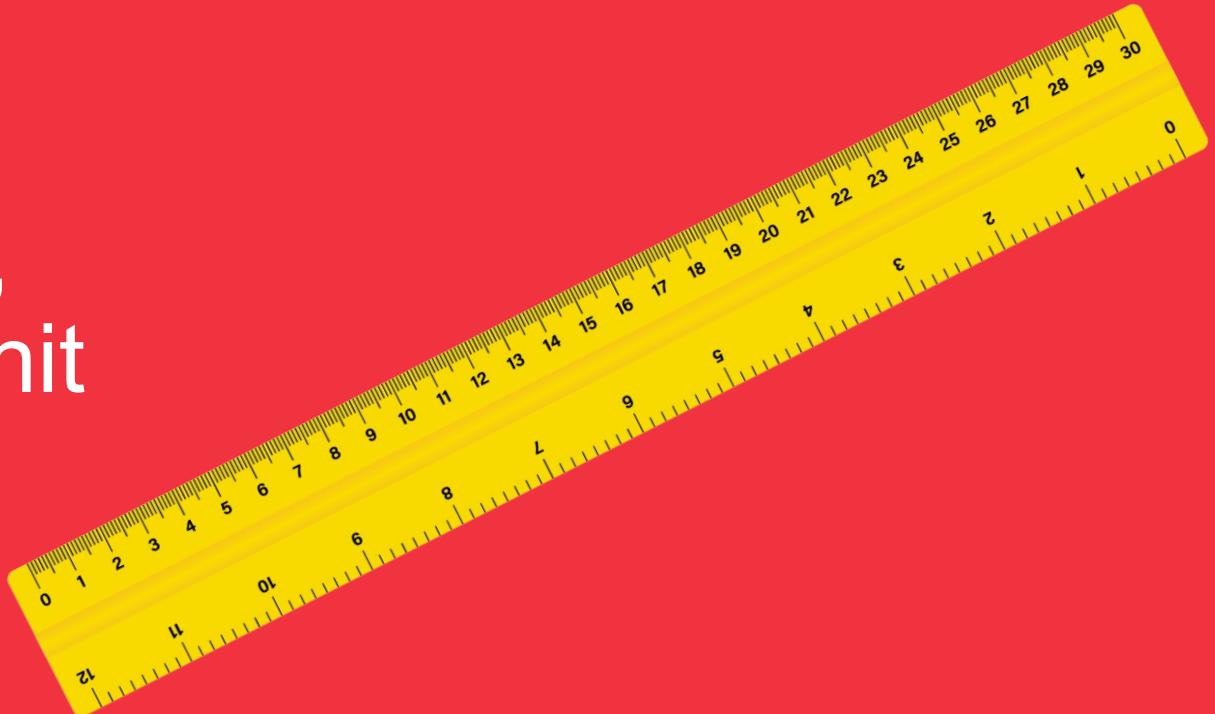
1. Before the <head> section
2. In the <head> section
3. In the <body> section
4. After the <body> section

# Poll 5 (Answer)

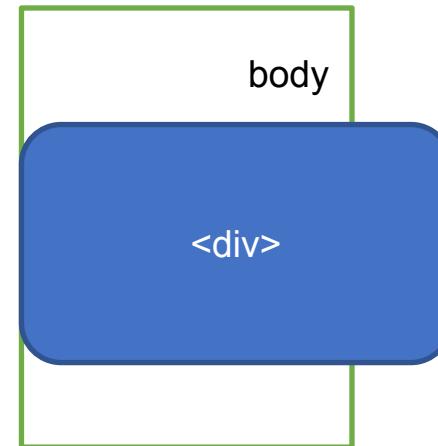
Where do we need to declare the reference to an external style sheet in our HTML document?

1. Before the <head> section
- 2. In the <head> section**
3. In the <body> section
4. After the <body> section

# CSS: Height, Width and Unit

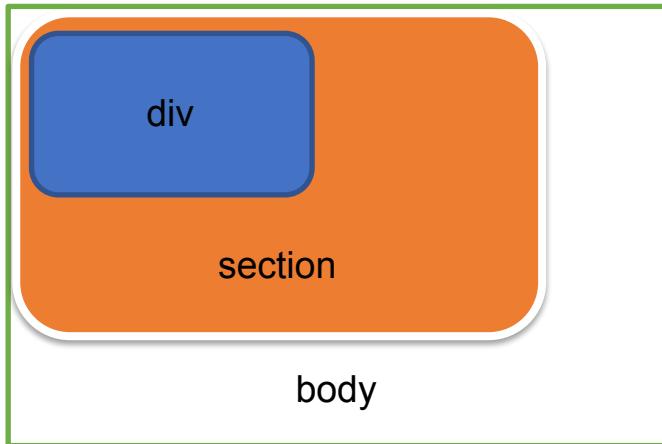


## CSS Units: Absolute vs Relative

**Absolute Units**

- Absolute units do not change size with respect to the container, i.e., their size are fixed.
- The examples of absolute units are px.

## CSS Units: px

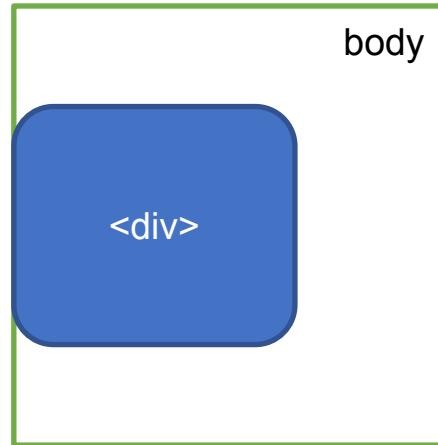
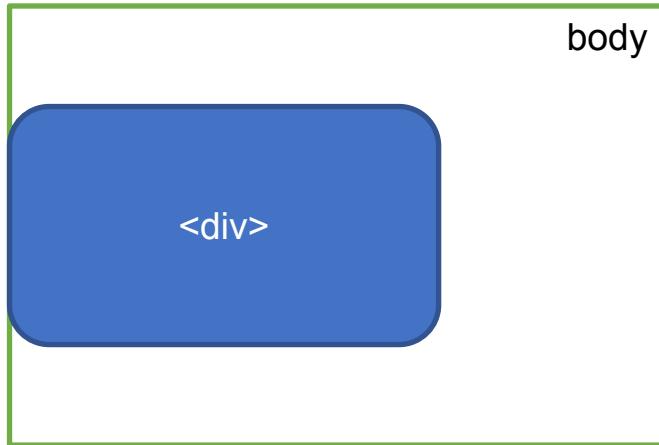


```
body {  
    font-size: 16px;  
}  
  
section {  
    font-size: 14px;  
}  
  
div {  
    font-size: 10px;  
}
```

***The unit px will maintain the exact size given to it.***

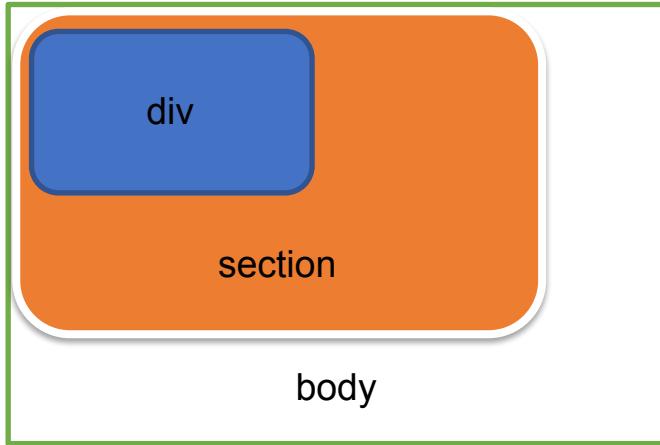
## CSS Units: Absolute vs Relative

### Relative Units



- Relative units change size along with their container, i.e., their size is not fixed.
- An example of a relative unit is em.

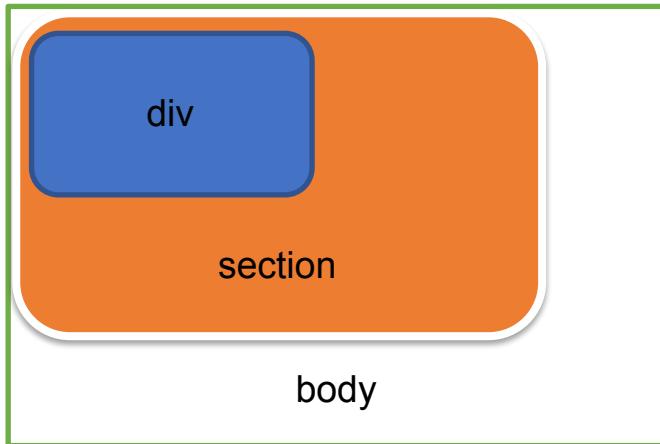
## CSS Units: em



```
body {  
    font-size: 16px;  
}  
  
section {  
    font-size: 14px;  
}  
  
div {  
    font-size: 1em; /* 1em = 14px */  
}
```

***The unit em will take the size of the parent element.  
Therefore,  
1em = 14px (size of parent 'section')  
2em = 28px, etc.***

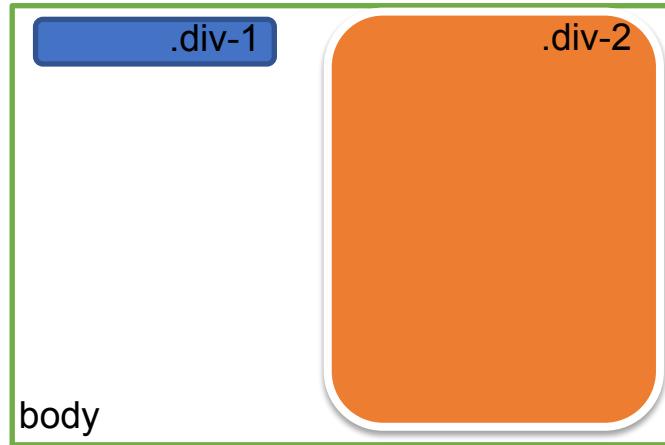
## CSS Units: rem



```
body {  
    font-size: 16px;  
}  
  
section {  
    font-size: 14px;  
}  
  
div {  
    font-size: 1rem; /* 1rem = 16px */  
}
```

**The unit rem will take the size of the root element, which is normally the body. Therefore, in the example given above, 1rem = 16px, 2em = 32px, etc.**

## CSS Units: vw and vh



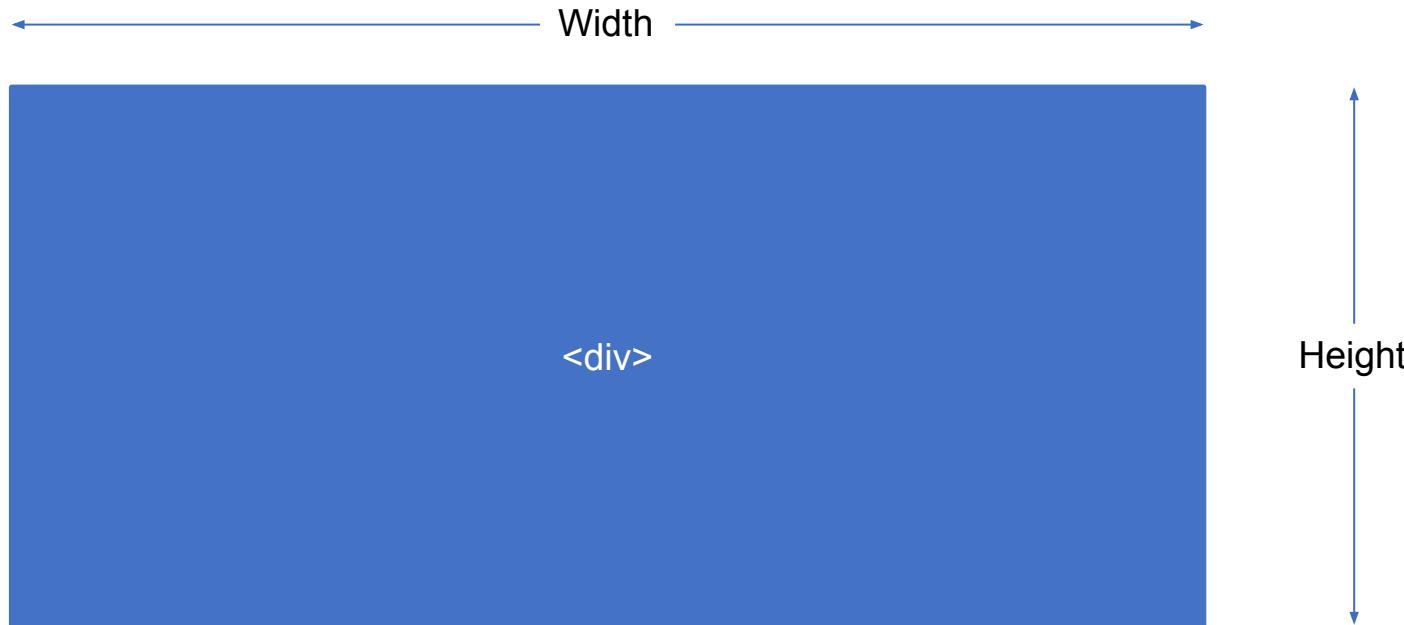
```
.div-1 {  
    height: 1vh;  
}  
  
.div-2 {  
    height: 100vh;  
}
```

- The unit **vw** stands for the viewport width, and **vh** stands for the viewport height.
- Therefore, 1vh stands for 1% of the viewport height (window height), and 100vh stands for 100% of the viewport height (window height). The same applies to vw.

## CSS Units: vmin, vmax and ch

- The unit **vmin** stands for the minimum of the viewport. It denotes the minimum between the viewport height and the viewport width.
- 1vmin is equal to 1vw or 1vh, whichever value is smaller.
- The unit **vmax** stands for the maximum of the viewport. It denotes the maximum between the viewport height and the viewport width.
- 1vmax is equal to 1vw or 1vh, whichever value is higher.
- 1ch sets the font size relative to the width of the character '0'.

## CSS Height and Width



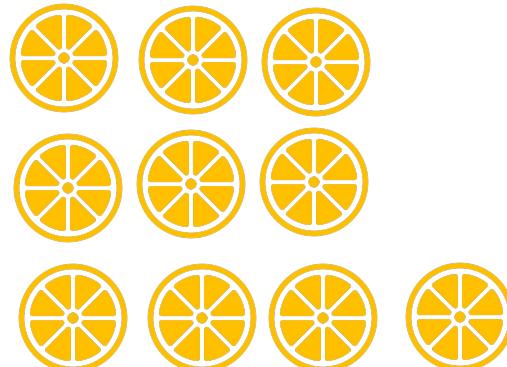
## CSS Heights and Widths: min



**Question:** A container has 10 oranges. If you are allowed to take a minimum of three oranges from this container, what is the least number of oranges that you can take?



**Question:** What is the maximum number of oranges that you can take?



**Question:** Can you also take more than three and less than 10 oranges?

**YES**

Since the minimum number of oranges you are allowed to take is three, you can also take four, five, six oranges or up to a maximum of 10 oranges.

## CSS Heights and Widths: min

```
div {  
    min-height: 200px;  
}
```

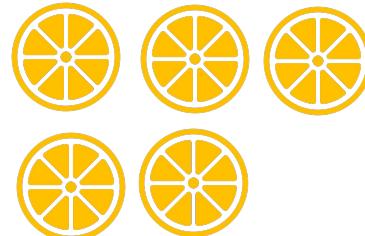
- When the **minimum height** is specified, it means that a **height that is lower than the specified value cannot be taken**, but there is no limit on the height taken above the specified value.
- In the example given above, the div can take any height higher than or equal to 200px (such as 300px, 450px and 1,000px), but it will not be allowed to take any height lower than 200px.
- The same applies to minimum width.

## CSS Heights and Widths: max



**Question:** A container has 10 oranges. If you are allowed to take a maximum of five oranges from this container, what is the least number of oranges you can take?

0



**Question:** What is the maximum number of oranges you can take?

**Question:** Can you also take more than 0 and less than five oranges?

YES

Since the maximum number of oranges you are allowed to take is five, you can either take no oranges or take only one, two or up to five oranges.

## CSS Heights and Widths: max

```
div {  
    max-height: 200px;  
}
```

- When the **maximum height** is specified, it means that a **height higher than the specified value cannot be taken**, but there is no limit on the height taken lower than the specified value.
- In the example given above, the div can take any height lower than or equal to 200px (such as 150px, 100px and 5px), but it will not be allowed to take height higher than 200px.
- The same applies for **maximum width**.

## CSS Heights and Widths: max and min

```
div {  
    min-height: 200px;  
    max-height: 1000px;  
}
```

- When both min and max are specified, they define a boundary between the two values.
- In the example given above, the div has a minimum height of 200px, and therefore, it cannot take any height value lower than 200px.
- Similarly, it has a maximum height of 1,000px, and therefore, it cannot take any value higher than 1,000px.
- Therefore, our **div** cannot have a height lower than 200px or higher than 1,000px.

# Project Work

(Let's start adding CSS to our project.)



You can refer to the solution [here](#).

# Hands-On Exercise 2 (3 min)

Write a program which has a main-container of 100% height and 100% width.

Inside the main container, you should have a video **Butterfly.mp4** given inside media folder. The video should play as soon as the page loads, should provide with the possibility to play/pause, should continue playing continuously, and should not have any sound. The video should have maximum absolute height 300 and maximum absolute width 700.

Below video you should include the audio **vibes.mp3** given inside media folder. The audio should not play automatically when the page loads and it should provide you with the possibility to play/pause the audio and it should play continuously after playing it until you manually pause it. The audio should have absolute width 540.

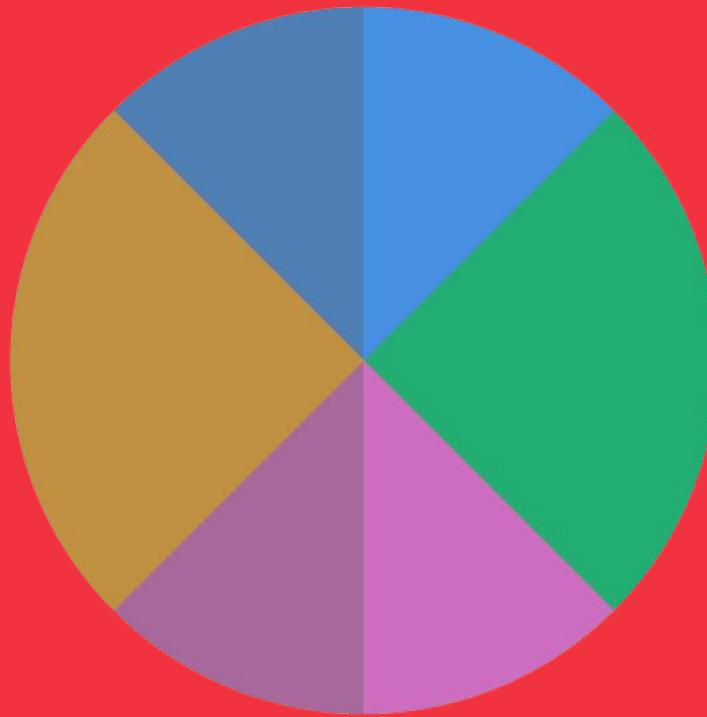
***Note: Watch the video for better understanding.***

The stub code is provided [here](#).

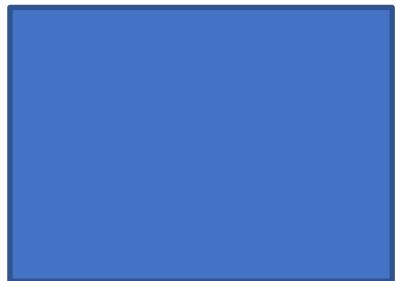
The solution is provided [here](#).



# CSS Colors



Where Can You Apply Colors?



background-color

A  
B  
C  
D  
E  
F  
G  
H  
I  
J  
K

text-color



border-color

## Different Ways of Typing Colors in CSS

red

```
div { color: red; }
```

rgb(255, 0, 0)

```
div { color: rgb(255, 0, 0); }
```

#ff0000

```
div { color: #ff0000; }
```

hsl(0, 100%, 50%)

```
div { color: hsl(0, 100%, 50%); }
```

rgba(255, 0, 0, 0.5)

```
div { color: rgba(255, 0, 0, 0.5); }
```

hsla(0, 100%, 50%, 0.5)

```
div { color: hsla(0, 100%, 50%, 0.5); }
```

## Different Ways of Typing Colors in CSS: Color Name

The color value is as follows:

red

The following syntax is used:

```
.container {  
    color: red;  
}
```

- To add a color using the color name, you simply need to add the color name to an HTML element.
- The values can be colors such as *orange*, *blue*, *green* and *yellow*.

You can read more about CSS colors [here](#).

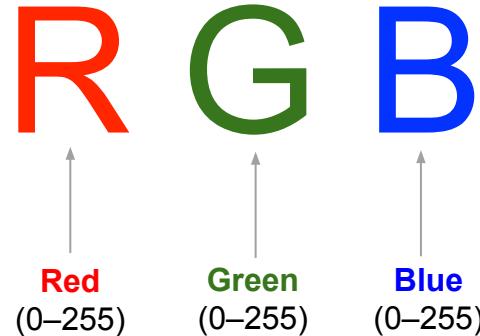
## Different Ways of Typing Colors in CSS: RGB

The color value is as follows:

```
rgb(255, 0, 0)
```

The following syntax is used:

```
div {  
  color: rgb(255, 0, 0);  
}
```



- To add a color using RGB, you simply need to add a value between 0 and 255 for the respective color.
- For example, for a pure red color, it will be `rgb(255, 0, 0)`, where 255 denotes the purest shade of red. For a lighter shade of red, you can reduce the value to less than 255, for example, `rgb(100, 0, 0)`.
- If you want a color such as yellow, then you can simply add its combination such as `rgb(255,255,0)`.

## Different Ways of Typing Colors in CSS: HEX

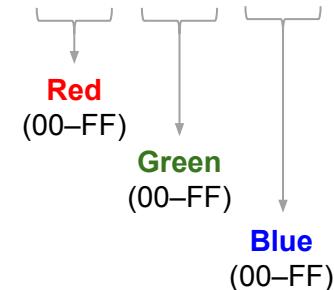
The color value is as follows:

#ff0000

The following syntax is used:

```
div {  
  color: #ff0000;  
}
```

#FFFFFF



- To add a color using HEX, you simply need to add a value between 00 and FF for the respective color.
- For example, for a pure red color, it will be #ff0000, wherein ff denotes the purest shade of red. For a lighter shade of red, you can reduce the value to less than ff, for example, #2a0000.
- If you want a color such as yellow, then you can simply add its combination #ffff00.

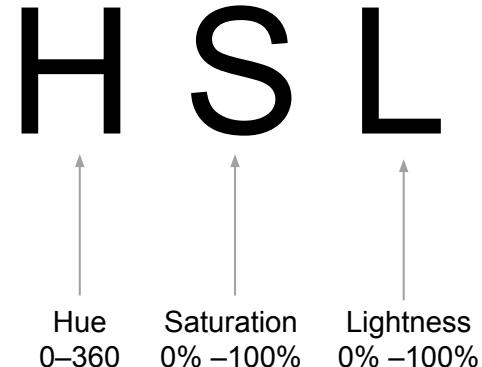
## Different Ways of Typing Colors in CSS: HSL

The color name is as follows:

```
hsl(0, 100%, 50%)
```

The following syntax is used:

```
div {  
  color: hsl(0, 100%, 50%);  
}
```



- **Hue** stands for **color**. Its value is a degree on the color wheel. 0 is **red**, 120 is **green** and 240 is **blue**.
- **Saturation** stands for the intensity of color, where 100% is pure color, 50% is half grey and half color, and 0% is pure grey.
- **Lightness** stands for the amount of light that you want to give the color. 0% is no light (black), 50% means neither light nor dark, and 100% means full light (white).

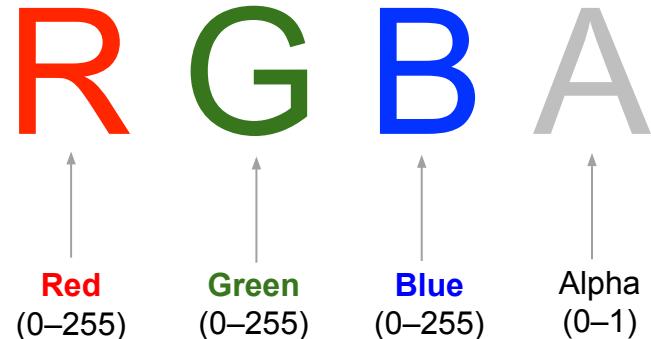
## Different Ways of Typing Colors in CSS: RGBA

The color name is as follows:

```
rgba(255, 0, 0, 0.5)
```

The following syntax is used:

```
div {  
  color: rgba(255, 0, 0, 0.5);  
}
```



- In addition to RGB, you can add a touch of transparency-opacity using the alpha value.
- It takes a value from 0 to 1, where 0 stands for fully transparent and 1 stands for fully opaque.

## Different Ways of Typing Colors in CSS: HSLA

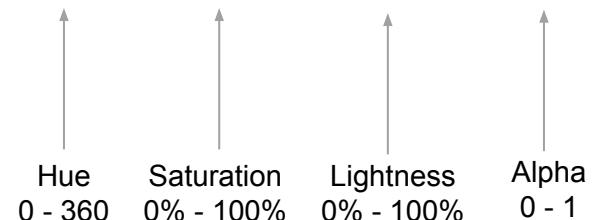
The color name is as follows:

```
hsla(0, 100%, 50%, 0.5)
```

# H S L A

The following syntax is used:

```
div {  
  color: hsla(0, 100%, 50%, 0.5);  
}
```



- In addition to HSL, you can add transparency to the color.
- It takes a value from 0 to 1, where 0 stands for fully transparent and 1 stands for fully opaque.

## Poll 6 (15 Sec)

Which of the following is true about the RGB stream?  
(Note: More than one option may be correct.)

1. RGB stands for red, green and blue.
2. It is represented within the range (0, 0, 0) and (255, 255, 255).
3. It is represented by #RRGGBB.
4. RGB stands for red, grey and black.

# Poll 6 (Answer)

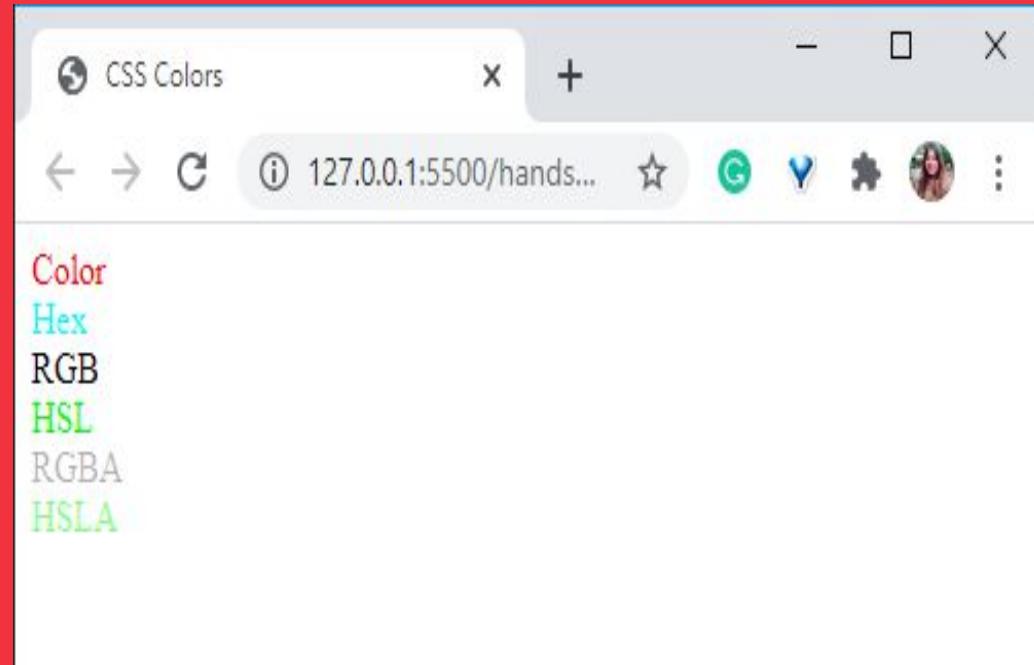
Which of the following is true about the RGB stream?  
(Note: More than one option may be correct.)

1. **RGB stands for red, green and blue.**
2. **It is represented within the range (0, 0, 0) and (255, 255, 255).**
3. It is represented by #RRGGBB.
4. RGB stands for red, grey and black.

# Hands-On Exercise 3 (3 min)

Your are given a HTML stub code to which you are required to write any external CSS such that following points are satisfied:

- The color of **container1** should be declared with color value “red”
- The color of **container2** should be declared with hex code “00ffff”
- The color of **container3** should be declared using RGB code for black.
- The color of **container4** should be declared using HSL code for green such that the saturation is complete and the lightness is half.
- The color of **container5** should be declared using RGBA code for black having opacity of 0.3.
- The color of **container6** should be declared using HSLA code for green such that the saturation is complete, the lightness is half and opacity is 0.5.



The stub code is provided [here](#).

The solution code is provided [here](#).

# CSS Text

## Chapter One

  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer nec lacinia risus. Integer a lorem malesuada, suscipit sapien sed, molestie velit. Sed augue urna, dictum id blandit vitae, posuere vitae justo. Fusce vitac urna ac lectus volutpat viverra sit amet cursus sem. Praesent sit amet nunc eros. Proin vulputate ante in fringilla pulvinar. In fringilla lorem ut suscipit blandit. Praesent in erat eu sem ullamcorper tincidunt ac nec dolor. Donec accumsan dignissim pharetra. Mauris ultrices, elit vel facilisis bibendum, risus tellus dictum nunc, at varius lacus elit eget nibh. Suspendisse sodales scelerisque magna, vita lobortis ex eleifend viverra. Fusce lacinia ornare libero. Sed ligula felis, aliquam sit amet sollicitudin non, venenatis et magna.

  Vivamus vitae accumsan libero. Maecenas sit amet interdum odio, quis imperdiet magna. Maecenas vehicula mi sagittis, malesuada nunc eget, faucibus felis. Sed in libero maximus, gravida erat quis, ullamcorper ipsum. Etiam varius tempor orci, a scelerisque tortor tincidunt eget. Curabitur auctor ipsum ex, vel mollis erat consequat non. Donec fermentum tristique ipsum nec fermentum. Morbi et leo id urna suscipit feugiat vel vitae nisi. Donec ullamcorper maximus sem et porttitor. Phasellus ac ultrices est. Nulla pulvinar enim felis, vel vulputate velit faucibus sed. Aliquam eget rhoncus felis, et dapibus nulla.

  Duis quis rhoncus mauris. Integer nisl nisl, vehicula sed metu vitae, sagittis malesuada ante. Vestibulum sit amet nisl efficitur, aliquam arcu a, efficitur nibh. Morbi vulputate elit id ligula varius condimentum. Aliquam vitae tortor mauris. Donec eu lorem at tortor tempus pulvinar. Etiam iaculis enim sed ligula vulputate, et condimentum est auctor. Interdum et malesuada fames ac ante ipsum primis in faucibus. Proin imperdiet, ante nec finibus sollici

  Mauris vestibulum scelerisque faucibus. Aliquam rhoncus eu mauris ut vestibulum. Pellentesque maximus mi ante, vel rutrum mi molestie in. Donec non justo a orci accumsan aliquam ut at libero. Duis eu nunc orci. Morbi vel enim et arcu facilisis cursus et vitae ante. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. In faucibus felis eget lorem sollicitudin, at faucibus erat porta. Vestibulum lorem lorem, posuere sagittis aliquet quis, facilisis et nunc. Donec et ex elementum, vestibulum mauris at, fermentum mi. Nam sagittis nibh felis, quis pulvinar elit laoreet et. Proin malesuada neque diam, sit amet hendrerit massa pulvinar a. Etiam dignissim, lorem ac scelerisque cursus, tellus nulla vestibulum elit, in efficitur lorem et odio. Duis ac metus ac odio molestie laoreet non sed velit. Proin a enim et turpis finibus vehicula. Praesent elementum leo non ante malesuada, eu lobortis erat scelerisque.

  Integer elementum condimentum feugiat. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Etiam feugiat, nisl a vulputate faucibus, massa lorem ornare erat, vel efficitur quam nunc in nunc. Suspendisse blandit urna eget laoreet hendrerit. Morbi in porta lorem, in pulvinar ipsum. Sed accumsan semper augue, ut facilisis est ornare id. Cras molestie neque molestie arcu interdum, quis efficitur enim ultrices. Praesent nec enim nisi. Phasellus ut urna nec dui luctus tristique. Etiam vestibulum lacinia tellus sagittis iaculis. Sed facilisis consequat neque, eu viverra neque facinata et. Vestibulum sit amet faucibus ipsum. Etiam quam augue, luctus in arcu vitae, eleifend condimentum leo. Vestibulum porttitor diam eu ante tristique, vitae rutrum dolor pulvinar. Curabitur molestie condimentum lorem, at feugiat nisi elementum quis. Quisque et nisi vitae massa bibendum auctor at sed lectus.

  Pellentesque a augue et dolor sagittis mattis. Maecenas nunc turpis, elementum eget lobortis sed, vehicula at dolor. Fusce tempus odio eget feugiat rhoncus. Pellentesque sed dignissim tellus, cursus viverra tortor.

- 25 -

- 26 -

## Text colors

- To give a color to the text of an element, you simply need to use the property **color** and assign it a value.
- The values can be normal colors, like ‘orange’, ‘blue’, ‘green’, ‘yellow’, etc., or RGB or HEX or HSL, or their alpha counterparts.
- Additionally, you can give a color to a body, div or basically any element, irrespective of whether it is a text element, such as p, span or h1-h6, or a container.

This is a paragraph.

```
p { color: green; }
```

## Text Shadow

- The '[text-shadow](#)' property adds a shadow to your text.
- This property constituents three different variants.
  - X and Y offset
  - blur radius
  - color

```
text-shadow: 2px 2px; /* horizontal-shadow vertical-shadow */
```

This has h-shadow and v-shadow

This has h-shadow, v-shadow and color

This has h-shadow, v-shadow, blur and color

```
text-shadow: 2px 2px red; /* horizontal-shadow vertical-shadow color */
```

```
text-shadow: 2px 2px 5px red; /* horizontal-shadow vertical-shadow blur color */
```

## Text Alignment – Horizontal Alignment

- A text can be aligned either vertically or horizontally.
- To align a text horizontally, use [text-align](#).
- For text-align, you can use ***left***(default) ***center or right***.

```
#text1  
      #text2  
  
      #text3
```

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>CSS Text</title>  
  </head>  
  <body>  
    <p id="text1">#text1</p>  
    <p id="text2">#text2</p>  
    <p id="text3">#text3</p>  
  </body>  
</html>
```

```
#text1 { text-align: left; }  
#text2 { text-align: center; }  
#text3 { text-align: right; }
```

## Text Alignment – Vertical Alignment

- A text can be aligned either vertically or horizontally.
- To align *an element (not simply a text)* vertically, use **vertical-align**. For vertical-align, the elements need to be set along a baseline, like `<span>` or `display: inline`.
- For vertical-align, you can use ***top, middle, bottom, sub, super, baseline***(default value), ***text-top or text-bottom***.

### **baseline**

The image  has vertical align as baseline

### **top**

The image  has vertical align as top

### **bottom**

The image  has vertical align as bottom

### **middle**

The image  has vertical align as middle

```
#baseline { vertical-align: baseline; }
#top { vertical-align: top; }
#bottom { vertical-align: bottom; }
#middle { vertical-align: middle; }
img {height: 10px; width: 20px }
```

`<h3>baseline</h3>`

`<p>The image `  
has vertical align as baseline`</p>`

`<h3>top</h3>`

`<p>The image ` has  
vertical align as top`</p>`

`<h3>bottom</h3>`

`<p>The image `  
has vertical align as bottom`</p>`

`<h3>middle</h3>`

`<p>The image `  
has vertical align as middle`</p>`

## Text Alignment – Direction and Justification

- The property direction, along with unicode-bidi, determines the direction of flow of the text.
- When the text-align property is set to *justify*, every line is stretched equally, so that each of them is of equal width. You can normally find justified text used in magazines and newspapers to maintain consistency.

This is normal direction.

.noitcerid txet tfel ot thgir si sihT

```
#text1 { direction: ltr; }

#text2 { text-align: rtl; unicode-bidi: bidi-override; }
```

In this text, we try to show the 'text-align: justify' property. This first example is not 'justified' aligned and so, you can see each line ends at different lengths.

Not aligned using justify

In this text, we try to show the 'text-align: justify' property. This second example is justified aligned and so, you can see that each line ends at equal length.

text-align: justify

# Poll 7 (15 Sec)

The '***text-align:***' CSS property is used to align a text:

1. Vertically
2. Horizontally
3. Both

# Poll 7 (Answer)

The '***text-align:***' CSS property is used to align a text:

1. Vertically
2. **Horizontally**
3. Both

## Text Decoration: text-decoration-line

- **text-decoration-line** is used to style the element with a line.
- It can take values such as none, underline, overline and line-through.

None

```
text-decoration:  
none;
```

Underline

```
text-decoration:  
underline;
```

~~Line-through~~

```
text-decoration:  
line-through;
```

Overline

```
text-decoration:  
overline;
```

Text Decoration: text-decoration-color

- **text-decoration-color** is used to add the color decoration to **text-decoration-line**.

This text is decorated using ~~text-line~~ text-decoration-line and text-decoration-color

```
#heading{text-decoration-line: none; text-decoration-color: purple;}  
#mistake {text-decoration-line: line-through;text-decoration-color: #ff0000;}  
#correct {text-decoration-line: underline; text-decoration-color: #00ffff;}  
#info { text-decoration-line: overline; text-decoration-color: #0000ff; }
```

```
<h1 id="heading">This text is decorated using  
  <span id="mistake">text-line</span>  
  <span id="correct">text-decoration-line</span> and  
  <span id="info">text-decoration-color</span>  
</h1>
```

## Text Decoration: text-decoration-style

- **text-decoration-style** is used to add style to **text-decoration-line**.
- text-decoration-style can have values: ***solid, dotted, dashed, wavy and double.***

This text is decorated using ~~text-line~~ text-decoration-line and text-decoration-color

```
<h1 id="heading">This text is decorated using  
    <span id="mistake">text-line</span>  
    <span id="correct">text-decoration-line</span> and  
    <span id="info">text-decoration-color</span>  
</h1>
```

```
#heading{text-decoration-line: none;text-decoration-color: purple;text-decoration-style: solid;}  
#mistake{text-decoration-line: line-through;text-decoration-color: #ff0000;text-decoration-style: wavy;}  
#correct {text-decoration-line: underline;text-decoration-color: #00ff00;text-decoration-style: dotted;}  
#info {text-decoration-line: overline;text-decoration-color: #0000ff;text-decoration-style: double;}  
5
```

Text Decoration: text-decoration

- **text-decoration** is a shorthand property which is used to set the ***text-decoration-line*, *text-decoration-color* and *text-decoration-style***.

This text is decorated using ~~text-line~~ text-decoration-line and text-decoration-color

```
<h1 id="heading">This text is decorated using  
  <span id="mistake">text-line</span>  
  <span id="correct">text-decoration-line</span> and  
  <span id="info">text-decoration-color</span>  
</h1>
```

```
#heading { text-decoration: none purple solid; }  
#mistake { text-decoration: line-through #ff0000 wavy; }  
#correct { text-decoration: underline #00ffff dotted; }  
#info { text-decoration: overline #0000ff double; }
```

## Text Transform

The '[text-transform](#)' property is used to set alphabets in uppercase or lowercase in a text.

Normal

```
text-transform:  
none;
```

UPPERCASE

```
text-transform:  
uppercase;
```

lowercase

```
text-transform:  
lowercase;
```

Capitalised Text

```
text-transform:  
capitalize;
```

## Text Spacing

The '[text-indent](#)' property is used to add a space to the left of the first line of a text.

This is a paragraph. It's typed with <p>. The text-indent property indents the first line.

```
text-indent: 50px;
```

## Text Spacing

The '[letter-spacing](#)' property is used to define the space between the **characters** in a text.

This has letter spacing 2px

```
#text1 { letter-spacing: 2px; }  
#text2 { letter-spacing: -2px; }
```

This has letter spacing -3px

```
<h1 id="heading1">This has letter spacing 2px</h1>  
<h1 id="heading2">This has letter spacing -3px</h1>
```

'[letter-spacing](#)' property can have value as **normal**(default) or any **value** with proper CSS unit.

## Text Spacing

The '[line-height](#)' property is used to define the space between the **lines** in a text.

This is normal line-height.  
This is normal line-height.

```
line-height: 1.2;  
/*default value usually is  
1.2. However, it depends on  
the font family.*/
```

This is bigger line-height.  
This is bigger line-height.

```
line-height: 2;
```

This is normal line-height:

```
line-height: 0.6;
```

## Text Spacing

The '[word-spacing](#)' property is used to add spaces between the **words** of a text.

This has word spacing 10px

```
#heading1 { word-spacing: 10px; }  
#heading2 { word-spacing: -10px; }
```

This has word spacing -10px

```
<h1 id="heading1">This has word spacing 10px</h1>  
<h1 id="heading2">This has word spacing -10px</h1>
```

## Text Spacing

The '[white-space](#)' property specifies how to handle white-spaces in an element.

This is some text. This is some text. This is some text. This is some text. This is some text.

```
white-space: nowrap;
```

This is some text. This is some text. This is some text. This is some text. This is some text.

```
white-space: normal;
```

# CSS Fonts



## Font Family

- We can use CSS **font-family** property for a HTML text such that it belongs to a certain font family.
- The syntax of font-family is:  
***font-family: <family-name><generic-name>***
- There are various font families that can be used to style the HTML text.
  - serif
  - sans-serif
  - cursive
  - monospace
  - fantasy
  - emoji
  - math

# Google Fonts

*Click [here](#) to go to Google Fonts.*



Google CGOQ  
Oooglo c g o q

## Google Font

```
<link href='https://fonts.googleapis.com/css?family=Sofia' rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Modak' rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Fondamento' rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Shadows+Into+Light' rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Yellowtail' rel='stylesheet'>

<style>
    body { font-size: 20px; }
    #para1 {font-family: 'Sofia';}
    #para2 {font-family: 'Modak';}
    #para3 {font-family: 'Fondamento';}
    #para4 {font-family: 'Shadows Into Light';}
    #para5 {font-family: 'Yellowtail';}
</style>
```

```
<p id="para1">Sofia: Lorem ipsum dolor sit amet consectetur adipisicing elit. </p>
<p id="para2">Modak: Lorem ipsum dolor sit amet consectetur adipisicing elit. </p>
<p id="para3">Fondamento: Lorem ipsum dolor sit amet consectetur adipisicing elit. </p>
<p id="para4">Shadows Into Light: Lorem ipsum dolor sit amet consectetur adipisicing elit. </p>
<p id="para5">Yellowtail: Lorem ipsum dolor sit amet consectetur adipisicing elit. </p>
```

## Google Font

Sofia: *Lorem ipsum dolor sit amet consectetur adipisicing elit.*

**Modak:**  *****Lorem ipsum dolor sit amet consectetur adipisicing elit.*****

Fondamento: *Lorem ipsum dolor sit amet consectetur adipisicing elit.*

Shadows Into Light: *Lorem ipsum dolor sit amet consectetur adipisicing elit.*

*Yellowtail:* *Lorem ipsum dolor sit amet consectetur adipisicing elit.*

## Font

[font-size](#) is used to declare the size of the font.

  Lorem ipsum dolor

**Lorem ipsum dolor**

  Lorem ipsum dolor

  Lorem ipsum dolor

  Lorem ipsum dolor

```
#para1 {font-size: 20px;}  
#para2 {font-size: 2em;}  
#para3 {font-size: small;}  
#para4 {font-size: medium;}  
#para5 {font-size: large;}
```

```
<p id="para1">Lorem ipsum dolor</p>  
<p id="para2">Lorem ipsum dolor</p>  
<p id="para3">Lorem ipsum dolor</p>  
<p id="para4">Lorem ipsum dolor</p>  
<p id="para5">Lorem ipsum dolor</p>
```

## Font

**'font-weight'** is used to specify the weight of a font. It could be ***normal; bold; bolder; lighter;*** or a ***value like 100, 400, 700, etc.*** The value 400 is the same as normal and the value 700 is the same as bold.

This is a text.

```
font-weight: 700;
```

**'font-style'** is used to make the text ***normal, italic or oblique.***

This is a text.

```
font-style: italic;
```

## Font

- font property is a shorthand property.
- The syntax for the shorthand property is **font-style font-variant font-weight font-size/line-height font-family**.

For example:

```
font: italic bold 12px/2 Georgia/serif;
```

*This property is styled as italic and bold with small caps, with 12px as font size and a line-height of 30px, with Georgia font and a fallback font as serif.*

# Poll 8 (15 Sec)

How can you make the <p> element bold?

1. `<p font-type:“bold”>`
2. `<p style=“font-type:bold”>`
3. `<p style=“font-weight:bold”>`
4. `<p style=“text-weight:bold”>`

# Poll 8 (Answer)

How can you make the <p> element bold?

1. <p font-type:“bold”>
2. <p style=“font-type:bold”>
3. **<p style=“font-weight:bold”>**
4. <p style=“text-weight:bold”>

# CSS Icons



- Often, you will need to add icons to different pages on your website.
- There are different ways of adding icons to your website.

Google Icons

<https://fonts.googleapis.com/?query=icons>

Font Awesome Icons

<https://fontawesome.com/>

Bootstrap Icons

<https://icons.getbootstrap.com/>

Material Icons

<https://material.io/resources/icons/?style=baseline>

# Hands-On Exercise 4 (3 mins)

Use the Font Awesome library to create the stars as shown in the screenshot.

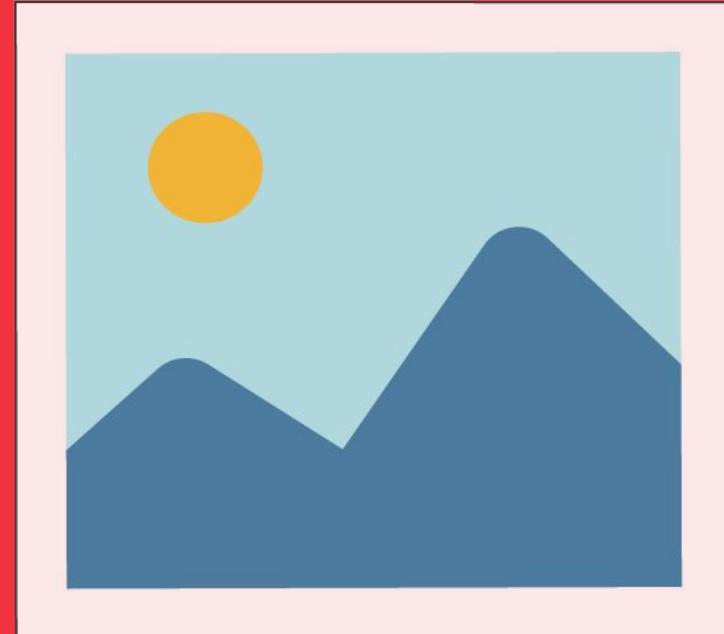
Please follow the TODOs in the given Stub Code.

Stub Code: [here](#)

Solution: [here](#)

The screenshot shows a web browser window with the URL `127.0.0.1:5500/TechM_Webtech_InSessionStandaloneCodes/tm-webtech-examples/Session%208/Hands-On%20Font%20Awesome/solution.html`. The page title is "Star Rating Using Font Awesome". Below the title, there is a red rectangular button containing five yellow star icons, each with a black outline. The browser interface includes standard navigation buttons (back, forward, search), a refresh icon, and various status icons in the top right corner.

# CSS Backgrounds



## Background Color and Opacity

### **background-color**



The syntax used is as follows:

```
div {  
    background-color: green;  
}
```

- To give a background color to an element, you can simply use the **background-color** syntax.

### **opacity**



The syntax used is as follows:

```
div {  
    background-color: green;  
    opacity: 0.5;  
}
```

- Giving opacity to an element will give the same opacity to every child element inside the *div*.

## Background Image

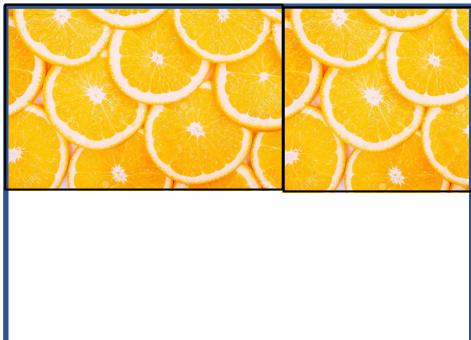


The syntax used is as follows:

```
body {  
    background-image: url('images/oranges.jpg');  
}
```

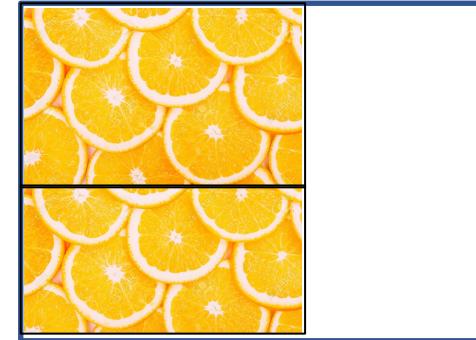
## Background Repeat

If the size of an image is smaller than that of the container for adding a background, then the image will repeat itself. The **background-repeat** property will define if the background image should be repeated or not.



```
body {  
    background-image:  
    url('images/oranges.jpg');  
    background-repeat:  
    repeat-x;  
}
```

repeat-x



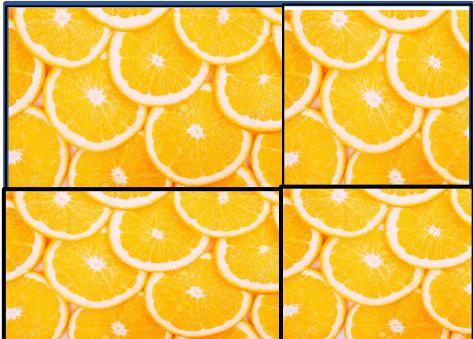
```
body {  
    background-image:  
    url('images/oranges.jpg');  
    background-repeat:  
    repeat-y;  
}
```

repeat-y

## Background Repeat

The **background-repeat** property will define if the background image should be repeated or not.

*The default value of background-repeat property is repeat.*



```
body {  
    background-image:  
    url('images/oranges.jpg');  
    background-repeat:  
    repeat;  
}
```

repeat



```
body {  
    background-image:  
    url('images/oranges.jpg');  
    background-repeat:  
    no-repeat;  
}
```

no-repeat

## Background Position

- The **background-position** property will define where the background image will be placed.
- It can be either **initial**, **inherit**, '**right/left/centre top/centre/bottom**', '**x% y%**' or '**xpos ypos**'.



```
body {  
    background-image:  
    url('images/oranges.jpg');  
    background-repeat: no-repeat;  
    background-position: right top;  
}
```

## Background Size

The **background-size** property is used to define the size of the background image. While you can set the height and width of the background image using simple height and width properties, you can also set it in the following three unique ways.



`background-size: auto`

This will give the original size of the image. This is the default value of `background-size` if not defined explicitly.



`background-size: cover`

This will resize the image to fit the container, even if it needs to stretch the image or cut off its edges.



`background-size: contain`

This will resize the image, ensuring that the image is fully visible.

## Background Attachment

The **background-attachment** property specifies whether the background will scroll with the rest of the page or remain fixed.



`background-attachment: fixed`



`background-attachment: scroll`

## Background Attachment

The **background-attachment** property specifies whether the background will scroll with the rest of the page or remain fixed.

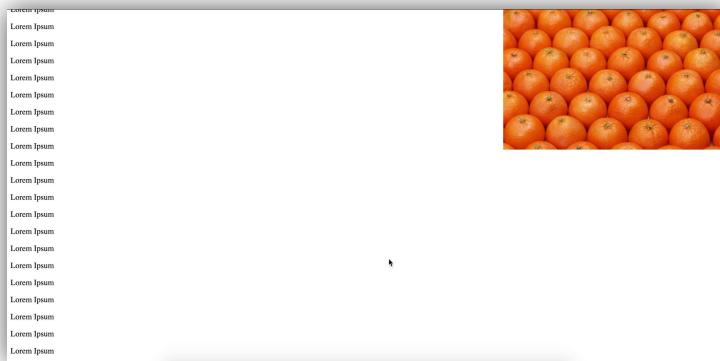
- The default value of **background-attachment** property is scroll.
- When we set the background-attachment as scroll, the background will scroll when the page is scrolled.



`background-attachment: scroll`

## Background Attachment

The **background-attachment** property specifies whether the background will scroll with the rest of the page or remain fixed.



- When we set the `background-attachment` as `fixed`, the background will not scroll when the page is scrolled instead it stays fixed at that position.

`background-attachment: fixed`

## Background Shorthand

- The **background** shorthand property specifies all the background properties in one line.
- The syntax used is as follows: **background: background-color,background-image,background-position,background-size,background-repeat,background-origin,background-clip,background-attachment.**
- If we omit any of the sub property value in the shorthand property, the value of that sub property is set to [initial value](#),

```
body {  
  background:#ffffff url('images/oranges.jpg') 0% 0% cover  
  no-repeat border-box padding-box fixed;  
}
```

Property	Initial Value
background-clip	border-box
background-color:	transparent
background-image:	none
background-origin:	padding-box
background-position:	0% 0%
background-repeat:	repeat
background-size:	auto
background-attachment:	scroll

# Project Work

(Let's add some more CSS to our project.)



You can refer to the solution [here](#).

# Poll 9 (15 Sec)

Which of the following statements is correct about background-attachment property?

1. It specifies how the background should be positioned.
2. It specifies how the background should be repeated.
3. It specifies whether the background should scroll with the page or remain fixed.
4. It specifies the size of the background image.

# Poll 9 (Answer)

Which of the following statements is correct about background-attachment property?

1. It specifies how the background should be positioned.
2. It specifies how the background should be repeated.
3. **It specifies whether the background should scroll with the page or remain fixed.**
4. It specifies the size of the background image.

# Hands-On Exercise 5 (3 min)

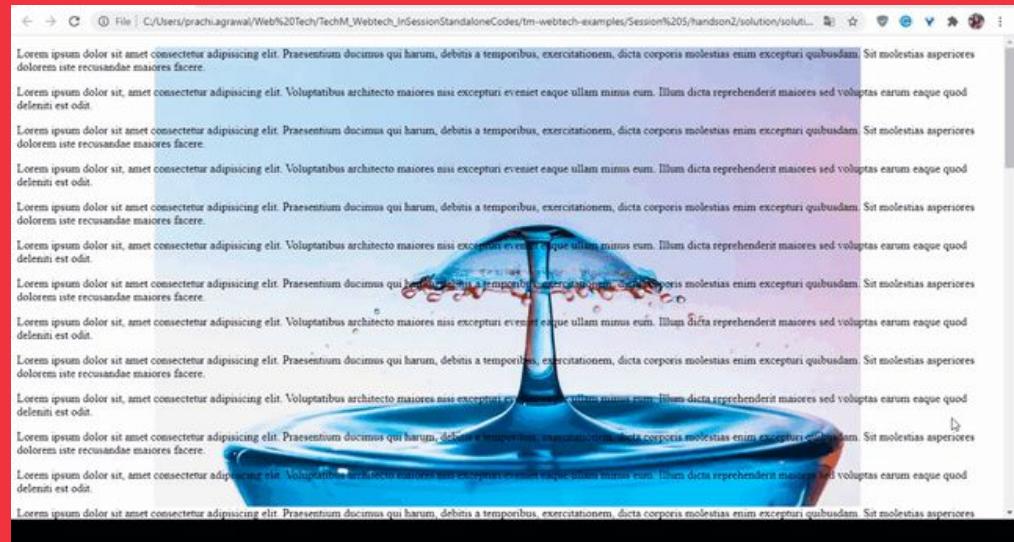
The HTML code is given to you in the stub and you are required to add the following internal CSS to the class **container**.

- Add the background image for the container:  
[https://cdn.pixabay.com/photo/2020/06/07/05/48/drop-5269146\\_960\\_720.jpg](https://cdn.pixabay.com/photo/2020/06/07/05/48/drop-5269146_960_720.jpg)
- The position of the image should be 50% on the x-axis and 0% on the y-axis.
- The image should not repeat.
- As we scroll across the web page, the image should remain static at the given position and only the remaining elements on the web page should scroll.

You are required to define all this properties on a single line inside CSS.

The stub code is provided [here](#).

The solution code is provided [here](#).



# Key Takeaways:

- CSS adds styling to HTML and makes HTML elements presentable.
- You can access elements in CSS using the tag name or the class ('.'), id ('#') or universal ('\*') selectors.
- CSS styles can be added in-line, internally in the <style> tag and via an external style sheet. These are arranged in the order of priority as follows: *inline* > *internal* > *external*.
- For an element, min-\* is the minimum value, and max-\* is the maximum value.
- Absolute units such as px do not change size with respect to the containers, and relative units such as em or rem change size with respect to the containers.
- The six ways of defining colors are as follows: color-name; HEX; RGB; HSL; RGBA; and HSLA, where A stands for opacity.
- You can set a background color or a background image. Background images can be made to fit an entire container or can be set to be repeated. Additionally, they can be set to scroll with the container or stay fixed.

The following tasks are to be completed after today's session:

MCQs
Coding Questions
Project: Checkpoint 4

In the next class, we will discuss...

- CSS Box Model



Thank you!