

## Pseudo-Elements

- A CSS pseudo-element is a keyword added to a selector that lets you style a specific part of the selected element.
- Pseudo-element in CSS is used to add style in specified parts of an element.
- Pseudo-element names are not case-sensitive.
- Pseudo-elements start with a double colon (::).

### Syntax

```
selector::pseudo-element {  
  property:value;  
}
```

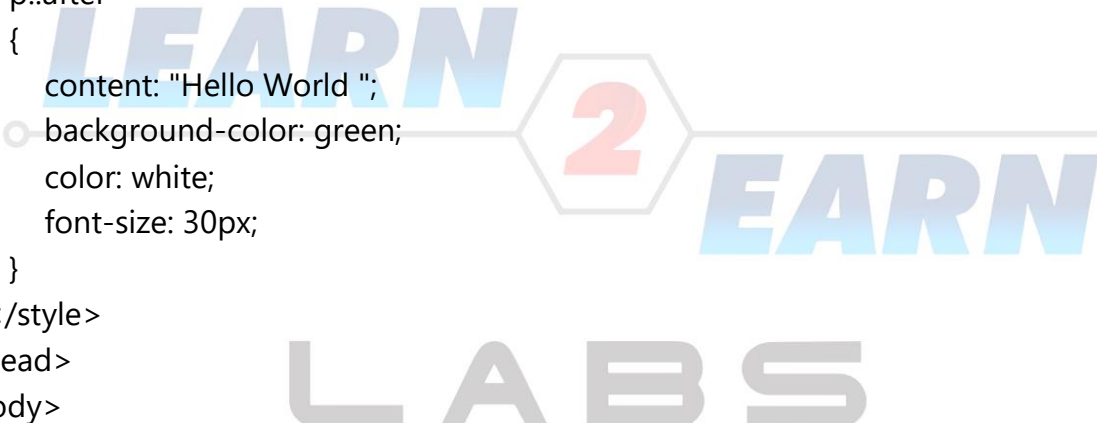


## 1) ::after / :after

- In CSS, ::after creates a pseudo-element that is the last child of the selected element.
- It is often used to add cosmetic content to an element with the content property.
- It is inline by default.

### Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Document</title>
  <style>
    p::after
    {
      content: "Hello World ";
      background-color: green;
      color: white;
      font-size: 30px;
    }
  </style>
</head>
<body>
  <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Voluptatibus at
  dignissimos cum, rem obcaecati minus officia similique animi ratione ex.</p>
</body>
</html>
```

The logo for 'Learn 2 Earn Labs' is positioned in the background. It features the word 'LEARN' in blue, a large red number '2' inside a white hexagon, the word 'EARN' in blue, and the word 'LABS' in grey below them.

## Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Document</title>
  <style>
    p::after
    {
      content: "Hello World ";
      background-color: green;
      color: white;
      font-size: 30px;
      display: block;
    }
  </style>
</head>
<body>
  <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Voluptatibus at
  dignissimos cum, rem obcaecati minus officia similique animi ratione ex.</p>
</body>
</html>
```

LABS

## Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Document</title>
  <style>
    p::after
    {
      content: "Hello World ";
      background-color: red;
      color: white;
      font-size: 30px;
      display: block;
      margin-top: 20px;
      padding: 10px;
    }
  </style>
</head>
<body>

  <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Voluptatibus at
  dignissimos cum, rem obcaecati minus officia similique animi ratione ex.</p>

</body>
</html>
```

## 2) ::before / :before

- In CSS, ::before creates a pseudo-element that is the first child of the selected element.
- It is often used to add cosmetic content to an element with the content property.
- It is inline by default.

### Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Document</title>
  <style>
    p::before
    {
      content: "Hello World ";
      background-color: green;
      color: white;
      font-size: 30px;
    }
  </style>
</head>
<body>
```

<p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Voluptatibus at dignissimos cum, rem obcaecati minus officia similique animi ratione ex.</p>

```
</body>
</html>
```

LABS

## Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Document</title>
  <style>
    p::before
    {
      content: "Hello World ";
      background-color: green;
      color: white;
      font-size: 30px;
      display: block;
    }
  </style>
</head>
<body>
```

<p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Voluptatibus at dignissimos cum, rem obcaecati minus officia similique animi ratione ex.</p>

```
</body>
</html>
```

## Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Document</title>
  <style>
    p::before
    {
      content: "Hello World ";
      background-color: green;
      color: white;
      font-size: 30px;
      display: block;
      margin-bottom: 20px;
      padding: 10px;
    }
  </style>
</head>
<body>
```

<p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Voluptatibus at dignissimos cum, rem obcaecati minus officia similique animi ratione ex.</p>

```
</body>
</html>
```

### 3) ::first-letter / :first-letter

The ::first-letter CSS pseudo-element applies styles to the first letter of the first line of a block-level element, but only when not preceded by other content (such as images or inline tables).

#### Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Document</title>
  <style>
    p::first-letter
    {
      font-size: 50px;
      color: red;
    }
  </style>
</head>
<body>
```

<p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Inventore numquam ad veritatis facere laboriosam quisquam, mollitia ipsam nihil praesentium sed exercitationem doloribus excepturi doloremque id quia expedita consequatur vitae eos?</p>

```
</body>
</html>
```



#### 4) ::first-line / :first-line

The ::first-line CSS pseudo-element applies styles to the first line of a block-level element.

##### Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Document</title>
  <style>
    p::first-line
    {
      font-size: 50px;
      color: red;
    }
  </style>
</head>
<body>
```

<p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Inventore numquam ad veritatis facere laboriosam quisquam, mollitia ipsam nihil praesentium sed exercitationem doloribus excepturi doloremque id quia expedita consequatur vitae eos? </p>

```
</body>
</html>
```

## 5) ::selection

The ::selection CSS pseudo-element applies styles to the part of a document that has been highlighted by the user (such as clicking and dragging the mouse across text).

### Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Document</title>
  <style>
    p::selection
    {
      background: green;
      color: white;
    }
  </style>
</head>
<body>
```

<p>Select paragraph to see the effect --<br>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Velit nihil sequi quo minima accusantium, eos assumenda laboriosam inventore laudantium minus hic consequatur magnam voluptas repellendus consectetur saepe est sed aperiam.</p>

```
</body>
</html>
```

## 6) ::placeholder

The ::placeholder CSS pseudo-element represents the placeholder text in an <input> or <textarea> element.

### Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Document</title>
  <style>
    input::placeholder
    {
      color: green;
    }
  </style>
</head>
<body>

  <input type="text" placeholder="Enter your name ...">

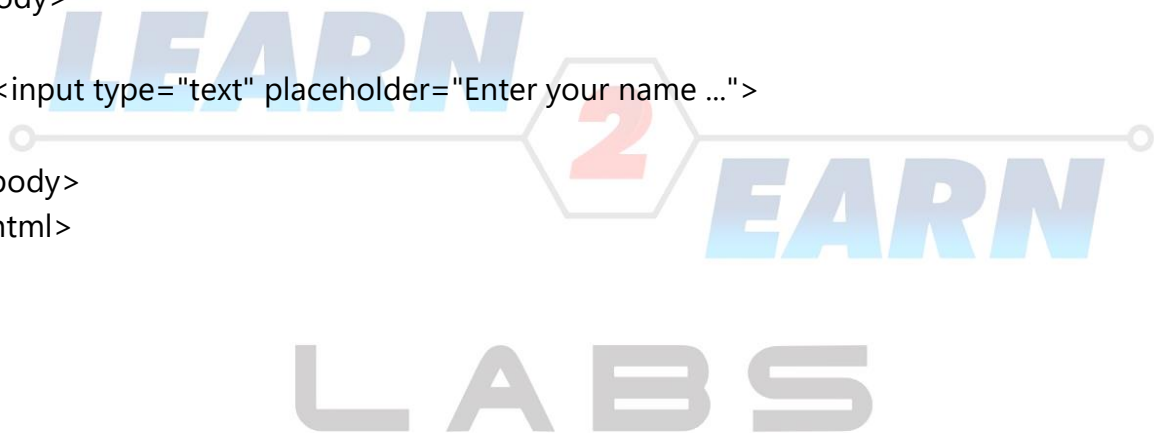
</body>
</html>
```

## Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Document</title>
  <style>
    input::placeholder
    {
      color: green;
      text-align: right;
    }
  </style>
</head>
<body>

  <input type="text" placeholder="Enter your name ...">

</body>
</html>
```



## The 'content' property

- We can style text or images in the content property almost any way we want.
- The content CSS property replaces an element with a generated value. Objects inserted using the content property are anonymous replaced elements.

The value for content can be :-

### a) String

- It specifies the "alt text" for the element.
- This value can be any number of text characters.
- Non-Latin characters must be encoded using their Unicode escape sequences: for example, \000A9 represents the copyright symbol.

Eg. -- content:"string";

### b) Image

- The image is inserted at it's exact dimensions and cannot be resized.
- Since things like gradients are actually images, a pseudo element can be a gradient.

E.g. -- content: url(/path/to/image.jpg);

### c) Nothing

- Useful for clearfix and inserting images as background-images (set width and height, and can even resize with background-size).

E.g. -- content: "";

### d) Counter

- The value of a CSS counter, generally a number. It can be displayed using either the counter() or counters() function.

E.g. -- content: counter(li)

**Note :-** We cannot render HTML using content property.

## Syntax

```
element
{
    content : value;
}
```

where,

a) element = p, div, span etc.

b) value = normal ,none ,counter() ,string ,url(url) ,attr(attribute) ,open-quote ,close-quote ,no-open-quote ,no-close-quote.




### Example -- "content : open-quote"

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Document</title>
  <style>
    div
    {
      width: 50%;
      height: 50vh;
    }
    div::before
    {
      content: open-quote;
      font-size: 50px;
    }
  </style>
</head>
<body>

  <h1>CSS Content Property</h1>

  <div>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Ut eum, blanditiis
sit recusandae magni quas quisquam doloribus odit. Praesentium dolore esse quasi
excepturi distinctio ad, enim exercitationem odio sequi deleniti.</div>

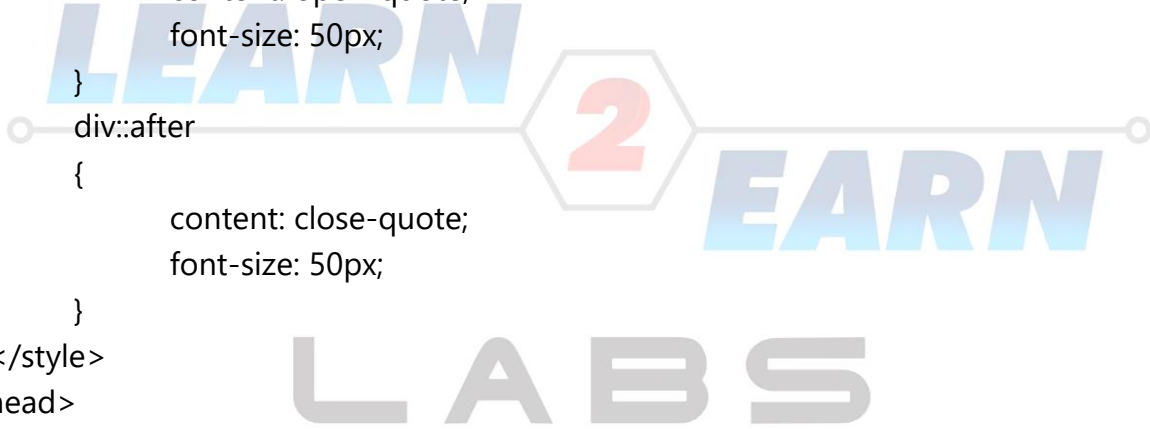
</body>
</html>
```

The logo for 'Learn 2 Earn Labs' is positioned in the background. It features the word 'LEARN' in large, blue, sans-serif capital letters. To its right is a white hexagon containing a red number '2'. Further right is the word 'EARN' in the same blue, sans-serif capital letters. Below these elements, the word 'LABS' is written in a smaller, grey, sans-serif font.

### Example -- "content : close-quote"

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Document</title>
  <style>
    div
    {
      width: 50%;
      height: 50vh;
    }
    div::before
    {
      content: open-quote;
      font-size: 50px;
    }
    div::after
    {
      content: close-quote;
      font-size: 50px;
    }
  </style>
</head>
<body>
  <h1>CSS Content Property</h1>
  <div>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Ut eum, blanditiis
sit recusandae magni quas quisquam doloribus odit. Praesentium dolore esse quasi
excepturi distinctio ad, enim exercitationem odio sequi deleniti.</div>

</body>
</html>
```

The logo for 'Learn 2 Earn Labs' is centered in the background. It features the word 'LEARN' in blue, a large red '2' inside a hexagon, the word 'EARN' in blue, and 'LABS' in grey below them. A horizontal line with circles at both ends passes behind the '2'.



### **Difference between (:) vs (::)**

- Every browser that supports the double colon (::) CSS3 syntax also supports just the (:) syntax, but IE 8 only supports the single-colon, so for now, it's recommended to just use the single-colon for best browser support.
- :: is the newer format intended to distinguish pseudo content from pseudo-selectors. If you don't need IE 8 support, feel free to use the double-colon.



## Miscellaneous Selectors

### 1) CSS Group selector

With Group selector we can group them into a comma-separated list to minimize the code in our style sheet.

#### Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Internal Stylesheets</title>
  <style>
    p, h1
    {
      color:red;
      font-size: 40px;
    }
  </style>
</head>
<body>
  <p>Hello Para 1</p>
  <h1>Hello Heading 1</h1>
</body>
</html>
```



## Combining Pseudo-Classes & Pseudo-Elements

If we want to make the first line of the first paragraph bold you could chain the :first-child and ::first-line selectors together.

### Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Document</title>
  <style>
    p:first-child::first-line {
      font-size: 120%;
      font-weight: bold;
    }
  </style>
</head>
<body>

  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Facilis obcaecati tempore
  temporibus eligendi. Officiis, fugit libero similique pariatur impedit, vero, adipisci culpa
  asperiores eos et iste at recusandae ad. Omnis?</p>

</body>
</html>
```

## CSS Important Rule

- This "important rule" changes the priority of processing within the style sheet.
- When an "important rule" is used on a style declaration, this declaration overrides any other declarations.
- The "!important" declarations that would help developers and users easily override normal specificity when making changes to their stylesheets.
- The !important means that "this is important", ignore all the subsequent rules, and apply !important rule and the !important keyword must be placed at the end of the line
- The !important rule in CSS means that all subsequent rules on an element are to be ignored, and the rule denoted by !important is to be applied.
- This rule overrides all previous styling rules -- the !important property increases its priority.
- We can override the !important rule, naturally by another one.
- We could apply this rule with any of the CSS property.

### When to use important rule

- \* The !important directive is helpful when we are testing and debugging a website.
- \* If we are not sure why a style is not being applied and think it may be a specificity conflict, add the !important declaration to our style to see if that fixes it — and if it does, change the order of the selectors and remove the !important directives from our production code.

### Syntax

```
element
{
    color : red !important;
}
```

Where,  
element = p, div, span etc.

Example : Before using important

```
<!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>

    h2
    {
      color: blue;
    }
    .elmClass
    {
      color: red;
    }
    #elmId
    {
      color: yellowgreen;
    }
  </style>
</head>
<body>
  <h1>CSS !important Rule</h1>
  <h2 class="elmClass" id="elmId">Web Development is used for creating
websites.</h2>

</body>
</html>
```

### Example : After using important

```
<!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>

    h2
    {
      color: blue;
    }
    .elmClass
    {
      color: red !important;
    }
    #elmId
    {
      color: yellowgreen;
    }
  </style>
</head>
<body>
  <h1>CSS !important Rule</h1>
  <h2 class="elmClass" id="elmId">Web Development is used for creating
websites.</h2>
</body>
</html>
```

## Colors in CSS

- The color property defines the text color (foreground color in general) of an element.
- The color CSS property represents a color in the RGB color space.
- Color keywords are case-insensitive identifiers which represent a specific color, e.g. red, green, blue, yellow, black, etc.
- The color property normally inherits the color value from their parent element, except the case of anchor elements.



## CSS Color Schemes

Colors in CSS most often specified in the following formats :-

### a) Color Name

In CSS Colors are also specified using predefined color names like "red","blue","yellow" etc.

#### Syntax

```
p
{
  color : "red"
}
```





## 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>
  <style>
    p
    {
      color: blue;
    }
  </style>
</head>
<body>
  <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Aliquam ratione
  nobis unde, minus distinctio deleniti nostrum magni maxime architecto velit quaerat est
  soluta eum aperiam, pariatur tenetur eius excepturi. Atque harum porro eligendi
  molestias, voluptatibus perspiciatis repudiandae modi? Doloremque voluptatum sed
  placeat, ipsum dignissimos aspernatur earum culpa similique asperiores! Mollitia
  dignissimos eius voluptatem. Doloribus, nam? Possimus aliquam quia illum architecto
  aut veritatis porro alias assumenda quis itaque commodi, dicta libero, distinctio
  consectetur quaerat molestiae perspiciatis error obcaecati eum mollitia! Tempora
  repellat earum, optio ipsum, quidem eligendi maiores officiis, aut neque enim possimus
  commodi delectus corrupti qui asperiores nam aliquid veniam.</p>
</body>
</html>
```

## b) Hexadecimal Style(Or, Hex Color)

- In this style, we define the color in 6 digit hexadecimal number (from 0 to F). It is denoted by '#'.  
#RRGGBB
- The first two digits indicate red color, next two green color and the last two blue color.

### Syntax

```
p
{
  color : #FF0000
}
```



## 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>
  <style>
    p
    {
      color: #FF0000;
    }
  </style>
</head>
<body>
  <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Aliquam ratione
  nobis unde, minus distinctio deleniti nostrum magni maxime architecto velit quaerat est
  soluta eum aperiam, pariatur tenetur eius excepturi. Atque harum porro eligendi
  molestias, voluptatibus perspiciatis repudiandae modi? Doloremque voluptatum sed
  placeat, ipsum dignissimos aspernatur earum culpa similique asperiores! Mollitia
  dignissimos eius voluptatem. Doloribus, nam? Possimus aliquam quia illum architecto
  aut veritatis porro alias assumenda quis itaque commodi, dicta libero, distinctio
  consectetur quaerat molestiae perspiciatis error obcaecati eum mollitia! Tempora
  repellat earum, optio ipsum, quidem eligendi maiores officiis, aut neque enim possimus
  commodi delectus corrupti qui asperiores nam aliquid veniam.</p>

</body>
</html>
```

### c) RGB Style (Red Green Blue)

- In this we need to give 3 numbers indicating the amount of red, green and blue colors respectively required in the mixed color.
- The range of each color is from 0 to 255.

#### Syntax

```
p  
{  
  color : rgb(255,0,0)  
}
```



## 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>
  <style>
    p
    {
      color: rgb(0, 0, 255);
    }
  </style>
</head>
<body>
  <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Aliquam ratione
  nobis unde, minus distinctio deleniti nostrum magni maxime architecto velit quaerat est
  soluta eum aperiam, pariatur tenetur eius excepturi. Atque harum porro eligendi
  molestias, voluptatibus perspiciatis repudiandae modi? Doloremque voluptatum sed
  placeat, ipsum dignissimos aspernatur earum culpa similique asperiores! Mollitia
  dignissimos eius voluptatem. Doloribus, nam? Possimus aliquam quia illum architecto
  aut veritatis porro alias assumenda quis itaque commodi, dicta libero, distinctio
  consectetur quaerat molestiae perspiciatis error obcaecati eum mollitia! Tempora
  repellat earum, optio ipsum, quidem eligendi maiores officiis, aut neque enim possimus
  commodi delectus corrupti qui asperiores nam aliquid veniam.</p>

</body>
</html>
```

#### d) RGBA Style (Red Green Blue Alpha)

This style allows us to make the color transparent according to our will. Alpha indicates the degree of transparency.

The range of green, blue and red is from 0 to 255 and that of alpha is from 0 to 1.

#### Syntax

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



## 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>
  <style>
    p
    {
      color: rgba(0, 0, 255,0.2);
    }
  </style>
</head>
<body>
  <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Aliquam ratione
nobis unde, minus distinctio deleniti nostrum magni maxime architecto velit quaerat est
soluta eum aperiam, pariatur tenetur eius excepturi. Atque harum porro eligendi
molestias, voluptatibus perspiciatis repudiandae modi? Doloremque voluptatum sed
placeat, ipsum dignissimos aspernatur earum culpa similique asperiores! Mollitia
dignissimos eius voluptatem. Doloribus, nam? Possimus aliquam quia illum architecto
aut veritatis porro alias assumenda quis itaque commodi, dicta libero, distinctio
consectetur quaerat molestiae perspiciatis error obcaecati eum mollitia! Tempora
repellat earum, optio ipsum, quidem eligendi maiores officiis, aut neque enim possimus
commodi delectus corrupti qui asperiores nam aliquid veniam.</p>

</body>
</html>
```

### e) HSL Style (Hue Saturation Lightness)

Here 'H' stands for hue, 'S' for Saturation and 'L' for Lightness. HSL color values are specified as:

#### 1) Hue

- Hue is the color of the image itself. It's range is from 0 to 360. 0 is for red, 120 is for green and 240 is for blue.
- In hue the "0 angle" represents "red color", "120 angle" represents "green color" & "360 angle" represents "blue color".

#### 2) Saturation

- Saturation is the intensity/purity of the hue. 0% is for a shade of gray and 100% is the full color.
- The Saturation parameter responsible for brightness / darkness of the particular color.

#### 3) Lightness

- Lightness is the color space's brightness. 0% is for black, 100% is for white.
- This parameter represents how "whitish" the color is.

### Syntax

```
p
{
  color : hsl(0,240%,120%)
}
```



## 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>
  <style>
    p
    {
      color: hsl(20, 80%, 50%);
    }
  </style>
</head>
<body>
  <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Aliquam ratione
  nobis unde, minus distinctio deleniti nostrum magni maxime architecto velit quaerat est
  soluta eum aperiam, pariatur tenetur eius excepturi. Atque harum porro eligendi
  molestias, voluptatibus perspiciatis repudiandae modi? Doloremque voluptatum sed
  placeat, ipsum dignissimos aspernatur earum culpa similique asperiores! Mollitia
  dignissimos eius voluptatem. Doloribus, nam? Possimus aliquam quia illum architecto
  aut veritatis porro alias assumenda quis itaque commodi, dicta libero, distinctio
  consectetur quaerat molestiae perspiciatis error obcaecati eum mollitia! Tempora
  repellat earum, optio ipsum, quidem eligendi maiores officiis, aut neque enim possimus
  commodi delectus corrupti qui asperiores nam aliquid veniam.</p>
</body>
</html>
```

## f) Transparent

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Document</title>
  <style>
    p
    {
      color:transparent;
    }
  </style>
</head>
<body>
```

<p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Voluptatibus at dignissimos cum, rem obcaecati minus officia similique animi ratione ex.</p>

```
</body>
</html>
```

LABS

## Transparency

- The opacity CSS property specifies the transparency of an element.
- The opacity property takes a value a value from 0.0 to 1.0. A setting of opacity: 1; would make the element completely opaque (i.e. 0% transparent), whereas opacity: 0; would make the element completely transparent (i.e. 100% transparent).
- We can also make transparent images using CSS Opacity.

## Syntax

```
div
{
  opacity:0.2;
}
```



## Example

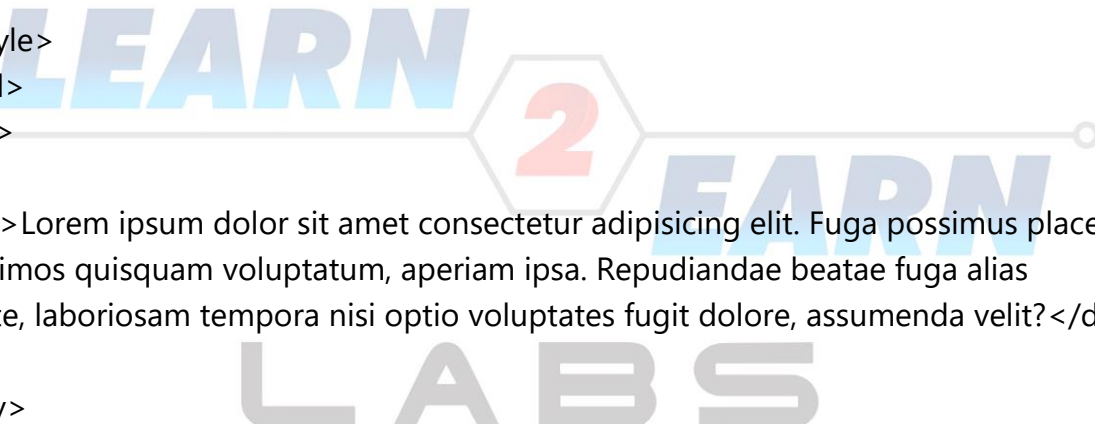
```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Document</title>
  <style>

    div
    {
      background-color: red;
      color: white;
      opacity: 0.7;
    }

  </style>
</head>
<body>

  <div>Lorem ipsum dolor sit amet consectetur adipisicing elit. Fuga possimus placeat
dignissimos quisquam voluptatum, aperiam ipsa. Repudiandae beatae fuga alias
sapiente, laboriosam tempora nisi optio voluptates fugit dolore, assumenda velit?</div>

</body>
</html>
```

The logo for 'Learn 2 Earn Labs' is centered in the background. It features the word 'LEARN' in blue, a large red number '2' inside a hexagon, the word 'EARN' in blue, and 'LABS' in grey below it.