CSS

CSS is an acronym for Cascading Style Sheet and it was created by HÃƒÂ¥kon Wium Lie in 1996 for the web developers to change the layout, colors, and fonts of their websites.

CSS is a style sheet or presentational language that is used to layout, format, and style documents that are written in HTML to make them look beautiful. CSS is generally used with HTML to change the style of web pages and user interfaces. You can use CSS to change the color, backgrounds, borders, paddings, margins, fonts, icons, position and various other properties of HTML elements in a web document.

CSS Examples

Below HTML & CSS code is editable where you can change the value of the code. You can click the icon run button to see the output of the following code. Try to change the values of width, font-size and padding etc and then run it again to see the modified result.

```
<html>

<head>

   <style>

      div{ background-color: lightgrey; width:100%;
padding:5px}


      h1{color: #40a944; text-align: center;}


      p{ font-family: verdana;font-size: 20px;}

   </style>

</head>

<body>

   <div>

      <h1>CSS Example</h1>

      <p>This is a paragraph.</p>

   </div>

</body>

</html>
```

Why to Learn CSS?

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning CSS:

- Create Stunning Web site - CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

- Become a web designer - If you want to start a career as a professional web designer, HTML and CSS designing is a must skill.

- Control web - CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

- Learn other languages - Once you understand the basic of HTML and CSS then other related technologies like javascript, php, or angular are become easier to understand.

Applications of CSS

As mentioned before, CSS is one of the most widely used style language over the web. I'm going to list few of the CSS applications here:

- CSS saves time - You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.

- Pages load faster - If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So, less code means faster download times.

- Easy maintenance - To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.

- Superior styles to HTML - CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

- Multiple Device Compatibility - Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.

- Global web standards - Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

What is CSS?
Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to transform the presentation of a Web Pages as well as many ostensibly nonweb environments.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs,variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

Where do we use CSS?
CSS is being used extensively in web and non web based applications :

- All modern websites make use of CSS to beautify their web pages.
- Embedded-device displays often use CSS to style their user interfaces.
- RSS clients also let you apply CSS to feeds and feed entries.
- Instant message clients also use CSS to format chat windows.

History of CSS

Cascading Style Sheets level 1 (CSS1) came out of W3C as a recommendation in December 1996. This version describes the CSS language as well as a simple visual formatting model for all the HTML tags.

CSS2 became a W3C recommendation in May 1998 and builds on top of CSS1. This version adds support for media-specific style sheets e.g. printers and aural devices, downloadable fonts, element positioning and tables.

CSS3 became a W3C recommendation in June 2012 and builds on older versions CSS. it has divided into documentations called as Modules and here each module having new extension features defined in CSS2.

Year	Description
1994	HÃ¥kon Wium Lie proposed the idea of CSS to allow web designers to change the layout, colors, and fonts of their websites.
1996	The first version of CSS was released while the newly established CSS Working Group moved forward with CSS2.
1998	The second version of CSS was released and work on CSS-3 started at the same time.
2011	A clarified version of CSS2 called CSS2.1, was released, which fixed the errors found in CSS 2
2012	As of June 2012, there are over fifty CSS modules published from the CSS-3 Working Group.

How CSS Works?

First of all we define some classes in the form of rules using CSS to apply some styles like fonts, colors, size etc: Under various conditions, we want certain things to happen for example if element X is a child of element Y, apply those styles. Next these classes are added to the HTML elements. The browser then takes these rules, figures out which ones apply where, and uses them to render the page.

Style Sheets Origin

The stylesheets you add to your web page aren’t the only ones the browser applies. There are different types, or origins, of stylesheets:

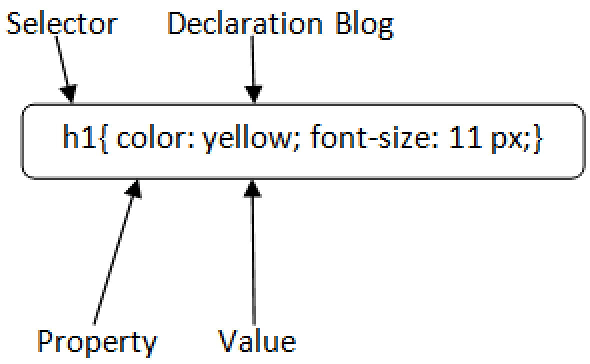
- User agent styles - This is the default style which browser applies to any web page.

- Author styles - Your stylesheets are called author styles which override user agent styles.

- User stylesheet - Some browsers let users define a user stylesheet and these are rarely used.

CSS Syntax

A CSS rule set contains a selector and a declaration block.



Selector: Selector indicates the HTML element you want to style. It could be any tag like <h1>, <title> etc.

Declaration Block: The declaration block can contain one or more declarations separated by a semicolon. For the above example, there are two declarations:

color: yellow;

font-size: 11 px;

Each declaration contains a property name and value, separated by a colon.

Property: A Property is a type of attribute of HTML element. It could be color, border etc.

Value: Values are assigned to CSS properties. In the above example, value "yellow" is assigned to color property.

Selector{Property1: value1; Property2: value2; ....;}

CSS Selector

CSS selectors are used to select the content you want to style. Selectors are the part of CSS rule set. CSS selectors select HTML elements according to its id, class, type, attribute etc.

There are several different types of selectors in CSS.

1. CSS Element Selector
2. CSS Id Selector
3. CSS Class Selector
4. CSS Universal Selector
5. CSS Group Selector

1) CSS Element Selector

The element selector selects the HTML element by name.

```
<!DOCTYPE html>
<html>
<head>
<style>
p{
    text-align: center;
    color: blue;
}
</style>
</head>
<body>
<p>This style will be applied on every paragraph.</p>
<p id="para1">Me too!</p>
<p>And me!</p>
</body>
</html>
```

2) CSS Id Selector

The id selector selects the id attribute of an HTML element to select a specific element. An id is always unique within the page so it is chosen to select a single, unique element.

It is written with the hash character (#), followed by the id of the element.

Let?s take an example with the id "para1".

```
<!DOCTYPE html>

<html>

<head>

<style>

#para1 {

    text-align: center;

    color: blue;

}

</style>

</head>

<body>

<p id="para1">Hello Every One</p>

<p>This paragraph will not be affected.</p>

</body>

</html>
```

3) CSS Class Selector

The class selector selects HTML elements with a specific class attribute. It is used with a period character. (full stop symbol) followed by the class name.

Note: A class name should not be started with a number.

Let's take an example with a class "center".

```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
    text-align: center;
    color: blue;
}
</style>
</head>
<body>
<h1 class="center">This heading is blue and center-aligned.</h1>
<p class="center">This paragraph is blue and center-aligned.</p>
</body>
</html>
```

CSS Class Selector for specific element

If you want to specify that only one specific HTML element should be affected then you should use the element name with class selector.

Let's see an example.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.center {
    text-align: center;
    color: blue;
}
</style>
</head>
<body>
<h1 class="center">This heading is not affected</h1>
<p class="center">This paragraph is blue and center-aligned.</p>
</body>
</html>
```

4) CSS Universal Selector

The universal selector is used as a wildcard character. It selects all the elements on the pages.

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
   color: green;
   font-size: 20px;
}
</style>
</head>
<body>
<h2>This is heading</h2>
<p>This style will be applied on every paragraph.</p>
<p id="para1">Me too!</p>
<p>And me!</p>
</body>
</html>
```

5) CSS Group Selector

The grouping selector is used to select all the elements with the same style definitions.

Grouping selector is used to minimize the code. Commas are used to separate each selector in grouping.

Let's see the CSS code without group selector.

```
h1 {
    text-align: center;
    color: blue;
}
h2 {
    text-align: center;
    color: blue;
}
p {
    text-align: center;
    color: blue;
}
```

As you can see, you need to define CSS properties for all the elements. It can be grouped in following ways:

```
h1,h2,p {
    text-align: center;
    color: blue;
}
```

Let's see the full example of CSS group selector.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1, h2, p {
    text-align: center;
    color: blue;
}
</style>
</head>
<body>
<h1>Hello Every One</h1>
<h2>Hello Google.com (In smaller font)</h2>
<p>This is a paragraph.</p>
</body>
</html>
```

How to add CSS

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.

1. Inline CSS
2. Internal CSS
3. External CSS

1) Inline CSS

Inline CSS is used to apply CSS on a single line or element.

For example:

```
<p style="color:blue">Hello CSS</p>
```

2) Internal CSS

Internal CSS is used to apply CSS on a single document or page. It can affect all the elements of the page. It is written inside the style tag within head section of html.

For example:

```
<style>

p{color:blue}

</style>
```

3) External CSS

External CSS is used to apply CSS on multiple pages or all pages. Here, we write all the CSS code in a CSS file. Its extension must be .css for example style.css.

For example:

```
p{color:blue}
```

You need to link this style.css file to your html pages like this:

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

The link tag must be used inside head section of html.

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:

```
<!DOCTYPE html>

<html>

<body>

<h1 style="color:red;margin-left:40px;">Inline CSS is
applied on this heading.</h1>

<p>This paragraph is not affected.</p>

</body>

</html>
```

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.

Let's take an example of a style sheet file named "mystyle.css".

```
File: mystyle.css
```

```
body {

    background-color: lightblue;

}

h1 {

    color: navy;

    margin-left: 20px;

}
```

Note: You should not use a space between the property value and the unit. For example: It should be margin-left:20px not margin-left:20 px.

CSS Comments

CSS comments are generally written to explain your code. It is very helpful for the users who reads your code so that they can easily understand the code.

Comments are ignored by browsers.

Comments are single or multiple lines statement and written within /*............*/ .

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    color: blue;
    /* This is a single-line comment */
    text-align: center;
}
/* This is
a multi-line
comment */
</style>
</head>
<body>
<p>Hello Google.com</p>
<p>This statement is styled with CSS.</p>
<p>CSS comments are ignored by the browsers and not shown in the output.</p>
</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

1) 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. This is an example of CSS background-color.</p>
</body>
</html>
```

2) 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 Google.com</h1>
</body>
</html>
```

Note: The background image should be chosen according to text color. The bad combination of text and background image may be a cause of poor designed and not readable webpage.

3) 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 Google.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 Google.com</h1>
</body>
</html>
```

4) 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.

```
<!DOCTYPE html>
<html>
<head>
<style>
body  {
background: white url('bbb.gif');
background-repeat: no-repeat;
background-attachment: fixed;
margin-left:200px;
}
</style>
</head>
<body>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
```

```
<p>This is a fixed background-image. Scroll down the page.</p>

<p>This is a fixed background-image. Scroll down the page.</p>

<p>This is a fixed background-image. Scroll down the page.</p>

<p>This is a fixed background-image. Scroll down the page.</p>

<p>This is a fixed background-image. Scroll down the page.</p>

<p>This is a fixed background-image. Scroll down the page.</p>

<p>This is a fixed background-image. Scroll down the page.</p>

<p>This is a fixed background-image. Scroll down the page.</p>

<p>This is a fixed background-image. Scroll down the page.</p>

<p>This is a fixed background-image. Scroll down the page.</p>

<p>This is a fixed background-image. Scroll down the page.</p>

<p>This is a fixed background-image. Scroll down the page.</p>

<p>If you do not see any scrollbars, Resize the browser window.</p>

</body>

</html>
```

5) 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:

1. center
2. top
3. bottom
4. left
5. right

```
<!DOCTYPE html>

<html>

<head>

<style>

body  {

background: white url('good-morning.jpg');

background-repeat: no-repeat;

background-attachment: fixed;

background-position: center;

}

</style>

</head>

<body>

<p>This is a fixed background-image. Scroll down the page.</p>

<p>This is a fixed background-image. Scroll down the page.</p>
```

```
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
```

```
<p>If you do not see any scrollbars, Resize the browser window.</p>

</body>

</html>
```

CSS Background-color

This property is used to set the background color of an element. The background of an element covers the total size, including the padding and border and excluding margin. It can be applied to all HTML elements.

Syntax

```
element {
    background-color:
color_name|transparent|initial|inherit;
}
```

Let's discuss the possible values of this property.

- **color_name:** It is used for defining the background color value or the color codes. It can be given by using the color name, hexadecimal value, or rgb() value.
- **transparent:** It is the default value of this property, which is used to specify the transparent background-color.
- **initial:** It is not used to set the background color. It sets the default value.
- **Inherit:** It is used to inherit the background-color from its parent.

Let's see an illustration of this property.

Example

```
<!DOCTYPE html>
<html>
    <head>
        <title>background-color property</title>
        <style>
            body {
                text-align:center;
                background-color: lightblue;
            }
         h1{
            color: blue;
         }
        </style>
    </head>
    <body>
       <h1>Hello World.</h1>
        <h1>Welcome to the Google.com</h1>
        </body>
</html>
```

CSS background-attachment property

The background-attachment property is used to specify that the background image is fixed or scroll with the rest of the page in the browser window.

This property has three values scroll, fixed, and local. Its default value is scroll, which causes the element to not scroll with its content. The local value of this property causes the element to scroll with the content. If we set the value to fixed, the background image will not move during scrolling in the browser.

This CSS property can support multiple background images. We can specify a different value of the background-attachment property for each background-image, separated by commas. Every image will match with the corresponding value of this property.

Syntax

```
background-attachment: scroll | fixed | local | initial
| inherit;
```

The values of this property are defined as follows.

Property Values

scroll: It is the default value that prevents the element from scrolling with the contents, but scrolls with the page.

fixed: Using this value, the background image doesn't move with the element, even the element has a scrolling mechanism. It causes the image to be locked in one place, even the rest of the document scrolls.

local: Using this value, if the element has a scrolling mechanism, the background image scrolls with the content of the element.

initial: It sets the property to its default value.

inherit: It inherits the property from its parent element.

Let's understand this property by using some illustrations.

Example

In this example, we are using the scroll value of the background-attachment property, which is the default behaviour. So when the page is scrolled, the background scrolls with it.

```
<!DOCTYPE html>
<html>
<head>
<title>
background-attachment property
</title>

<style>
#example {
background-image:  url("lion.png");
font-size: 35px;
border: 4px solid red;
color: white;
background-position: center;
background-color: green;
background-repeat: no-repeat;
background-attachment: scroll;
}
```

```
</style>

</head>


<body>

<h1> background-attachment: scroll;</h1>

<p> If there is no scrollbar on your screen, then try
to resize the browser's window to see the effect.
</p>

<div id="example">

<p>

Hi, Welcome to the Google.com. This site is developed
so that students may learn computer science related
technologies easily. The Google.com is always
providing an easy and in-depth tutorial on various
technologies. No one is perfect in this world, and
nothing is eternally best. But we can try to be
better.

</p>

</div>


</body>

</html>
```

Example - Using fixed value

In this example, we are using the fixed value of the background-attachment property. This value fixed the background image, and the image will not move even the rest of the document scrolls.

```
<!DOCTYPE html>
<html>
<head>
<title>
background-attachment property
</title>

<style>
#example {
background-image:  url("lion.png");
font-size: 35px;
border: 4px solid red;
color: white;
background-position: center;
background-color: green;
background-repeat: no-repeat;
background-attachment: fixed;
}

</style>
</head>

<body>
```

```html
<h1> background-attachment: fixed;</h1>

<p> If there is no scrollbar on your screen, then try
to resize the browser's window to see the effect.
</p>

<div id="example">

<p>

Hi, Welcome to the Google.com. This site is developed
so that students may learn computer science related
technologies easily. The Google.com is always
providing an easy and in-depth tutorial on various
technologies. No one is perfect in this world, and
nothing is eternally best. But we can try to be
better.

</p>

</div>


</body>

</html>
```

Example - Using local value

In this example, we are using the local value of the background-attachment property. Here, the background-image will scroll with the scrolling of the element's content.

```
<!DOCTYPE html>
<html>
<head>
<title>
background-attachment property
</title>

<style>
#example {
background-image:  url("lion.png");
font-size: 35px;
border: 4px solid red;
color: white;
background-position: center;
background-color: green;
background-repeat: no-repeat;
background-attachment: local;
}

</style>
</head>

<body>
```

```html
<h1> background-attachment: local;</h1>

<p> If there is no scrollbar on your screen, then try
to resize the browser's window to see the effect.
</p>

<div id="example">

<p>

Hi, Welcome to the Google.com. This site is developed
so that students may learn computer science related
technologies easily. The Google.com is always
providing an easy and in-depth tutorial on various
technologies. No one is perfect in this world, and
nothing is eternally best. But we can try to be
better.

</p>

</div>


</body>

</html>
```

Example

Here, there are two background images on which we are applying the background-attachment property. The attachment for the first image is set to fixed, whereas the attachment for the second image is set to scroll.

```
<!DOCTYPE html>
<html>
<head>
<title>
background-attachment property
</title>

<style>
#example {
background-image:  url("jtp.png"), url("forest.jpg");
height: 500px;
border: 4px solid red;
background-position: center;
background-repeat: no-repeat;
background-attachment: fixed, scroll;
}

</style>
</head>

<body>
<h1> background-attachment: scroll;</h1>
```

```
<p> If there is no scrollbar on your screen, then try
to resize the browser's window to see the effect.
</p>

<div id="example">

</div>

</body>

</html>
```

CSS background-size property

The background-size CSS property is used to set the size of a background image of an element. The background image can be stretched or constrained to fit into the existing space. It allows us to control the scaling of the background image.

This property can be defined using length, percentage, or keyword values. It has two possible keyword values that are contain and cover. Its single-value syntax defines the width of the image (in this case, the height sets to auto), whereas the double values define the value of both height and width in which the first value sets the width and second sets the height.

If an element has multiple background images, we can define the comma-separated values to define the different sizes of each one.

The cover value of the background-size property is used to cover the entire background area of the element. In contrast, the contain value of this property scales the image as much as possible without clipping the image.

Syntax

```
background-size: auto | length | cover | contain |
initial | inherit;
```

The values of this property are defined as follows.

Property Values

auto: This is the default value, which displays the background image in its original size.

length: It is used to set the width and height of the background image. This value stretches the image in the corresponding dimension of the given length. Its single value specifies the width of the image, and the height sets to auto. If two values are given, the first value sets the width, and the second value sets the height. It does not allow negative values.

percentage: This value defines the width and height of the background image to the percentage (%) of the background positioning area. Negative values are not allowed.

cover: This value is used to resize the background image to cover the entire container. Sometimes, it crops the little bit off one of the edges or stretches the image. It resizes the image to ensure the element is completely covered.

contain: Without stretching or cropping, it resizes the background image to ensure the image is completely visible.

initial: It sets the property to its default value.

inherit: It inherits the property from its parent element.

Let's understand this CSS property by using some illustrations.

Example

In this example, there are three div elements with a width of 300px and a height of 200px. Every div element has a background-image on which we are applying the background-size property.

Here we are using the length and percentage values to set the background-size of div element. The background-size of first div element set to auto, second div element is set to 150px 150px, and the background-size of third div element is set to 30%.

```
<!DOCTYPE html>

<html>

<head>
```

```
<title>
background-size property
</title>
<style>
div {
width: 300px;
height: 200px;
border: 2px solid red;
}
#div1{
background-image: url('lion.png');
background-size: auto;
}
#div2{
background-image: url('lion.png');
background-size: 150px 150px;
}
#div3{
background-image: url('lion.png');
background-size: 30%;
}
</style>
</head>

<body>
<h2> background-size: auto; </h2>
<div id = "div1"></div>
<h2> background-size: 150px 150px; </h2>
```

```
<div id = "div2"></div>
<h2> background-size: 30%; </h2>
<div id = "div3"></div>
</body>
</html>
```

Now, in the next example, we are using the cover, contain, and initial values of the background-size property.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>
background-size property
</title>
<style>
div {
width: 300px;
height: 250px;
border: 2px solid red;
background-repeat: no-repeat;
}
#div1{
background-image: url('lion.png');
background-size: contain;
}
#div2{
```

```
background-image: url('lion.png');
background-size: cover;
}
#div3{
background-image: url('lion.png');
background-size: initial;
}
</style>
</head>

<body>
<h2> background-size: contain; </h2>
<div id = "div1"></div>
<h2> background-size: cover; </h2>
<div id = "div2"></div>
<h2> background-size: initial; </h2>
<div id = "div3"></div>
</body>
</html>
```

Example - Combining multiple images

We can also combine the values of this property and can apply them to multiple images. It can be done by comma-separated syntax.

In this example, there are three div elements, each having two background-images. Now, we are applying the background-size property on both images.

```
<!DOCTYPE html>
<html>
<head>
<title>
background-size property
</title>
<style>
div {
width: 250px;
height: 250px;
border: 2px solid red;
background-repeat: no-repeat;
background-position: center;
}
#div1{
background-image: url('lion.png'), url('forest.jpg');
background-size: 300px 150px, cover;
}
#div2{
background-image: url('lion.png'), url('forest.jpg');
background-size: 200px 150px, 300px 200px;
}
#div3{
background-image: url('lion.png'), url('forest.jpg');
background-size: 150px 175px, contain;
}
</style>
</head>
```

```
<body>
<h2> background-size: 300px 150px, cover; </h2>
<div id = "div1"></div>
<h2> background-size: 200px 150px, 300px 200px; </h2>
<div id = "div2"></div>
<h2> background-size: 150px 175px, contain; </h2>
<div id = "div3"></div>

</body>
</html>
```

CSS Position

The CSS position property is used to set position for an element. it is also used to place an element behind another and also useful for scripted animation effect.

You can position an element using the top, bottom, left and right properties. These properties can be used only after position property is set first. A position element's computed position property is relative, absolute, fixed or sticky.

Let's have a look at following CSS positioning:

- CSS Static Positioning
- CSS Fixed Positioning
- CSS Relative Positioning
- CSS Absolute Positioning

1) CSS Static Positioning

This is a by default position for HTML elements. It always positions an element according to the normal flow of the page. It is not affected by the top, bottom, left and right properties.

2) CSS Fixed Positioning

The fixed positioning property helps to put the text fixed on the browser. This fixed test is positioned relative to the browser window, and doesn't move even you scroll the window.

```
<!DOCTYPE html>

<html>

<head>

<style>

p.pos_fixed {

    position: fixed;

    top: 50px;
    right: 5px;
    color: blue;
}
</style>
</head>
<body>

<p>Some text...</p><p>Some text...</p><p>Some
text...</p><p>........</p><p>.... ...</p
><p>........</p><p>........</p><p>........</p><p>....
....</p>
<p>........
</p><p>........</p><p>........</p><p>........</p><p>.
.......</p>
<p>........</p><p>........</p><p>Some
text...</p><p>Some text...</p><p>Some text...</p>
<p class="pos_fixed">This is the fix positioned
text.</p>
</body>
</html>
```

3) CSS Relative Positioning

The relative positioning property is used to set the element relative to its normal position.

```
<!DOCTYPE html>
<html>
<head>
<style>
h2.pos_left {
    position: relative;
    left: -30px;
}
h2.pos_right {
    position: relative;
    left: 30px;
}
</style>
</head>
<body>
<h2>This is a heading with no position</h2>
<h2 class="pos_left">This heading is positioned left
according to its normal position</h2>
<h2 class="pos_right">This heading is positioned
right according to its normal position</h2>
<p>The style "left:-30px" subtracts 30 pixels from
the element's original left position.</p>
<p>The style "left:30px" adds 30 pixels to the
element's original left position.</p>
</body>
</html>
```

4) CSS Absolute Positioning

The absolute positioning is used to position an element relative to the first parent element that has a position other than static. If no such element is found, the containing block is HTML.

With the absolute positioning, you can place an element anywhere on a page.

```
<!DOCTYPE html>

<html>

<head>

<style>

h2 {

    position: absolute;

    left: 150px;

    top: 250px;

}

</style>

</head>

<body>

<h2>This heading has an absolute position</h2>

<p> The heading below is placed 150px from the left
and 250px from the top of the page.</p>

</body>

</html>
```

All CSS Position Properties

No.	property	description	values
1)	bottom	It is used to set the bottom margin edge for a positioned box.	auto, length, %, inherit
2)	clip	It is used to clip an absolutely positioned element.	shape, auto, inherit
3)	cursor	It is used to specify the type of cursors to be displayed.	url, auto, crosshair, default, pointer, move, e-resize, ne-resize, nw-resize, n-resize, se-resize, sw-resize, s-resize, w-resize, text, wait, help
4)	left	It sets a left margin edge for a positioned box.	auto, length, %, inherit
5)	overflow	This property is used to define what happens if content overflow an element's box.	auto, hidden, scroll, visible, inherit
6)	position	It is used to specify the type of positioning for an element.	absolute, fixed, relative, static, inherit
7)	right	It is used to set a right margin edge for a positioned box.	auto, length, %, inherit
8)	top	It is used to set a top margin edge for a positioned box.	auto, length, %, inherit
9)	z-index	It is used to set stack order of an element.	number, auto, inherit

CSS text-indent

This CSS property sets the indentation of the first line in a block of text. It specifies the amount of horizontal space that puts before the lines of text.

It allows the negative values, and if any negative value is defined, then the indentation of the first line will be towards left.

Syntax

text-indent: length | inherit | initial;

This property has the value length, but here, we will discuss its experimental values.

Values

length: This value sets the fix indentation with the units cm, pt, em, px, and others. Its default value is 0. It allows negative values. The indentation of the first line is on the left when its value is negative.

percentage: It specifies the amount in space in the percentage of the width of the containing block.

initial: It sets the property to its default value.

This CSS property has two experimental values, which are discussed as follows. These two following values are not supported in browsers.

hanging: It is unofficial and experimental value. It inverts the indented lines. It indents each line except the first. Usually, bibliographies are written with the hanging indents, in which the first line is with the left-hand margin, and the rest of the content is indented.

each-line: It is also an experimental value. It affects every line, including the first line after a forced line break (using <br>).

Example

In this example, we are using the text-indent property with the length values in px, em, and cm. We are also applying the text-align: justify; property in order to see the better results.

```
<!DOCTYPE html>

<html>

    <head>

        <title>

            CSS text-indent Property

        </title>

          <style>

        div{

        font-size: 20px;

        width: 500px;

        height:200px;

        text-align: justify;

        }

            .jtpper {

                text-indent: 65%;

            }


        </style>

    </head>


    <body>
    <center>
        <h1>Example of text-indent Property</h1>

        <h2>text-indent: 65%;</h2>
        <div class = "jtpper">
            Hi, Welcome to the Google.com. This site
is developed so that students may learn computer
science related technologies easily. The Google.com
is always providing an easy and in-depth tutorial on
various technologies. No one is perfect in this
world, and nothing is eternally best. But we can try
to be better.
        </div>

</center>

    </body>
</html>
```

CSS text-stroke

This CSS property adds a stroke to the text and also provides decoration options for them. It defines the color and width of strokes for text characters.

This CSS property is the shorthand of the following two properties:

text-stroke-width: It describes the thickness of the stroke effect and takes the unit value.

text-stroke-color: It takes the value of a color.

The text-stroke can only be used with the -webkit- prefix.

Example

```
<!DOCTYPE html>
<html>

<head>
    <title>
        CSS text-stroke property
    </title>
          <style>
        body{
        text-align: center;
        }
        .jtp {
            color: white;
            font-size: 50px;
            -webkit-text-stroke-width: 2px;

            -webkit-text-stroke-color: red;

        }

    </style>

</head>


<body>

    <h1 class= "jtp">Welcome to the Google.com</h1>

    <h2 class= "jtp" style= "-webkit-text-stroke-
color: blue;">This is an example of CSS text-stroke
property</h2>

</body>


</html>
```

CSS Text Effects

We can apply different effects on the text used within an HTML document. Some properties can be used for adding the effects on text.

Using CSS, we can style the web documents and affects the text. The properties of the text effect help us to make the text attractive and clear. There are some text effect properties in CSS that are listed below:

- word-break
- text-overflow
- word-wrap
- writing-mode

Let's discuss the above CSS properties along with illustrations.

word-break

It specifies how words should break at the end of the line. It defines the line break rules.

Syntax

```
word-break: normal |keep-all |  break-all | inherit ;
```

The default value of this property is normal. So, this value is automatically used when we don't specify any value.

Values

keep-all: It breaks the word in the default style.

break-all: It inserts the word break between the characters in order to prevent the word overflow.

Example

```
<!DOCTYPE html>
<html>
    <head>
        <title>word-break: break-all</title>
        <style>
            .jtp{
                width: 150px;
                border: 2px solid black;
                word-break: break-all;
                text-align: left;
                font-size: 25px;
    color: blue;
            }
            .jtp1{
                width: 156px;
                border: 2px solid black;
                word-break: keep-all;
                text-align: left;
                font-size: 25px;
    color: blue;
            }
        </style>
    </head>
      <center>
    <body>
        <h2>word-break: break-all;</h2>
        <p class="jtp">
           Welcome to the Google.com
        </p>
        <h2>word-break: keep-all;</h2>
        <p class="jtp1">
            Welcome to the Google.com
        </p>
</center>
    </body>
</html>
```

word-wrap
CSS word-wrap property is used to break the long words and wrap onto the next line. This property is used to prevent overflow when an unbreakable string is too long to fit in the containing box.

Syntax

```
word-wrap: normal| break-word| inherit ;
```

Values

normal: This property is used to break words only at allowed break points.

break-word: It is used to break unbreakable words.

initial: It is used to set this property to its default value.

inherit: It inherits this property from its parent element.

Example

```
<!DOCTYPE html>

<html>

<head>

<style>

.test {

    width: 200px;

    background-color: lightblue;

    border: 2px solid black;

    padding:10px;

    font-size: 20px;


}

.test1 {

    width: 11em;

    background-color: lightblue;

    border: 2px solid black;

    padding:10px;

    word-wrap: break-word;

    font-size: 20px;

}

</style>

</head>

<body>

<center>

<h1> Without Using word-wrap </h1>

<p class="test"> In this paragraph, there is a very
long word:
```

```
iamsooooooooooooooooooooooooooooolongggggggggggggg
g. </p>

<h1> Using word-wrap: break-word; </h1>

<p class="test1"> In this paragraph, there is a very
long word:

iamsooooooooooooooooooooooooooooolongggggggggggggg
g. The long word will break and wrap to the next
line. </p>

 </center>

</body>

</html>
```

text-overflow
It specifies the representation of overflowed text, which is not visible to the user. It signals the user about the content that is not visible. This property helps us to decide whether the text should be clipped or show some dots (ellipsis).

This property does not work on its own. We have to use white-space: nowrap; and overflow: hidden; with this property.

Syntax

```
text-overflow: clip | ellipsis;
```

Property Values

clip: It is the default value that clips the overflowed text.

ellipsis: This value displays an ellipsis (…) or three dots to show the clipped text. It is displayed within the area, decreasing the amount of text.

Example

```
<!DOCTYPE html>
<html>
    <head>
        <style>
            .jtp{
                white-space: nowrap;
        height: 30px;
                width: 250px;
                overflow: hidden;
                border: 2px solid black;
                font-size: 25px;
                text-overflow: clip;
            }
     .jtp1 {
                white-space: nowrap;
      height: 30px;
                width: 250px;
                overflow: hidden;
                border: 2px solid black;
                font-size: 25px;
                text-overflow: ellipsis;
            }

    h2{
    color: blue;
    }
            div:hover {
                overflow: visible;
            }
    p{
    font-size: 25px;
    font-weight: bold;
    color: red;
    }
        </style>
    </head>
    <center>
    <body>
<p> Hover over the bordered text to see the full content. </p>

        <h2>
            text-overflow: clip;
        </h2>

        <div class="jtp">
            Welcome to the Google.com
        </div>
        <h2>
            text-overflow: ellipsis;
        </h2>

        <div class="jtp1">
            Welcome to the Google.com
        </div>
        </center>
    </body>
</html>
```

writing-mode
It specifies whether the text will be written in the horizontal or vertical direction. If the writing direction is vertical, then it can be from left to right (vertical-lr) or from right to left (vertical-rl).

Syntax
writing-mode: `horizontal-tb | vertical-lr | vertical-rl | inherit ;`

Property values
horizontal-tb: It is the default value of this property in which the text is in the horizontal direction and read from left to right and top to bottom.

vertical-rl: It displays the text in the vertical direction, and the text is read from right to left and top to bottom.

vertical-lr: It is similar to vertical-rl, but the text is read from left to right.

Example

```
<!DOCTYPE html>
<html>
 <head>
  <style>
    h2 {
      border: 2px solid black;
      width: 300px;
      height: 100px;
    }
    #tb {
      writing-mode: horizontal-tb;
    }

    #lr {
      writing-mode: vertical-lr;
    }

    #rl {
       writing-mode: vertical-rl;
    }
  </style>
 </head>
 <center>
 <body>
   <h1> Example of CSS writing-mode property </h1>
   <h2 id="tb"> writing-mode: horizontal-tb; </h2>
   <h2 id="lr" style= "height: 300px;"> writing-mode:
vertical-lr; </h2><br>
   <h2 id="rl" style= "height: 300px;"> writing-mode:
vertical-rl; </h2>
   </center>
 </body>
</html>
```

CSS text-align

This CSS property is used to set the horizontal alignment of a table-cell box or the block element. It is similar to the vertical-align property but in the horizontal direction.

Syntax

```
text-align: justify | center | right | left | initial
| inherit;
```

Possible values

justify: It is generally used in newspapers and magazines. It stretches the element's content in order to display the equal width of every line.

center: It centers the inline text.

right: It is used to align the text to the right.

left: It is used to align the text to the left.

Let's see an example that will demonstrate the text-align property.

Example

```
<html>
   <head>
   </head>
<style>
h2{
color: blue;
}
</style>
```

```
   <body>
   <h1>Example of text-align proeprty</h1>

   <h2 style = "text-align: center;">
         text-align: center;
      </h2>

      <h2 style = "text-align: right;">
         text-align: right;
      </h2>

      <h2 style = "text-align: left;">
          text-align: left;
      </h2>
      <h2 style = "text-align: justify;">
          text-align: justify; To see its effect, it
should be applied on large paragraph.
      </h2>

      </body>
</html>
```

CSS text-decoration

It is a CSS property that decorates the content of the text. It adds lines under, above, and through the text. It sets the appearance of decorative lines on text. This CSS property decorates the text with several kinds of lines. This is shorthand for text-decoration-line, text-decoration-color, and text-decoration-style.

The syntax of this CSS property is given as follows-

Syntax

```
text-decoration: text-decoration-line text-decoration-color text-decoration-style|initial|inherit;
```

Let's discuss its property values along with an example.

text-decoration-line

It sets the kind of text-decoration like overline, underline, or line-through. It can be used to add a combination of lines to the selected text.

Example

In this example, we are going to use the values underline, overline, and line-through. We will also see how to use these values simultaneously.

```
<!DOCTYPE html>

<html>

<head>

    <title>text-decoration</title>

    <style>

        h1 {

            color: red;

        }

        h2{
```

```
        color: blue;
        }
        body {
            text-align: center;
        }
         p{
          font-size: 30px;
          }
        #p1 {
            text-decoration: underline;
        }
        #p2 {
            text-decoration: line-through;
        }
        #p3 {
            text-decoration: overline;
        }
        #p4 {
            text-decoration: overline underline line-
through;
        }
    </style>
</head>

<body>
    <h1>Welcome to the Google.com</h1>
    <h2>text-decoration: text-decoration-line;</h2>
    <div>
```

```
        <p id="p1">This is underline</p>

        <p id="p2">This is line-through</p>

        <p id="p3">This is overline</p>

        <p id="p4">This is the combination of
lines</p>

    </div>

</body>

</html>
```

text-decoration-style

This property is used to set the style of the line. Its values are solid, dotted, wavy, double, and dashed.

The following example explains this property more clearly.

Example

```
<!DOCTYPE html>

<html>

<head>

    <title>text-decoration</title>

    <style>

        h1 {

            color: red;

        }

        h2{

        color: blue;

        }

        body {

            text-align: center;
        }
         p{
          font-size: 30px;
          }
        #p1 {
            text-decoration: underline double;
        }
        #p2 {
            text-decoration: line-through dashed;
        }
        #p3 {
            text-decoration: dotted overline;
        }
        #p4 {
            text-decoration: lightblue wavy overline underline line-through;
            color:red;
        }
    </style>
</head>

<body>
    <h1>Welcome to the Google.com</h1>
    <h2>text-decoration: text-decoration-line text-decoration-style;</h2>
    <div>
        <p id="p1">This is double underline</p>
        <p id="p2">This is dashed line-through</p>
        <p id="p3">This is dotted overline</p>
        <p id="p4">This is the wavy combination of
lines</p>
    </div>
</body>

</html>
```

text-decoration-color
This property is used to provide color to the decoration. Its value is any color in a valid format.

Example

```
<!DOCTYPE html>
<html>
<head>
    <title>text-decoration</title>
    <style>
        h1 {
            color: red;
        }
        h2{
        color: blue;
        }
        body {
            text-align: center;
        }
         p{
          font-size: 30px;
          }
        #p1 {
            text-decoration: underline double cyan;
        }
        #p2 {
            text-decoration: line-through dashed
green;
        }
        #p3 {
            text-decoration: dotted overline blue;
        }
        #p4 {
            text-decoration: lightblue wavy overline
underline line-through;
            color:red;
        }
    </style>
</head>

<body>
    <h1>Welcome to the Google.com</h1>
    <h2>text-decoration: text-decoration-line text-
decoration-style;</h2>
    <div>
        <p id="p1">This is double underline</p>
        <p id="p2">This is dashed line-through</p>
        <p id="p3">This is dotted overline</p>
        <p id="p4">This is the wavy combination of
lines</p>
```

```
    </div>
</body>

</html>
```

CSS text-overflow property

This property specifies the representation of overflowed text, which is not visible to the user. It signals the user about the content that is not visible. This property helps us to decide whether the text should be clipped, show some dots (ellipsis), or display a custom string.

This property does not work on its own. We have to use white-space: nowrap; and overflow: hidden; with this property

Syntax

```
text-overflow: clip | ellipsis | string | initial |
inherit;
```

Property Values

clip: It is the default value that clips the overflowed text. It truncates the text at the limit of the content area, so that it can truncate the text in the middle of the character.

ellipsis: This value displays an ellipsis (?) or three dots to show the clipped text. It is displayed within the area, decreasing the amount of text.

string: It is used to represent the clipped text to the user using the string of the programmer's choice. It works only in the Firefox browser.

initial: It sets the property to its default value.

inherit: It inherits the property from its parent element.

Example

```
<!DOCTYPE html>

<html>
```

```
<head>
<style>
div{
white-space: nowrap;
height: 30px;
width: 250px;
overflow: hidden;
border: 2px solid black;
font-size: 25px;

}
.jtp{
text-overflow: clip;
}
.jtp1 {
text-overflow: ellipsis;
}

h2{
color: blue;
}
div:hover {
overflow: visible;
}
p{
font-size: 25px;
font-weight: bold;
color: red;
}
</style>
</head>
<center>
<body>
<p> Hover over the bordered text to see the full
content. </p>

<h2>
text-overflow: clip;
</h2>

<div class="jtp">
Welcome to the Google.com
</div>
<h2>
text-overflow: ellipsis;
</h2>

<div class="jtp1">
Welcome to the Google.com
</div>
</center>
</body>
</html>
```

Example

In this example, we are using the ellipsis and inherit values of the text-overflow property. There is a div element on which we are applying the text-overflow: ellipsis; and inside the div, there is a paragraph element on which we are applying the text-overflow: inherit; property.

We can see the full content by hovering over the elements. When we hover the paragraph element's content, the content of the div element will be automatically visible because the paragraph element is the child of the div element.

```
<html>


<head>
<title>
CSS text-overflow Property
</title>
<style>
div {
width: 250px;
font-size: 20px;
white-space: nowrap;
border: 2px solid red;
overflow: hidden;
text-overflow: ellipsis;
}
h1, h4{
color: red;
}
p{
```

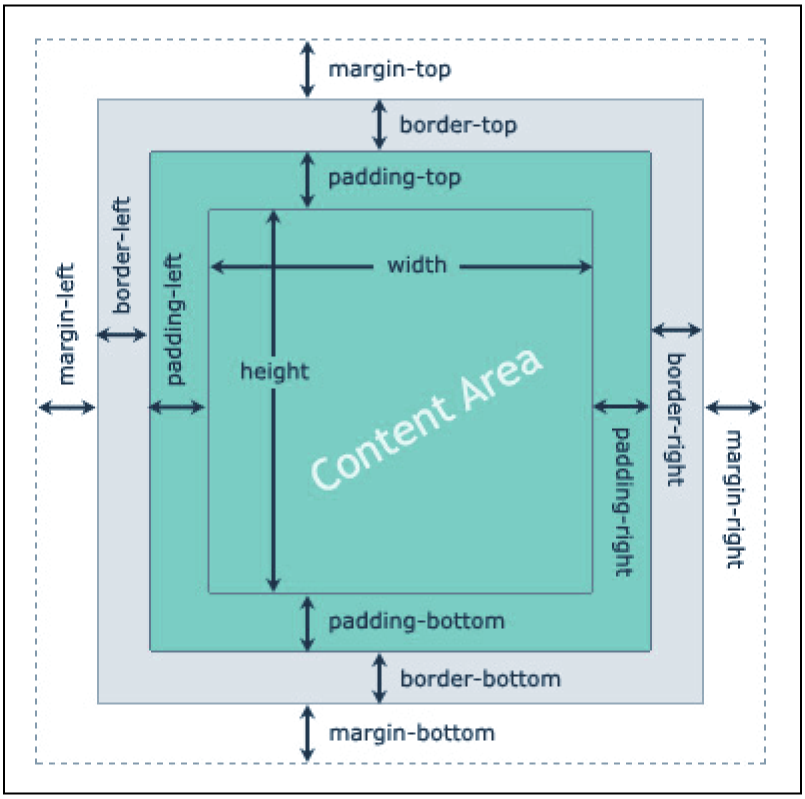
```
white-space: nowrap;

overflow: hidden;

text-overflow: inherit;

}

div:hover{

overflow: visible;

}

p:hover{

overflow: visible;

}

</style>

</head>


<body>

<h1> Hover over the text to see the full content
</h1>

<div>

<h4> text-overflow: ellipsis; </h4>

Welcome to the Google.com

<h4> text-overflow: inherit; </h4>

<p>

This paragraph inherited the value from its parent
div element.

</p>

</div>

</body>


</html>
```

CSS Box Model

The components that can be depicted on the web page consist of one or more than one rectangular box.

A CSS box model is a compartment that includes numerous assets, such as edge, border, padding and material. It is used to develop the design and structure of a web page. It can be used as a set of tools to personalize the layout of different components. According to the CSS box model, the web browser supplies each element as a square prism.

The following diagram illustrates how the CSS properties of width, height, padding, border and margin dictate that how much space an attribute will occupy on a web page.



The CSS box model contains the different properties in CSS. These are listed below.

- Border
- Margin
- Padding
- Content

Now, we are going to determine the properties one by one in detail.

Border Field

It is a region between the padding-box and the margin. Its proportions are determined by the width and height of the boundary.

Margin Field

This segment consists of the area between the boundary and the edge of the border.

The proportion of the margin region is equal to the margin-box width and height. It is better to separate the product from its neighbor nodes.

Padding Field

This field requires the padding of the component. In essence, this area is the space around the subject area and inside the border-box. The height and the width of the padding box decide its proportions.

Content Field

Material such as text, photographs, or other digital media is included in this area.

It is constrained by the information edge, and its proportions are dictated by the width and height of the content enclosure.

Elements of the width and height

Typically, when you assign the width and height of an attribute using the CSS width and height assets, it means you just positioned the height and width of the subject areas of that component. The additional height and width of the unit box is based on a range of influences.

The specific area that an element box may occupy on a web page is measured as follows-

Size of the box	**Properties of CSS**
Height	height + padding-top + padding-bottom + border-top + border-bottom + margin-top + margin-bottom
Width	width + padding-left + padding-right + border-left + border-right + margin-left + margin-right

Example 1

Here, to explain the CSS box model, we have an instance.

```
<!DOCTYPE html>
<head>
<title>CSS Box Model</title>
<style>
            .main
{
                font-size:30px;
```

```
                        font-weight:bold;
                        Text-align:center;
                    }
                    .gfg
{
                        margin-left:50px;
                        border:50px solid Purple;
                        width:300px;
                        height:200px;
                        text-align:center;
                        padding:50px;
                    }
                    .gfg1
{
                        font-size:40px;
                        font-weight:bold;
                        color:black;
                        margin-top:60px;
                        background-color:purple;
                    }
                    .gfg2
{
                        font-size:20px;
                        font-weight:bold;
                        background-color:white;
                    }
</style>
</head>
```

```
<body>
<div class = "main">CSS Box-Model Property</div>
            <div class = "gfg">
            <div class = "gfg1">JavaTpoint</div>
            <div class = "gfg2">A best portal for
learn Technologies</div>
</div>
</body>
</html>
```

Example 2

Here, we also have an illustration to describe the CSS box model.

```
<!DOCTYPE html>
<head>
<style>
        .main
      {
            font-size:30px;
            font-weight:bold;
            text-align:left;
        }
#box
     {
            padding-top:30px;
            width: 300px;
            height: 100px;
            border: 40px solid red;

            margin: 30px;

            text-align:center;

            font-size:32px;

            font-weight:bold;

        }

</style>

</head>

<body>

        <div class="main">CSS Box-Model Property</div>

        <div id="box">JavaTpoint</div>

</body>

</html>
```

Important Point: In the CSS box model, the subject area of an entity box is the region where the content, such as image, text, video, etc., initially appeared. It may also retain boxes of decedent elements.

CSS Styles Conflicts Between Selectors

CSS conflicts between selectors occur when two or more selectors have conflicting styles applied to the same element, leading to unexpected results. In such cases, the browser has to decide which style to apply to the element. In this article, we’ll understand what causes conflicting styles in CSS, and how to avoid and resolve them.

Causes of Conflicting Styles in CSS: There are several ways that conflicting styles can arise in CSS. One common cause is using multiple stylesheets, either external or internal, that have overlapping or conflicting rules. When styles from multiple stylesheets are applied to the same element, the browser will have to decide which style to use, leading to conflicts.

Another cause of conflicting styles is the use of multiple selectors that apply to the same element. If these selectors have conflicting styles, the browser will have to decide which style to apply, based on the specificity and order of the selectors.

There are three reasons for CSS style conflict:

- Specificity
- Inheritance
- !important

Specificity: The more specific selector will override the less specific one. Specificity is calculated based on the number of elements, classes, and IDs in the selector. For example, the selector with an ID has a higher specificity than the one with a class, and the one with a class has a higher specificity than the one with an element.

Syntax:

```
p {
    color: blue;
}

.special {
    color: red;
}
```

Example: In this example, we have two conflicting styles for the color property applied to the same element p.special. However, the .special selector has a higher specificity than the p selector, so the color: red style will override the color: blue style.

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible"
content="IE=edge" />
    <meta name="viewport" content=
        "width=device-width, initial-scale=1.0" />
    <style>
        p {
            color: blue;
        }

        .special {
                color: red;
            }
    </style>
</head>

<body>
    <p class="special">This is a red text.</p>
</body>

</html>
```

Inheritance: Inheritance is the process by which styles applied to a parent element are passed down to its child elements. If two selectors have the same specificity, the one that comes later in the stylesheet will override the earlier one. If one of the conflicting styles is inherited, the browser will use the inherited style over the non-inherited style, even if the non-inherited style has a higher specificity or is defined later in the cascade order.

Syntax:

```
/* Style applied to the parent element */
.parent {
    font-size: 14px;
}

/* Conflicting style applied to the child element */
.child {
    font-size: 16px;
}
```

Example: In this example, the parent element has a font size of 14px applied to it, while the child element has a conflicting font size of 16px applied to it. However, because the font size is an inherited property, the child element will inherit the font size of 14px from the parent element, unless a more specific style is applied directly to the child element.

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible"
content="IE=edge" />
    <meta name="viewport" content=
        "width=device-width, initial-scale=1.0" />

    <style>
        /* Style applied to the parent element */
        .parent {
            font-size: 14px;
        }

        /* Conflicting style applied to
        the child element */
        .child {
            font-size: 16px;
        }
    </style>
</head>
```

```
<body>
    <div class="parent">
        <p>
            Parent element with font-size of 14px
        </p>
        <div class="child">
            <p>
                Child element with conflicting
                font-size of 16px
            </p>
        </div>
    </div>
</body>

</html>
```

!important: The styles marked with the !important keyword will always take precedence over any other styles.

```
Syntax:

p {
    color: blue !important;
}

.special {
    color: red;
}
```

Example: In this example, we have two conflicting styles for the color property applied to the same element p.special. However, the color: blue !important style has the highest priority due to the !important keyword, so it will override the color: red style.

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible"
content="IE=edge" />
    <meta name="viewport" content=
        "width=device-width, initial-scale=1.0" />
    <style>
        p {
            color: blue !important;
        }

        .special {
            color: red;
        }
    </style>
</head>
<body>
    <p class="special">This is a red text.</p>
</body>

</html>
```