Topic:

- What is CSS? Advantages of CSS, CSS Structure and Syntax.
- Types of CSS: Internal, External, Inline.
- CSS Color, Background and Border.
- CSS Margin, Padding, Height and Width.
- CSS Text, Fonts. CSS Icons and Links.
- CSS List and Tables.
- CSS Pseudo Class and CSS Pseudo Elements.

❖ What is CSS? Advantages of CSS, CSS Structure and Syntax.

- ✔ CSS stands for Cascading style sheets. It describes to the user how to display HTML elements on the screen in a proper format. CSS is the language that is used to style HTML documents. In simple words, cascading style sheets are a language used to simplify the process of making a webpage.
- ✔ CSS is used to handle some parts of the webpage. With the help of CSS, we can control the colour of text and style of fonts, and we can control the spacing between the paragraph and many more things. CSS is easy to understand but provides strong control on the Html documents.CSS is combined with HTML.

Advantages of CSS

Here are the following advantages of CSS, such as:

- ✔ **Faster page speed:** It has a faster page speed than other code's page speeds. With the help of the CSS rule, we can apply it to all occurrences of certain tags in HTML documents.
- ✔ **Better user experience:** CSS makes a webpage very attractive to the eyes. Also, CSS makes it user-friendly. When the button or text is in a proper format, it improves the user experience.
- ✔ **Quicker Development time:** With the help of CSS, we can specify the format and style the multiple pages into one code string. In cascading style sheet, we can make a duplicate copy of several website pages.If we make a webpage, it has the same formatting, looks, and feel, so with the help of the CSS rule for one page, and it is sufficient for all the pages.
- ✔ **Easy Formatting changes:** In CSS, if we need to make changes in the format, it is very easy; we only need to change the one-page format it will automatically apply to the other pages of CSS.There is no need to correct individual pages in a CSS style sheet. If we fix a CSS style sheet, it will automatically update the other CSS style sheet.
- ✔ **Compatibility:** Compatibility is very important in today's age. If we create any webpage, it should be very responsive and user-friendly. CSS is used with Html to make webpage design responsive.

Use CSS

- ✔ As we all know, CSS is a powerful style sheet language used to control the HTML document to improve the webpage design.
- ✔ **CSS provides efficiency in webpage design:** It also provides updates so our webpage works appropriately. With the help of CSS, we can create and apply those rules within the website. If we create a webpage design separately, we can make changes in our style sheet, and it will affect all the style sheets.
- ✔ **CSS provides faster page download:** CSS helps with faster page download because when we download a page, we get the cache that helps to load a page, but with the help of CSS, we can lead to load a lighter page which helps to improve the performance.

✔ **CSS is easy to work:** In CSS, we can visual aspect of the website separate entirely from the content; using CSS, we can create a website that allows us to make quick layout.

CSS Syntax

✔ The CSS provides the style to the HTML element, which the browser interprets. After being interpreted by the browser, the CSS style property will be applied to all the elements of the HTML. We can provide style property to the HTML element in three parts. These three parts are as follows.

✔ **Selector**

It is an HTML tag. All the style properties of the CSS will be applied to the selector. The selector tag like <h1> or <table> etc.

✔ **Property**

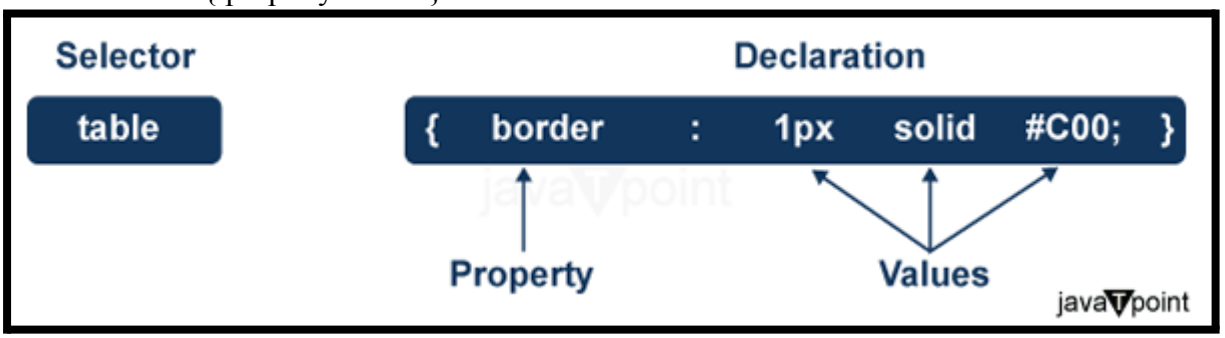
It is a type of attribute that is present in HTML tags. All the attributes of the HTML will be converted to the CSS properties. The CSS properties like *color, border*, etc.

✔ **Value**

In HTML, these are assigned to the properties. For example, *the color* property can have a value of either *red* or *#F1F1F1*, etc.

Syntax:

We must provide the CSS property to the HTML element in a proper way. We must follow the syntax below to implement the CSS property.
selector { property: value }



Example:

Let's implement the CSS property by seeing an example.

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Document</title>
 <style>
  /* styles.css */
body {
 font-family: Arial, sans-serif;
 margin: 0;
 padding: 0;
 background-color: #f2f2f2;
```

```
}
#header {
 background-color: #333;
 color: white;
 text-align: center;
 padding: 20px;
}
h1 {
 font-size: 36px;
 margin-bottom: 20px;
}
p {
 font-size: 18px;
 margin-bottom: 10px;
}
.button {
 background-color: #ff0000;
 color: white;
 padding: 10px 20px;
 border: none;
 text-align: center;
 text-decoration: none;
 display: inline-block;
 font-size: 16px;
 border-radius: 5px;
 cursor: pointer;
}
.button:hover {
 background-color: #990000;
}
 </style>
</head>
<body>
 <div id="header">
  <h1>Hello, CSS Example!</h1>
  <p>Welcome to javaTpoint. Here is a styled button:</p>
  <button class="button">Click Me!</button>
 </div>
</body>
</html>
```

❖ **Types of CSS: Internal, External, Inline.**

CSS is added to HTML pages to format the document according to information in the style sheet. There are three ways to insert CSS in HTML documents.

- ✔ Inline CSS
- ✔ Internal CSS
- ✔ External CSS

❖ **Inline CSS**

- ✔ We can apply CSS in a single element by inline CSS technique.
- ✔ The inline CSS is also a method to insert style sheets in HTML document. This method mitigates some advantages of style sheets so it is advised to use this method sparingly.
- ✔ If you want to use inline CSS, you should use the style attribute to the relevant tag.

Syntax:

```
<htmltag style="cssproperty1:value; cssproperty2:value;"> </htmltag>
```

Example:

```
<h2 style="color:red;margin-left:40px;">Inline CSS is applied on this headin
g.</h2>
<p>This paragraph is not affected.</p>
```

Output:

Inline CSS is applied on this heading.

This paragraph is not affected.

Disadvantages of Inline CSS

- ✔ You cannot use quotations within inline CSS. If you use quotations the browser will interpret this as an end of your style value.
- ✔ These styles cannot be reused anywhere else.
- ✔ These styles are tough to be edited because they are not stored at a single place.
- ✔ It is not possible to style pseudo-codes and pseudo-classes with inline CSS.
- ✔ Inline CSS does not provide browser cache advantages.

❖ **Internal CSS**

- ✔ The internal style sheet is used to add a unique style for a single document. It is defined in <head> section of the HTML page inside the <style> tag.

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: linen;
}
h1 {
  color: red;
  margin-left: 80px;
}
</style>
</head>
<body>
<h1>The internal style sheet is applied on this heading.</h1>
<p>This paragraph will not be affected.</p>
</body>
</html>
```

❖ **External CSS**

- ✔ The external style sheet is generally used when you want to make changes on multiple pages. It is ideal for this condition because it facilitates you to change the look of the entire web site by changing just one file.
- ✔ It uses the <link> tag on every pages and the <link> tag should be put inside the head section.

Example:

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

- ✔ The external style sheet may be written in any text editor but must be saved with a .css extension. This file should not contain HTML elements.

❖ CSS Color, Background and Border.

CSS Colors:

- ✔ The color property in CSS is used to set the color of HTML elements. Typically, this property is used to set the background color or the font color of an element.
- ✔ In CSS, we use color values for specifying the color. We can also use this property for the border-color and other decorative effects.
- ✔ We can define the color of an element by using the following ways:
  - RGB format.
  - RGBA format.
  - Hexadecimal notation.
  - HSL.
  - HSLA.
  - Built-in color.
- ✔ Let's understand the syntax and description of the above ways in detail.

RGB Format

- ✔ RGB format is the short form of '**RED GREEN** and **BLUE**' that is used for defining the color of an HTML element simply by specifying the values of R, G, B that are in the range of 0 to 255.
- ✔ The color values in this format are specified by using the **rgb()** property. This property allows three values that can either be in percentage or integer (range from 0 to 255).
- ✔ This property is not supported in all browsers; that's why it is not recommended to use it.

Syntax

```
color: rgb(R, G, B);
```

RGBA Format

- ✔ It is almost similar to RGB format except that **RGBA** contains **A (Alpha)** that specifies the element's transparency. The value of alpha is in the range **0.0 to 1.0**, in which **0.0** is for fully transparent, and **1.0** is for not transparent.

Syntax

```
color:rgba(R, G, B, A);
```

Hexadecimal notation

- ✔ Hexadecimal can be defined as a six-digit color representation. This notation starts with the # symbol followed by six characters ranges from 0 to F. In hexadecimal notation, the first two digits represent the red (RR) color value, the next two digits

represent the green (GG) color value, and the last two digits represent the blue (BB) color value.

- ✔ The black color notation in hexadecimal is #000000, and the white color notation in hexadecimal is #FFFFFF. Some of the codes in hexadecimal notation are #FF0000, #00FF00, #0000FF, #FFFF00, and many more.

Syntax

color:#(0-F)(0-F)(0-F)(0-F)(0-F)(0-F);

 Short Hex codes

- ✔ It is a short form of hexadecimal notation in which every digit is recreated to arrive at an equivalent hexadecimal value.
- ✔ For example, #7B6 becomes #77BB66 in hexadecimal.
- ✔ The black color notation in short hex is #000, and the white color notation in short hex is #FFF. Some of the codes in short hex are #F00, #0F0, #0FF, #FF0, and many more.

 HSL

- ✔ It is a short form of **Hue, Saturation,** and **Lightness**. Let's understand them individually.
- ✔ **Hue:** It can be defined as the degree on the color wheel from 0 to 360. 0 represents red, 120 represents green, 240 represents blue.
- ✔ **Saturation:** It takes value in percentage in which 100% represents fully saturated, i.e., no shades of gray, 50% represent 50% gray, but the color is still visible, and 0% represents fully unsaturated, i.e., completely gray, and the color is invisible.
- ✔ **Lightness:** The lightness of the color can be defined as the light that we want to provide the color in which 0% represents black (there is no light), 50% represents neither dark nor light, and 100% represents white (full lightness).

Let's see the syntax of HSL in color property.

Syntax

color:hsl(H, S, L);

 HSLA

- ✔ It is entirely similar to HSL property, except that it contains A (alpha) that specifies the element's transparency. The value of alpha is in the range 0.0 to 1.0, in which 0.0 indicates fully transparent, and 1.0 indicates not transparent.

Syntax

color:hsla(H, S, L, A);

 Built-in Color

- ✔ As its name implies, built-in color means the collection of previously defined colors that are used by using a name such as red, blue, green, etc.

Syntax

color: color-name;

Let's see the list of built-in colors along with their decimal and hexadecimal values.

S.no.	Color name	Hexadecimal Value	Decimal Value or rgb() value
1.	Red	#FF0000	rgb(255,0,0)

2.	Orange	#FFA500	rgb(255,165,0)
3.	Yellow	#FFFF00	rgb(255,255,0)
4.	Pink	#FFC0CB	rgb(255,192,203)
5.	Green	#008000	rgb(0,128,0)
6.	Violet	#EE82EE	rgb(238,130,238)
7.	Blue	#0000FF	rgb(0,0,255)
8.	Aqua	#00FFFF	rgb(0,255,255)
9.	Brown	#A52A2A	rgb(165,42,42)
10.	White	#FFFFFF	rgb(255,255,255)
11.	Gray	#808080	rgb(128,128,128)
12.	Black	#000000	rgb(0,0,0)

The illustration of CSS colors, which includes the above properties, is given below.

Example

```
<html>
  <head>
    <title>CSS hsl color property</title>
    <style>
      h1{
        text-align:center;
      }
      #rgb{
        color:rgb(255,0,0);
      }
      #rgba{
        color:rgba(255,0,0,0.5);
      }
      #hex{
        color:#EE82EE;
      }
      #short{
        color: #E8E;
      }
      #hsl{
        color:hsl(0,50%,50%);
      }
      #hsla{
        color:hsla(0,50%,50%,0.5);
      }
```

```
        #built{
          color:green;
        }
      </style>
    </head>
    <body>
      <h1 id="rgb">
         Hello World. This is RGB format.
      </h1>
      <h1 id="rgba">
        Hello World. This is RGBA format.
     </h1>
     <h1 id="hex">
      Hello World. This is Hexadecimal format.
   </h1>
   <h1 id="short">
    Hello World. This is Short-hexadecimal format.
  </h1>
   <h1 id="hsl">
    Hello World. This is HSL format.
  </h1>
  <h1 id="hsla">
    Hello World. This is HSLA format.
  </h1>
  <h1 id="built">
    Hello World. This is Built-in color format.
  </h1>
     </body>
  </html>
```

❖ CSS Background

CSS background property is used to define the background effects on element. There are 5 CSS background properties that affects the HTML elements:

1. background-color
2. background-image
3. background-repeat
4. background-attachment
5. background-position

CSS background-color

The background-color property is used to specify the background color of the element.
You can set the background color like this:

```
<!DOCTYPE html>
<html>
<head>
<style>
h2,p{
   background-color: #b0d4de;
}
</style>
```

```
</head>
<body>
<h2>My first CSS page.</h2>
<p>Hello Javatpoint. This is an example of CSS background-color.</p>
</body>
</html>
```

Output:

My first CSS page.

Hello Javatpoint. This is an example of CSS background-color.

- **CSS background-image**

The background-image property is used to set an image as a background of an element. By default the image covers the entire element. You can set the background image for a page like this.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
background-image: url("paper1.gif");
margin-left:100px;
}
</style>
</head>
<body>
<h1>Hello Javatpoint.com</h1>
</body>
</html>
```

- **CSS background-repeat**

By default, the background-image property repeats the background image horizontally and vertically. Some images are repeated only horizontally or vertically.
The background looks better if the image repeated horizontally only.

background-repeat: repeat-x;

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
   background-image: url("gradient_bg.png");
   background-repeat: repeat-x;
}
</style>
</head>
<body>
<h1>Hello Javatpoint.com</h1>
</body>
```

```
    </html>
background-repeat: repeat-y;
    <!DOCTYPE html>
    <html>
    <head>
    <style>
    body {
       background-image: url("gradient_bg.png");
       background-repeat: repeat-y;
    }
    </style>
    </head>
    <body>
    <h1>Hello Javatpoint.com</h1>
    </body>
    </html>
```

- **CSS background-attachment**

The background-attachment property is used to specify if the background image is fixed or scroll with the rest of the page in browser window. If you set fixed the background image then the image will not move during scrolling in the browser. Let?s take an example with fixed background image.

```
background: white url('bbb.gif');
background-repeat: no-repeat;
background-attachment: fixed;
```

- **CSS background-position**

The background-position property is used to define the initial position of the background image. By default, the background image is placed on the top-left of the webpage.
You can set the following positions:

```
center
top
bottom
left
right
background: white url('good-morning.jpg');
background-repeat: no-repeat;
background-attachment: fixed;
background-position: center;
```

- **CSS Margin, Padding, Height and Width.**

✔ **CSS Margin:**

CSS Margin property is used to define the space around elements. It is completely transparent and doesn't have any background color. It clears an area around the element.
Top, bottom, left and right margin can be changed independently using separate properties. You can also change all properties at once by using shorthand margin property.
There are following CSS margin properties:

CSS Margin Properties

Property	Description
margin	This property is used to set all the properties in one declaration.
margin-left	it is used to set left margin of an element.
margin-right	It is used to set right margin of an element.
margin-top	It is used to set top margin of an element.
margin-bottom	It is used to set bottom margin of an element.

CSS Margin Values

These are some possible values for margin property.

Value	Description
auto	This is used to let the browser calculate a margin.
length	It is used to specify a margin pt, px, cm, etc. its default value is 0px.
%	It is used to define a margin in percent of the width of containing element.
inherit	It is used to inherit margin from parent element.

CSS margin Example

You can define different margin for different sides for an element.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
   background-color: pink;
}
p.ex {
   margin-top: 50px;
   margin-bottom: 50px;
   margin-right: 100px;
   margin-left: 100px;
}
</style>
</head>
<body>
<p>This paragraph is not displayed with specified margin. </p>
<p class="ex">This paragraph is displayed with specified margin.</p>
</body>
</html>
```

Output:

This paragraph is not displayed with specified margin.

This paragraph is displayed with specified margin.

Margin: Shorthand Property

CSS shorthand property is used to shorten the code. It specifies all the margin properties in one property.

There are four types to specify the margin property. You can use one of them.

1. margin: 50px 100px 150px 200px;
2. margin: 50px 100px 150px;
3. margin: 50px 100px;
4. margin 50px;

margin: 50px 100px 150px 200px;

It identifies that:

top margin value is 50px
right margin value is 100px
bottom margin value is 150px
left margin value is 200px

margin: 50px 100px 150px;

It identifies that:
top margin value is 50px
left and **right** margin values are 100px
bottom margin value is 150px

margin: 50px 100px;

It identifies that:
top and **bottom** margin values are 50px
left and right margin values are 100px

margin: 50px;

It identifies that:

✔ **top** right bottom and left margin values are 50px

✔ **CSS Padding**

CSS Padding property is used *to define the space between the element content and the element border*.

It is different from CSS margin in the way that CSS margin defines the space around elements. CSS padding is affected by the background colors. It clears an area around the content.

Top, bottom, left and right padding can be changed independently using separate properties. You can also change all properties at once by using shorthand padding property.

CSS Padding Properties

Property	Description
padding	It is used to set all the padding properties in one declaration.
padding-left	It is used to set left padding of an element.
padding-right	It is used to set right padding of an element.

padding-top	It is used to set top padding of an element.
padding-bottom	It is used to set bottom padding of an element.

CSS Padding Values

Value	Description
length	It is used to define fixed padding in pt, px, em etc.
%	It defines padding in % of containing element.

CSS Padding Example

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
   background-color: pink;
}
p.padding {
   padding-top: 50px;
   padding-right: 100px;
   padding-bottom: 150px;
   padding-left: 200px;
}
</style>
</head>
<body>
<p>This is a paragraph with no specified padding.</p>
<p class="padding">This is a paragraph with specified paddings.</p>
</body>
</html>
```

This is a paragraph with no specified padding.

This is a paragraph with specified paddings.

✔ CSS Width

The **CSS width property** is used *to set the width of the content area of an element*.
It does not include padding borders or margins. It sets width of the area inside the padding, border, and margin of the element.

CSS width values

Value	Description
auto	It is a default value. it is used to calculate the width.
length	It is used to define the width in px, cm etc.

%	It defines the width of the containing block in %.
initial	It is used to set the property to its default value.
inherit	It is used to inherit the property from its parent element.

CSS Width Example: width in px

```
<!DOCTYPE html>
<html>
<head>
<style>
img.normal {
   width: auto;
}
img.big {
   width: 150px;
}
p.ex {
   height: 150px;
   width: 150px;
}
</style>
</head>
<body>
<img class="normal" src="good-morning.jpg" width="95" height="84"><br>
<img class="big" src="good-morning.jpg" width="95" height="84">
<p class="ex">The height and width of this paragraph is 150px.</p>
<p>This is a paragraph.</p>
</body>
</html>
```

Output:

The height and width of this paragraph is 150px.

This is a paragraph.

CSS Width Example: width in %

The percent width is a measurement unit for the containing block. It is great for images.

```
<!DOCTYPE html>
<html>
<head>
```

```
<style>
img.normal {
    width: auto;
}
img.big {
    width: 50%;
}
img.small {
    width: 10%;
}
</style>
</head>
<body>
<img class="normal" src="good-morning.jpg" width="95" height="84"><br>
<img class="big" src="good-morning.jpg" width="95" height="84"><br>
<img class="small" src="good-morning.jpg" width="95" height="84">
</body>
</html>
```

Output:

CSS height property

✔ This CSS property sets the height of an element. It is used to set the height of content area of an element.

✔ It does not include padding borders or margins, whereas it sets the height of the area inside the padding, border, and margin of the element. It can accept the length and percentage values. But it does not allow negative values.

✔ If we set the height to a numeric value (like in px, %, etc.), the content can be overflow if it does not fit in the given height. We can manage the overflowing content by defining the **overflow** property.

✔ If the height of the container is not explicitly defined, and the element is not absolutely positioned (i.e., **position: absolute;**), the value of **height** property is set to **auto**. The min-height and max-height properties can also be used to control the size.

Syntax

height: auto | length | initial | inherit;

Property Values

The values of this property are tabulated as follows.

Value	Description
auto	It is a default value. Using this value browser is responsible for calculating the height of the element. Negative values are not allowed.
length	It specifies the height of an element using the length units such as px, cm, pt, etc. Negative values are not allowed.
%	It defines the height of the container in %. Negative values are not allowed.
initial	It is used to set the property to its default value.
inherit	It is used to inherit the property from its parent element.

Now, we will see some of the examples to understand this property more clearly.

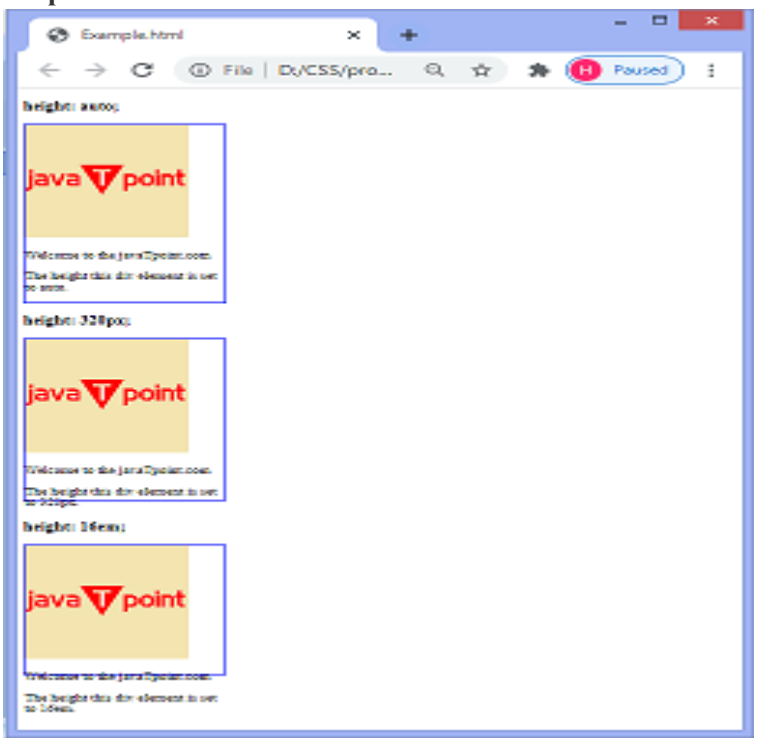
Example

Here, we are using the keyword value **auto** and the length values of height property in **px** and **em**.

```
<!DOCTYPE html>
<html>
<head>
<style>
#auto{
height: auto;
width: 275px;
border: 2px solid blue;
}
#px{
height: 320px;
width: 275px;
border: 2px solid blue;
}
#em{
height: 16em;
width: 275px;
border: 2px solid blue;
}
p{
font-size: 20px;
}
```

```
</style>
</head>
<body>
<h2> height: auto; </h2>
<div id ="auto">
<img src="jtp.png">
<p> Welcome to the javaTpoint.com </p>
<p> The height this div element is set to auto. </p>
</div>
<h2> height: 320px; </h2>
<div id ="px">
<img src="jtp.png">
<p> Welcome to the javaTpoint.com </p>
<p> The height this div element is set to 320px. </p>
</div><br>
<h2> height: 16em; </h2>
<div id ="em">
<img src="jtp.png">
<p> Welcome to the javaTpoint.com </p>
<p> The height this div element is set to 16em. </p>
</div>
</body>
</html>
```

Output



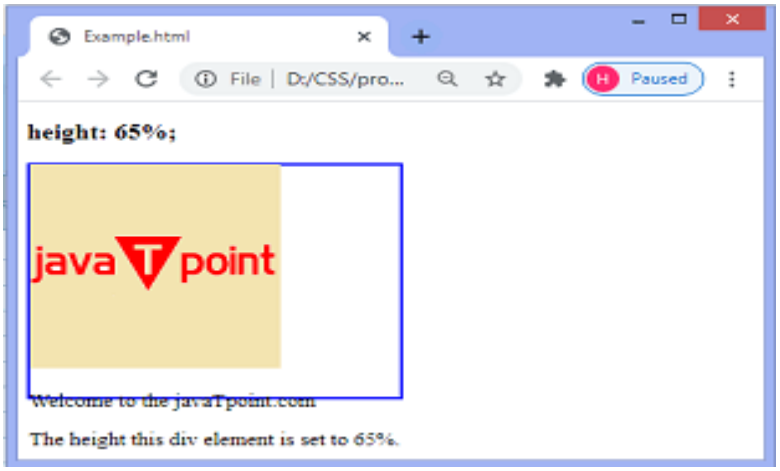
Example

Here, we are specifying the value of **height** property in percentage.

```
<!DOCTYPE html>
<html>
<head>
<style>
```

```
#per{
        position: absolute;
        width: auto;
        height: 65%;
        border: 2px solid blue;
}
p{
            font-size: 20px;
}
</style>
</head>
<body>
<h2> height: 65%; </h2>
<div id ="per">
<img src="jtp.png">
<p> Welcome to the javaTpoint.com </p>
<p> The height this div element is set to 65%. </p>
</div>
</body>
</html>
```

Output:



☐ CSS Text, Fonts. CSS Icons and Links.

❖ CSS Text:

- ✔ A text refers to a piece of written or printed information in the form of words or characters that can be read and understood. Texts can include content such as books, articles, emails, messages, web pages, etc.
- ✔ Styling a text involves modifying its appearance to make it more visually appealing or to convey a particular message. This chapter demonstrates how to manipulate text using CSS properties.
- ✔ There are several ways to style text. Following properties provided by CSS can be used for the styling purpose:

❖ **color**: Sets the color of the text.
❖ **text-align** Sets the alignment of the text.
❖ **text-align-last**: Sets the alignment of the last line of a block of text.

- **direction**: Sets the direction of the text of an element.
- **text-indent**: Sets the indentation of the first line of the text.
- **letter-spacing**: Specifies the space between the letters or characters that make up a word or text.
- **word-spacing**: Specifies the space between the words in a text.
- **white-space**: Controls the white space flow inside the text in an element.
- **text-decoration**: Adds an underline, overline, and strikethrough over a piece of text.
- **text-decoration-skip**: Determines what part of the content of the element, where text decoration is affecting, needs to be skipped.
- **text-decoration-skip-ink**: Specifies how the overline and underline text decoration lines are drawn around glyph ascenders and descenders.
- **text-transform**: Converts text to uppercase or lowercase letters.
- **text-emphasis**: Applies emphasis marks to text (except spaces and control characters).
- **text-shadow**: Adds shadow to the text.
- **line-break**: Controls how to set the rule for a line break.
- **word-break**: Controls how to set the rule for a word break.

1. CSS Text - Text Color

Altering the color of the text can add visual interest or align with a specific design scheme.The **color** property is used to set the color of the text. The possible values for this property are as follows:

- **Color Name**: Example = red, blue, green.
- **Hexadecimal value**: Example = #ff00ff.
- **RGB value**: Example = rgb(125, 255, 0).

Example:

```
<html>
<head>
</head>
<body>
  <h2>Text Color</h2>
    <p style = "color: blueviolet;">
      Color Name
    </p>
    <p style = "color:#ff00ff;">
      Hexadecimal value
    </p>
    <p style = "color: rgb(255,124,100);">
      RGB value
    </p>
</body>
</html>
```

Output:

Text Color

Color Name

Hexadecimal value

RGB value

2. CSS Text - Text Alignment

The position or placement of text on a page is called its alignment. The text is aligned based on the left and right margins of the page.
The CSS property that defines the alignment of the text is text-align. It sets the horizontal alignment of the text.
Following are the possible values that this property can hold:

- **start**: Same as left, if direction is LTR and right if direction is RTL.
- **end**: Same as right, if direction is LTR and left if direction is RTL.
- **left**: Aligned with the left-margin.
- **right**: Aligned with the right-margin.
- **center**: Aligned at the center of the page.
- **justify**: Aligned with both the margins.
- **justify-all**: Same as justify, making the last line justified as well.
- **match-parent**: Similar to inherit. Values of start and right taken from parent's value and replaced by left and right.

*When the **direction** is ltr, the default alignment is **left-aligned** and when the **direction** is rtl, the default alignment is **right-aligned**.*

Here is an example:

```
<html>
<head>
</head>
<body>
  <h2>Text Alignment</h2>
    <p style="text-align: left;">Text Left Alignment.</p>
    <p style="text-align: right;">Text Right Alignment.</p>
    <p style="text-align: center;">Text Center Alignment.</p>
    <p style="text-align: justify; border: 2px solid red; width: 200px; height: 100px;">
      Text Justify Alignment. This alignment aligns the text based on both the margins, left and right.
    </p>
</body>
</html>
```

Output:

Text Alignment

Text Left Alignment.

Text Right Alignment.

Text Center Alignment.

Text Justify Alignment. This alignment aligns the text based on both the margins, left and right.

3. CSS Text - Direction

Text direction refers to the orientation of text characters within a document or element. It determines whether text should be displayed from left to right (LTR) or right to left (RTL) based on the writing system used.
In CSS, you can set the text direction using the **direction** property. The **direction** property accepts two main values:

- **LTR (Left-to-Right)**: Default value, used for languages that are written from left to right, like English. You don't need to specify this value explicitly unless you want to override an inherited right-to-left direction.
- **RTL (Right-to-Left)**: Used for languages that are written from right to left, such as Arabic or Hebrew. When using rtl, text will be aligned right by default.

Additionally, CSS provides a shorthand property, **unicode-bidi**, to control the bidi algorithm, which specifies how characters with different writing directions are displayed when they appear in the same paragraph. The most common value for unicode-bidi is **normal**, which allows the browser to handle text direction automatically.

Here is an example:

```
<html>
<head>
</head>
<body>
  <p style = "direction: rtl;">
    Right to Left
  </p>
  <p style = "direction: ltr;">
    Left to Right
  </p>
</body>
</html>
```

4. CSS Text - Text Decoration

The property **text-decoration** helps in adding extra decoration to the text, such as, adding a line (underline, strikethrough, overline) and color, style and thickness to the line.

It is a shorthand property to the following properties:

- **text-decoration-line**: Sets the type of decoration line (underline, strikethrough or overline).
- **text-decoration-color**: Sets the color to the text decoration.
- **text-decoration-style**: Sets the style to the text decoration (dotted, dashed, solid, wavy, double, etc.)
- **text-decoration-thickness**: Sets the thickness to the text decoration.

Example:

```
<html>
<head>
</head>
<body>
  <h2>Text Decoration</h2>
    <p style="text-decoration: overline solid red 5px;">Overline text decoration.</p>
    <p style="text-decoration: line-through solid green 1px;">Line-through text decoration.</p>
    <p style="text-decoration: underline dashed 2pt blue;">Underline text decoration.</p>
</body>
</html>
```

5. CSS Text - Text Decoration Skip

The **text-decoration-skip** property determines what part of the content of the element, where text decoration is affecting, needs to be skipped.

The **text-decoration-skip** property can have following values:

- **none**: Skips nothing and text decoration is applied for all the text content.
- **objects**: Skips the whole margin box of the element,such as for an image or inline-block.
- **spaces**: Skips all spacing.
- **leading-spaces**: Same as spaces, skipping only leading spaces.
- **trailing-spaces**: Same as spaces, skipping only trailing spaces.
- **edges**: Edges such as start and end of text insets slightly. Separate underlines for adjacent elements.
- **box-decoration**: Skips the text decoration over margin, border and padding areas of a box.
- **initial**: Sets the text-decoration-skip value to its default value (none).
- **inherit**: Inherits the text-decoration-skip value from its parent element.
- **unset**: Removes any previously set text-decoration-skip value**.**

This property is supported only on Safari as of now.

Here is an example:

```
<html>
<head>
</head>
<body>
  <h2>Text Decoration Skip</h2>
    <p style="font-size:3em; text-decoration: underline solid red 5px; text-decoration-skip: edges;">Observe the edges of the line.</p>
    <p style="font-size:3em; text-decoration: underline solid green 5px; text-decoration-skip: spaces;">Its text decoration skip spaces.</p>
</body>
</html>
```

6. CSS Text - Text Indentation

Indentation is the space between the margin and the first line of text. Proper indentation enhances the readability and clarity of text on a page.

CSS also provides a property to set the text indentation and that is **text-indent**. The following values can be passed to this property:

- **length**: Any specific length {pixels (px), points (pt), ems (em), etc}. Default value is 0.
- **percentage (%)**: The value passed is in relation to the percentage of the width of the parent element.
- **each-line**: Affects the first line of a block of text along with each line after a forced line break.
- **hanging**: Indentation is inverted, except for the first line.
- **initial**: Sets the text-indent to its default value.
- **inherit**: Allows to inherit the text-indent value from its parent element.

***Note**: As of the now, the values **each-line** and **hanging** are not supported by any of the browsers.*

```
<html>
<head>
</head>
<body>
  <h2>Text indentation</h2>
    <p style="text-indent: 2cm;">Text indentation at 2 cm.</p>
    <p style="text-indent: 2in;">Text indentation at 2 inches.</p>
    <p style="text-indent: 20%;">Text indentation at 20%.</p>
</body>
</html>
```

Output:

Text indentation

Text indentation at 2 cm.

Text indentation at 2 inches.

Text indentation at 20%.

7. CSS Text - Letter Spacing

Adjusting the space between the letters in a text, impacts the readability and overall aesthetics of the webpage.

The property **letter-spacing** is used to adjust the space between the letters of a text. The space can either be increased or decreased between the letters.

Following are the possible values that this property can have:

- **normal**: Default value and represents the normal amount of space between letters.
- **length**: Any specific length {pixels (px), points (pt), ems (em), or percentages (%)}.
- **initial**: Sets the letter-spacing to its default value.
- **inherit**: Allows to inherit the letter-spacing value from its parent element.

Note: The actual spacing between letters may vary depending on the font being used.

Here is an example:

```
<html>
<head>
</head>
<body>
  <h2>Letter spacing</h2>
    <p style="letter-spacing: normal;">Letter spacing normal.</p>
    <p style="letter-spacing: 5px;">Letter spacing increased.</p>
    <p style="letter-spacing: -1px;">Letter spacing decreased.</p>
</body>
</html>
```

8. CSS Text - Word Spacing

CSS provides property to adjust the spacing between the words in a piece of text, just like letter-spacing. The property to adjust the space between words is **word-spacing**.

This property can take following values:

- **normal**: Default value and represents the normal amount of space between words.
- **length**: Any specific length {pixels (px), points (pt), ems (em), or percentages (%)}.
- **initial**: Sets the word-spacing to its default value.
- **inherit**: Allows to inherit the word-spacing value from its parent element.

Example:

```
<html>
<head>
</head>
<body>
  <h2>Word spacing</h2>
    <p style="word-spacing: normal;">Word spacing normal.</p>
    <p style="word-spacing: 15pt;">Word spacing increased.</p>
    <p style="word-spacing: -1px;">Word spacing decreased.</p>
</body>
</html>
```

9. CSS Text - With Shadow

The **text-shadow** property is used to add a shadow effect to text. It allows you to specify the color, offset, blur-radius, and spread-radius of the shadow.

By using this property, you can create various text effects like giving the text a 3D or glowing appearance, or adding emphasis and depth to the text.

The property specifies the combination of X and Y offsets, blur radius and color values. No fixed order of values is mandated. It is specified as a comma-separated list of shadow values.

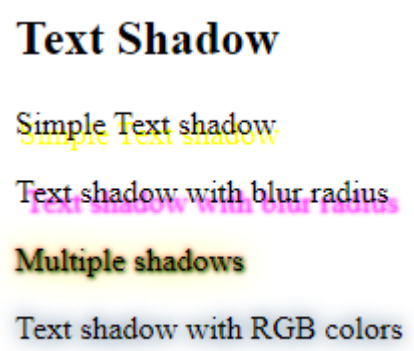
text-shadow property accepts following values:

- **<color>**:
  - Sets the color of the shadow.
  - It is optional.
  - It can be specified either before or after the offset values.
  - Any value for color can be specified, such as, name, HEX or RGB value.
- **<offset-x><offset-y>**:
  - Any length value, specifying the x and y values.
  - x value represents the shadow's horizontal distance from text.
  - y value represents the shadow's vertical distance from text.
  - If x and y values equal 0, the shadow appears behind the text.
- **<blur-radius>**
  - Any length value, specifying the value of blur-radius.
  - It is optional.
  - To make the blur look bigger, you need to provide higher value.
  - If no value is passed, it is taken as 0.

Example

```
<html>
<head>
</head>
<body>
  <h2>Text Shadow</h2>
    <p style="text-shadow: 2px 5px yellow;"> Simple Text shadow </p>
    <p style="text-shadow: 5px 5px 2px #ff00ff;">Text shadow with blur radius</p>
    <p style="text-shadow: 1px 1px 2px green, 0 0 1em yellow, 0 0 0.2em
red;">Multiple shadows</p>
    <p style="text-shadow: 0px 0px 10px rgb(26, 69, 105); ">Text shadow with RGB
colors</p>
</body>
</html>
```

Output:



10. CSS Text - Line Break

CSS provides property **line-break** that is useful in determining how to break lines in a block of text.

Following are the values that this property can have:

- **auto**: Default line break rule applied.
- **loose**: Least restrictive line break rule applied.
- **normal**: Most common line break rule applied.
- **strict**: Most stringent line break rule applied.
- **anywhere**: Allows the browser to apply line break rule anywhere, at any character.
- **initial**: Set the initial value.
- **inherit**: Inherits the value of the parent element.

Here is an example:

```
<html>
<head>
<style>
        p {
          border: 2px solid blue;
          width: 200px;
        }
        .normal {
          line-break: normal;
        }
        .loose {
          line-break: loose;
        }
        .strict {
          line-break: strict;
        }
        .auto {
          line-break: auto;
        }
        .anywhere {
          line-break: anywhere;
```

```
        }
      </style>
</head>
<body>
 <h2>Line Break</h2>
<p class="normal">Normal - CSS provides property <b>line-break</b> that is useful in determining how to break lines in a block of text.</p>
    <p class="loose">Loose - CSS provides property <b>line-break</b> that is useful in determining how to break lines in a block of text</p>
    <p class="strict">Strict - CSS provides property <b>line-break</b> that is useful in determining how to break lines in a block of text</p>
    <p class="auto">Auto - CSS provides property <b>line-break</b> that is useful in determining how to break lines in a block of text</p>
    <p class="anywhere">Anywhere - CSS provides property <b>line-break</b> that is useful in determining how to break lines in a block of text</p>
</body>
</html>
```

Output:

Line Break

Normal - CSS provides property **line-break** that is useful in determining how to break lines in a block of text.
Loose - CSS provides property **line-break** that is useful in determining how to break lines in a block of text
Strict - CSS provides property **line-break** that is useful in determining how to break lines in a block of text
Auto - CSS provides property **line-break** that is useful in determining how to break lines in a block of text
Anywhere - CSS provides property **line-break** that is useful in determining how to break lines in a block of text

CSS Text - Word Break

The **word-break** property in CSS is used to specify how words should be broken or wrapped in case they exceed the available width of an element. It determines if the browser should allow the words to break at any point or if they should be kept together.

Following are the values that this property can have:

- **normal**: Uses default line break rule.
- **break-all**: Word breaks to be applied between any two characters, in order to prevent overflow.
- **keep-all**: Word breaks not to be used for Chinese, Japanese and Korean (CJK) text; and for other languages or non-CJK text behavior is the same as for **normal**.
- **break-word**: This behaves same as **overflow-wrap: anywhere**, i.e. word break at any word is applied. But this value is deprecated.

example:

```
<html>
<head>
<style>
  p {
    border: 2px solid green;
    width: 200px;
  }
  .normal {
    word-break: normal;
  }
  .all {
    word-break: break-all;
  }
  .keep {
    word-break: keep-all;
  }
  .wordbreak {
    word-break: break-word;
  }
</style>
</head>
<body>
  <h2>Word Break</h2>
    <p class="normal">normal - CSS provides property <b>word-break</b> that is useful in determining how to break words in a block of text</p>
    <p class="all">break-all - CSS provides property <b>word-break</b> that is useful in determining how to break words in a block of text</p>
    <p class="keep">keep-all - CSS provides property <b>word-break</b> that is useful in determining how to break words in a block of text</p>
    <p class="wordbreak">break-word - CSS provides property <b>word-break</b> that is useful in determining how to break words in a block of text</p>
</body>
</html>
```

Word Break

normal - CSS provides property **word-break** that is useful in determining how to break words in a block of text

break-all - CSS provides property word-break that is useful in determining how to break words in a block of text

keep-all - CSS provides property word-break that is useful in determining how to break words in a block of text

break-word - CSS provides property word-break that is useful in determining how to break words in a block of text.

CSS Font

CSS Font property is used to control the look of texts. By the use of CSS font property you can change the text size, color, style and more. You have already studied how to make text bold or underlined. Here, you will also know how to resize your font using percentage.

These are some important font attributes:

1. **CSS Font color**: This property is used to change the color of the text. (standalone attribute)
2. **CSS Font family**: This property is used to change the face of the font.
3. **CSS Font size**: This property is used to increase or decrease the size of the font.
4. **CSS Font style**: This property is used to make the font bold, italic or oblique.
5. **CSS Font variant**: This property creates a small-caps effect.
6. **CSS Font weight**: This property is used to increase or decrease the boldness and lightness of the font.

1) CSS Font Color

CSS font color is a standalone attribute in CSS although it seems that it is a part of CSS fonts. It is used to change the color of the text.

There are three different formats to define a color:

- By a color name
- By hexadecimal value
- By RGB

In the above example, we have defined all these formats.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
   font-size: 100%;
}
h1 { color: red; }
```

```
h2 { color: #9000A1; }
p { color:rgb(0, 220, 98); }
}
</style>
</head>
<body>
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<p>This is a paragraph.</p>
</body>
</html>
```

Output:

This is heading 1

This is heading 2

This is a paragraph.

2) CSS Font Family

CSS font family can be divided in two types:

- Generic family: It includes Serif, Sans-serif, and Monospace.
- Font family: It specifies the font family name like Arial, New Times Roman etc.

Serif: Serif fonts include small lines at the end of characters. Example of serif: Times new roman, Georgia etc.

Sans-serif: A sans-serif font doesn't include the small lines at the end of characters. Example of Sans-serif: Arial, Verdana etc.


```
<!DOCTYPE html>
```

```
<html>
<head>
<style>
body {
font-size: 100%;
}
h1 { font-family: sans-serif; }
h2 { font-family: serif; }
p { font-family: monospace; }
}
</style>
</head>
<body>
<h1>This heading is shown in sans-serif.</h1>
<h2>This heading is shown in serif.</h2>
<p>This paragraph is written in monospace.</p>
</body>
</html>
```

Output:

This heading is shown in sans-serif.

This heading is shown in serif.

This paragraph is written in monospace.

3) CSS Font Size

CSS font size property is used to change the size of the font.

These are the possible values that can be used to set the font size:

Font Size Value	Description
xx-small	used to display the extremely small text size.
x-small	used to display the extra small text size.
small	used to display small text size.
medium	used to display medium text size.
large	used to display large text size.
x-large	used to display extra large text size.
xx-large	used to display extremely large text size.
smaller	used to display comparatively smaller text size.
larger	used to display comparatively larger text size.

size in pixels or %	used to set value in percentage or in pixels.

Example:

```
<html>
<head>
<title>Practice CSS font-size property</title>
</head>
<body>
<p style="font-size:xx-small;"> This font size is extremely small.</p>
<p style="font-size:x-small;"> This font size is extra small</p>
<p style="font-size:small;"> This font size is small</p>
<p style="font-size:medium;"> This font size is medium. </p>
<p style="font-size:large;"> This font size is large. </p>
<p style="font-size:x-large;"> This font size is extra large. </p>
<p style="font-size:xx-large;"> This font size is extremely large. </p>
<p style="font-size:smaller;"> This font size is smaller. </p>
<p style="font-size:larger;"> This font size is larger. </p>
<p style="font-size:200%;"> This font size is set on 200%. </p>
<p style="font-size:20px;"> This font size is 20 pixels. </p>
</body>
</html>
```

Output:

Output:

This font size is extremely small.

This font size is extra small

This font size is small

This font size is medium.

This font size is large.

This font size is extra large.

This font size is extremely large.

This font size is smaller.

This font size is larger.

This font size is set on 200%.

This font size is 20 pixels.

4) CSS Font Style

CSS Font style property defines what type of font you want to display. It may be italic, oblique, or normal.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
font-size: 100%;
}
h2 { font-style: italic; }
h3 { font-style: oblique; }
h4 { font-style: normal; }
}
</style>
</head>
<body>
<h2>This heading is shown in italic font.</h2>
<h3>This heading is shown in oblique font.</h3>
<h4>This heading is shown in normal font.</h4>
</body>
</html>
```

Output:

Output:

This heading is shown in italic font.

This heading is shown in oblique font.

This heading is shown in normal font.

5) CSS Font Variant

CSS font variant property specifies how to set font variant of an element. It may be normal and small-caps.

```
<!DOCTYPE html>
```

```
<html>
<head>
<style>
p { font-variant: small-caps; }
h3 { font-variant: normal; }
</style>
</head>
<body>
<h3>This heading is shown in normal font.</h3>
<p>This paragraph is shown in small font.</p>
</body>
</html>
```

Output:

This heading is shown in normal font.

THIS PARAGRAPH IS SHOWN IN SMALL FONT.

6) CSS Font Weight

CSS font weight property defines the weight of the font and specify that how bold a font is. The possible values of font weight may be normal, bold, bolder, lighter or number (100, 200..... upto 900).

```
<!DOCTYPE html>
<html>
<body>
<p style="font-weight:bold;">This font is bold.</p>
<p style="font-weight:bolder;">This font is bolder.</p>
<p style="font-weight:lighter;">This font is lighter.</p>
<p style="font-weight:100;">This font is 100 weight.</p>
<p style="font-weight:200;">This font is 200 weight.</p>
<p style="font-weight:300;">This font is 300 weight.</p>
<p style="font-weight:400;">This font is 400 weight.</p>
<p style="font-weight:500;">This font is 500 weight.</p>
<p style="font-weight:600;">This font is 600 weight.</p>
<p style="font-weight:700;">This font is 700 weight.</p>
<p style="font-weight:800;">This font is 800 weight.</p>
<p style="font-weight:900;">This font is 900 weight.</p>
</body>
</html>
```

Output:

This font is bold.

This font is bolder.

This font is lighter.

This font is 100 weight.

This font is 200 weight.

This font is 300 weight.

This font is 400 weight.

This font is 500 weight.

This font is 600 weight.

This font is 700 weight.

This font is 800 weight.

This font is 900 weight.

❖ CSS Icons

Icon Libraries

With W3.CSS you can use the icon library you like, such as:

- Font Awesome Icons
- Google Material Design Icons
- Bootstrap Icons

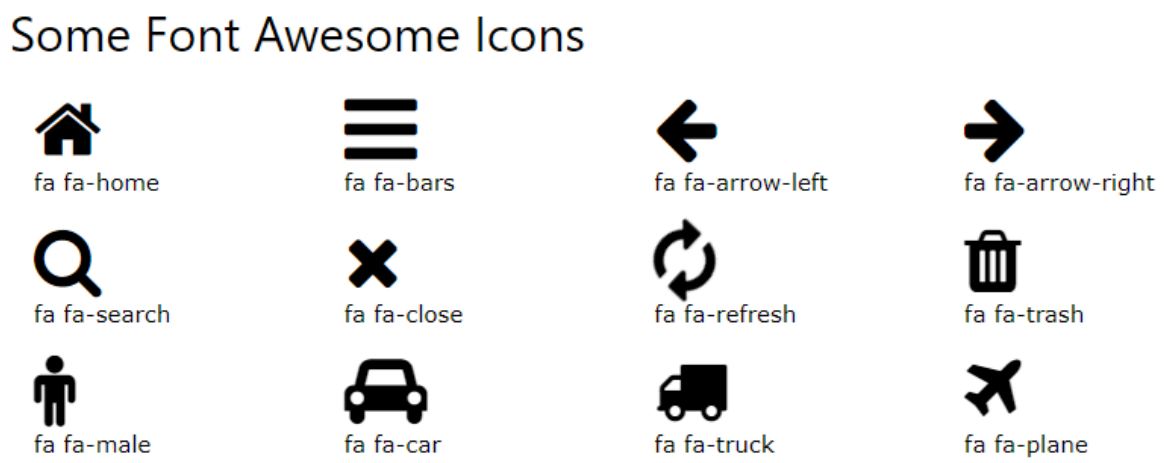
Using an Icon Library

To insert an icon:

1. Include the icon library from a CDN (Content Delivery Network) in the <head> section.
2. Add the name of the icon class to any inline HTML element.

Tip: The <i> and elements are widely used to add icons.
To control the size of the icon, change the font-size property of the icon, or use one of the **w3-*size*** classes:

- w3-tiny
- w3-small
- w3-large
- w3-xxlarge
- w3-xxxlarge
- w3-jumbo



Example

```
<!DOCTYPE html>
<html>
<title>W3.CSS</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
<body>

<i class="fa fa-home"></i>
<i class="fa fa-search"></i>
<i class="fa fa-cloud"></i>
<i class="fa fa-trash"></i>

</body>
</html>
```

Output:

❖ CSS Lists

Unordered Lists:	Ordered Lists:
o Coffee o Tea o Coca Cola ▪ Coffee ▪ Tea ▪ Coca Cola	1. Coffee 2. Tea 3. Coca Cola I. Coffee II. Tea III. Coca Cola

HTML Lists and CSS List Properties

In HTML, there are two main types of lists:

- unordered lists (<ul>) - the list items are marked with bullets
- ordered lists (<ol>) - the list items are marked with numbers or letters

The CSS list properties allow you to:

- Set different list item markers for ordered lists
- Set different list item markers for unordered lists
- Set an image as the list item marker
- Add background colors to lists and list items

Different List Item Markers

- The list-style-type property specifies the type of list item marker.
- The following example shows some of the available list item markers:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
ul.a {
  list-style-type: circle;
}

ul.b {
  list-style-type: square;
}

ol.c {
  list-style-type: upper-roman;
}

ol.d {
  list-style-type: lower-alpha;
}
</style>
</head>
<body>

<h2>The list-style-type Property</h2>

<p>Example of unordered lists:</p>
<ul class="a">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>
```

```
<ul class="b">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>

<p>Example of ordered lists:</p>
<ol class="c">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ol>

<ol class="d">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ol>

</body>
</html>
```

Output:

The list-style-type Property

Example of unordered lists:

- Coffee
- Tea
- Coca Cola

- Coffee
- Tea
- Coca Cola

Example of ordered lists:

I. Coffee
II. Tea
III. Coca Cola

a. Coffee
b. Tea
c. Coca Cola

An Image as The List Item Marker

The list-style-image property specifies an image as the list item marker:

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-image: url('sqpurple.gif');
}
</style>
</head>
<body>

<h2>The list-style-image Property</h2>

<p>The list-style-image property specifies an image as the list item marker:</p>

<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>

</body>
</html>
```

Output:

The list-style-image Property

The list-style-image property specifies an image as the list item marker:

- Coffee
- Tea
- Coca Cola

Position The List Item Markers

The `list-style-position` property specifies the position of the list-item markers (bullet points).

"list-style-position: outside;" means that the bullet points will be outside the list item. The start of each line of a list item will be aligned vertically. This is default:

- Coffee - A brewed drink prepared from roasted coffee beans...
- Tea
- Coca-cola

"list-style-position: inside;" means that the bullet points will be inside the list item. As it is part of the list item, it will be part of the text and push the text at the start:

- Coffee - A brewed drink prepared from roasted coffee beans...
- Tea
- Coca-cola

Example

```
ul.a {
  list-style-position: outside;
}

ul.b {
  list-style-position: inside;
}
```