CSS Tutorial:

CSS is Language That Decribes The Style of an HTML Document
CSS describes how HTML elements should be displayed

Ex:<style>

```
body{
background-color:#669900
}
h1{
color:white;
text-align:center;
}
h2{
font-family:verdand;
font-size:20px;
}
</style>
</head>
<body>
<h1>ALVIN AND CHIPMUNKS THE ROAD CHIP</h1>
<h2>ALVIN</h2>
```

%%What is CSS?
---
*CSS is Cascading Style Sheets
*CSS describes how HTML elements are to be displayed on screen,paper,or in other media
*CSS saves a lot of work.It Can control The layout of multiple web pages all at once
*External stylesheets are stored in CSS files.

%%Why Use CSS?

CSS is used to define styles for your web pages,including the design,layout and variations in display for different devices and screen sizes.

%%CSS Solved a Big Problem

HTML was NEVER intended to contain tags for formatting a web page!
HTML was created to describe the content of a web page, like:

<h1>This is a heading</h1>

This is a paragraph.

When tags like <font>, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created CSS.

CSS removed the style formatting from the HTML page!

%%CSS Saves a Lot of Work!

The style definitions are normally saved in external .css files.

With an external stylesheet file, you can change the look of an entire website by changing just one file!

CSS Syntax and Selectors

##################
###CSS Syntax::###
##################
-
A CSS rule-set consists of a selector and a declaration block:

EX:::
Selector: h1
      Declaration{color:blue;}   In This Color is Property and blue is Value
      Declaration{font-size:20px;}

The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

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

A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

Ex:
```
<style>
p {
    color: red;
    text-align: center;
}
</style>
</head>
<body>

<p>Hello World!</p>
<p>This paragraph is styled with CSS.</p>
```

#################
##CSS Selectors##
#################
-
CSS Selectors are used to "find" (OR SELECT) HTML elements based on their element name,id,class,attribute,and more.

--#>The element Selector

The element selector elements based on the element name.

you can select all <p> elements on a page like this (in this case,all<p> elements will be center-aligned,with a red text color):

Ex:
```
<style>
p {
    text-align: center;
    color: red;
}
</style>
```

```
</head>
<body>

<p>Every paragraph will be affected by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>
-------------------------
--#>The ID Selctor

The id selector uses the id attribute of an HTML element to select a
specific element.

The id of an element should be unique within a page, so the id selector
is used to select one unique element!

To select an element with a specific id, write a hash (#) character,
followed by the id of the element.

The style rule below will be applied to the HTML element with id="#p1":

**Note**: An id name cannot start with a number!

Ex:
#p1{
text-align:right;
color:blue;
font-size:21px;
font-family:arial;
}
/style>
</head>
<body>
 id="p1" >SIMON is also My Friend</p>
<p id="p1" >SIMON is also My Friend</p>
<p id="p1" >SIMON is also My Friend</p>
</body>
---------------------------
--#>The class Selector

The class selector selects elements with a specific class attribute.

To select elements with a specific class, write a period (.) character,
followed by the name of the class.

In the example below, all HTML elements with class="left" will be blue
and left-aligned:

Ex:
.left{
text-align:left;
color:blue;}

 class="left">ALVIN is Leader</h2>
<h2 class="left">ALVINis Leader</h2>
===
You can also specify that only specific HTML elements should be affected
by a class.

In the example below, only <p> elements with class="center" will be
center-aligned:
```

Ex:
div.c{
background-color:#5c8a8a;
}

Note: A class name cannot start with a number!

--#>Grouping Selectors

If you have elements with the same style definitions,

It will be better to group the selectors, to minimize the code.

To group selectors, separate each selector with a comma.

In the example below we have grouped the selectors from the code above:

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

--#>CSS Comments

Comments are used to explain the code, and may help when you edit the source code at a later date.

Comments are ignored by browsers.

A CSS comment starts with /* and ends with */. Comments can also span multiple lines:

```
Ex:/* This is
a multi-line
comment */
```

CSS How TO....

When a browser reads a style sheet,it will format the HTML document according to the information in the style sheet.

Three Ways to Insert CSS
i)External style sheet
ii)Internal style shhet
iii)Inline Style
---
i)External Style Sheet

With an external style sheet, you can change the look of an entire website by changing just one file!

Each page must include a reference to the external style sheet file inside the <link> element. The <link> element goes inside the <head> section:

An external style sheet can be written in any text editor. The file should not contain any html tags. The style sheet file must be saved with a .css extension.

Note: Do not add a space between the property value and the unit (such as margin-left: 20 px;). The correct way is: margin-left: 20px;

Here is how the "2.css" looks:

```
body{
      background-color:#33ffbb;
      color: #1a000d;
      text-family:vardana;
}
```

Actual Program in HTML:

```
html>
<head>
<link
rel="stylesheet"type="text/css"href="file:///C:/Users/danda.satish/Deskto
p/Alvin/CSS/2.css">
</head>
<body>
<h1>hkdb</h1>
</body>
</html>
```

---
ii)Internal Style Sheet

An internal style sheet may be used if one single page has a unique style.

Internal styles are defined within the <style> element, inside the <head> section of an HTML page:

Ex:

```
 <head>
<style>
body {
    background-color: linen;
}
```

```
h1 {
    color: maroon;
    margin-left: 40px;
}
</style>
</head>
```
---
iii)Inline Styles

An inline style may be used to apply a unique style for a single element.

To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

The example below shows how to change the color and the left margin of a <h1> element:

```
 <h1 style="color:blue;margin-left:30px;">This is a heading.</h1>
```

Tip: An inline style loses many of the advantages of a style sheet (by mixing content with presentation). Use this method sparingly
---
#$#$
Multiple Style Sheets

If some properties have been defined for the same selector(element) in different style sheets,
the value from the last read style sheet will be used.

Example:

Assume that an external style sheet has the following style for the <h1> element:

```
h1 {
    color: navy;
}
```

then, assume that an internal style sheet also has the following style for the <h1> element:

```
h1 {
    color: orange;
}
```

If the internal style is defined after the link to the external style sheet, the <h1> elements will be "orange":
Ex:

```
 <head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<style>
h1 {
    color: orange;
}
</style>
</head>
```

However, if the internal style is defined before the link to the external style sheet, the <h1> elements will be "navy":
Ex:

```
 <head>
<style>
```

```
h1 {
    color: orange;
}
</style>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

Cascading Order

What style will be used when there is more than one style specified for an HTML element?

Generally speaking we can say that all the styles will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:

1. Inline style (inside an HTML element)
2. External and internal style sheets (in the head section)
3. Browser default

So, an inline style (inside a specific HTML element) has the highest priority, which means that it will override a style defined inside the <head> tag, or in an external style sheet, or a browser default value.

Colors in CSS are most often specified by:

    a valid color name - like "red"
    an RGB value - like "rgb(255, 0, 0)"
    a HEX value - like "#ff0000"

Color Names

Colors set by using color names:
Example
ColorName:::Red,Green,Blue,Orange,Yellow,Cyan,Black

Note: Color names are case-insensitive: "Red" is the same as "red" or "RED".

HTML and CSS supports 140 standard color names.

RGB (Red, Green, Blue)

RGB color values can be specified using this formula: rgb(red, green, blue).

Each parameter (red, green, blue) defines the intensity of the color between 0 and 255.

For example, rgb(255,0,0) is displayed as red, because red is set to its highest value (255) and the others are set to 0. Experiment by mixing the RGB values below:

Red	Green	Blue
255	0	0

rgb(255, 0, 0)
Example
ColorRGB:::rgb(255,0,0),rgb(0,255,0),rgb(0,0,255),rgb(255,165,0),rgb(255,255,0),rgb(0,255,255).

Shades of grey are often defined using equal values for all the 3 light sources:
Example

Color	RGB
	rgb(0,0,0)
	rgb(128,128,128)
	rgb(255,255,255)

Hexadecimal Colors

RGB values can also be specified using hexadecimal color values in the form: #RRGGBB, where RR (red), GG (green) and BB (blue) are hexadecimal values between 00 and FF (same as decimal 0-255).

For example, #FF0000 is displayed as red, because red is set to its highest value (FF) and the others are set to the lowest value (00). Note: HEX values are case-insensitive: "#ff0000" is the same as "FF0000".
Example
Color HEX:::
 #FF0000,#00FF00,#0000FF,#FFA500,#FFFF00,#00FFFF.

Shades of grey are often defined using equal values for all the 3 light sources:

Example

Color	HEX
	#000000
	#808080
	#FFFFFF

CSS Backgrounds

The CSS background properties are used to define the background effects for elements.

CSS background properties:

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

1.background-color:
The background-color property specify the backkground color of an element.

The background color of a page is set like this:

Ex:
```
body {
    background-color: lightblue;
}
```
With CSS, a color is most often specified by:

- a valid color name - like "red"
- a HEX value - like "#ff0000"
- an RGB value - like "rgb(255,0,0)"

EX:
```
<head>
<style>
h1{
background-color:green;
}
div{background-color:lightblue;
}
p{
background-color:yellow;
}
</style>
<link
rel="stylesheet"type="text/css"href="file:///C:/Users/danda.satish/Deskto
p/Alvin/CSS/2.css">
</head>
```

2.Background Image:
The Background-image property specifies an image to use as the background of an element.
By default,the image is repuated so it covers the entire element.
The background image for a page can be set like this:

Ex:
```
body{
background-image:url("ngh.gif")
}
```
Below is an example of a bad combination of text and background image. The text is hardly readable:
Ex:
```
body {
    background-image: url("bgdesert.jpg");
```

```
}
```
Note: When using a background image, use an image that does not disturb the text.

CSS.File is::

```
body{
      background-
image:url("file:///C:/Users/danda.satish/Desktop/Alvin/1.jpg");
      color: #1a000d;
      text-family:vardana;
}
<link
rel="stylesheet"type="text/css"href="file:///C:/Users/danda.satish/Deskto
p/Alvin/CSS/2.jpg">
```

Background Image - Repeat Horizontally or Vertically

By default, the background-image property repeats an image both horizontally and vertically.

Some images should be repeated only horizontally or vertically, or they will look strange, like this:

Ex:

```
body{
      background-
image:url("file:///C:/Users/danda.satish/Desktop/Alvin/1.jpg");
      background-repeat: repeat-x;
      color: #1a000d;
      text-family:vardana;
}
<link
rel="stylesheet"type="text/css"href="file:///C:/Users/danda.satish/Deskto
p/Alvin/CSS/2.jpg">
```

Another Ex:

Tip: To repeat an image vertically, set background-repeat: repeat-y;

```
body{
      background-
image:url("file:///C:/Users/danda.satish/Desktop/Alvin/1.jpg");
      background-repeat: repeat-y;
      color: #1a000d;
      text-family:vardana;
}
<link
rel="stylesheet"type="text/css"href="file:///C:/Users/danda.satish/Deskto
p/Alvin/CSS/2.jpg">
```

Background Image - Set position and no-repeat

Showing the background image only once is also specified by the background-repeat property:

Ex:

```
body{
background-
image:url("file:///C:/Users/danda.satish/Desktop/Alvin/2.jpg");
```

```
background-repeat:no-repeat;
}
```

In the example above, the background image is shown in the same place as the text. We want to change the position of the image, so that it does not disturb the text too much.

The position of the image is specified by the background-position property:

```
body{
background-
image:url("file:///C:/Users/danda.satish/Desktop/Alvin/2.jpg");
background-repeat:no-repeat;
background-position:right-top;
margin-right:200px;
}
```

Background Image - Fixed position

To specify that the background image should be fixed (will not scroll with the rest of the page), use the background-attachment property:

Ex:

```
body{
background-
image:url("file:///C:/Users/danda.satish/Desktop/Alvin/2.jpg");
background-repeat:no-repeat;
background-position:right-top;
margin-right:200px;
background-attachment:fixed;
}
```

Background - Shorthand property

To shorten the code, it is also possible to specify all the background properties in one single property. This is called a shorthand property.

The shorthand property for background is background:

EX:

```
body{
background:blue url("file:///C:/Users/danda.satish/Desktop/Alvin/3.jpg")
no-repeat right top fixed;
}
```

When using the shorthand property the order of the property values is:

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

It does not matter if one of the property values is missing, as long as the other ones are in this order.

All CSS Background Properties

Property	Description
background	Sets all the background properties in one declaration
background-attachment	Sets whether a background image is fixed or scrolls with the rest of the page
background-color	Sets the background color of an element
background-image	Sets the background image for an element
background-position	Sets the starting position of a background image
background-repeat	Sets how a background image will be repeated

CSS Borders

CSS Border Properties

The CSS border properties allow you to specify the style, width, and color of an element's border.

Border Style:::

The border-style property specifies what kind of border to display.

The following values are allowed:

 1) dotted - Defines a dotted border
 2) dashed - Defines a dashed border
 3) solid   - Defines a solid border
 4) double - Defines a double border
 5) groove - Defines a 3D grooved border. The effect depends on the border-color value
 6) ridge   - Defines a 3D ridged border. The effect depends on the border-color value
 7) inset   - Defines a 3D inset border. The effect depends on the border-color value
 8) outset - Defines a 3D outset border. The effect depends on the border-color value
 9) none   - Defines no border
10) hidden - Defines a hidden border

The border-style property can have from one to four values (for the top border, right border, bottom border, and the left border).

Example::
```
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {border-style: dotted dashed solid double;}
```

Note: None of the OTHER CSS border properties described below will have ANY effect unless the border-style property is set!

Border Width::

The border-width property specifies the width of the four borders.

The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: thin, medium, or thick.

The border-width property can have from one to four values (for the top border, right border, bottom border, and the left border).

EX:

```
p.one{
border-style:solid;
border-width:2px;
border-color:red;
}
p.two{
border-style:solid;
border-width:medium;
}
p.three{
border-style:dotted;
border-width:2px;
}
p.four{
border-style:dotted;
border-width:thick;
}
p.five{
border-style:double;
border-width:thick;
}
p.six{
border-style:double;
border-width:thick;
}
p.seven{
border-style:groove;
border-width:12px;
}
p.eight{
border-style:ridge;
border-width:12px;
}
p.ni{
border-style:inset;
border-width:12px;
}
p.ten{
border-style:outset;
border-width:12px;
}
p.el{
border-style:none;
border-width:12px;
}
 class="one">SATISH</p>
      <p class="two">SATISH</p>
      <p class="three">SATISH</p>
      <p class="four">SATISH</p>
      <p class="five">SATISH</p>
      <p class="six">SATISH</p>
      <p class="seven">SATISH</p>
      <p class="eight">SATISH</p>
      <p class="ni">SATISH</p>
      <p class="ten">SATISH</p>
      <p class="el">SATISH</p>
------------------------------
------------------------------
Border Color
```

The border-color property is used to set the color of the four borders.

The color can be set by:

    name - specify a color name, like "red"
    Hex - specify a hex value, like "#ff0000"
    RGB - specify a RGB value, like "rgb(255,0,0)"
    transparent

The border-color property can have from one to four values (for the top border, right border, bottom border, and the left border).

If border-color is not set, it inherits the color of the element.

EX:border-color:red;

Border - Individual Sides

From the examples above you have seen that it is possible to specify a different border for each side.

In CSS, there is also properties for specifying each of the borders (top, right, bottom, and left):
Ex:
```
#qw{
border-top-style:dotted;
border-right-style:solid;
border-bottom-style:outset;
border-left-style:solid;
border-width:12px;}
<p id="qw">ALVIN AND CHIPMUNKS</p>
```

Ex:
```
p {
    border-style: dotted solid;
}
```

So, here is how it works:

If the border-style property has four values:

    border-style: dotted solid double dashed;
        top border is dotted
        right border is solid
        bottom border is double
        left border is dashed

If the border-style property has three values:

    border-style: dotted solid double;
        top border is dotted
        right and left borders are solid
        bottom border is double

If the border-style property has two values:

    border-style: dotted solid;
        top and bottom borders are dotted
        right and left borders are solid

If the border-style property has one value:

```
    border-style: dotted;
        all four borders are dotted
```

The border-style property is used in the example above. However, it also works with border-width and border-color.

Border - Shorthand Property

As you can see from the examples above, there are many properties to consider when dealing with borders.

To shorten the code, it is also possible to specify all the individual border properties in one property.

The border property is a shorthand property for the following individual border properties:

```
    border-width
    border-style (required)
    border-color
```

Ex:

```
p {
    border: 5px solid red;
}
```

You can also specify all the individual border properties for just one side:

Left Border:

```
p {
    border-left: 6px solid red;
    background-color: lightgrey;
}
```

Bottom Border:

```
p {
    border-bottom: 6px solid red;
    background-color: lightgrey;
}
```

Rounded Borders::

```
p.round{
border : 3px groove red;
border-radius:6px;}

<p class = "round" > Groove</p>
```

Note: The border-radius property is not supported in IE8 and earlier versions.

All CSS Border Properties

Property	Description

```
border           Sets all the border properties in one declaration
border-bottom          Sets all the bottom border properties in one
declaration
border-bottom-color    Sets the color of the bottom border
border-bottom-style    Sets the style of the bottom border
border-bottom-width    Sets the width of the bottom border
border-color           Sets the color of the four borders
border-left            Sets all the left border properties in one
declaration
border-left-color      Sets the color of the left border
border-left-style      Sets the style of the left border
border-left-width      Sets the width of the left border
border-radius          Sets all the four border-*-radius properties for
rounded corners
border-right           Sets all the right border properties in one
declaration
border-right-color     Sets the color of the right border
border-right-style     Sets the style of the right border
border-right-width     Sets the width of the right border
border-style           Sets the style of the four borders
border-top       Sets all the top border properties in one declaration
border-top-color       Sets the color of the top border
border-top-style       Sets the style of the top border
border-top-width       Sets the width of the top border
border-width           Sets the width of the four borders
-----------------------------------------
```

CSS Margins:

CSS Margins are used to generate space around elments.

The Margin properties set the size of the white space outside the border.

With CSS , you have full control over the margins.There are CSS properties for setting the margin for each side of an element(top,right,bottom,and left).

Margin - Individual Sides

1)margin-top
2)margin-right
3)margin-bottom
4)margin-left

All The Margin properties can have the foll values:

auto-the browser calculates the margin
length-specifes a margin in px,pt,cm,etc.
%-specifies a margin in % of the width of the containing element
inherit - specifies that the margin should be inherited from the parent element

Tip: Negative values are allowed.

```
p {
    margin-top: 100px;
    margin-bottom: 100px;
    margin-right: 150px;
    margin-left: 80px;
}
```

Margin - Shorthand Property

To shorten the code, it is possible to specify all the margin properties in one property.

The margin property is a shorthand property for the following individual margin properties:

margin-top
margin-right
margin-bottom
margin-left

Example
```
p {
    margin: 100px 150px 100px 80px;
}
```

So, here is how it works:

If the margin property has four values:

margin: 25px 50px 75px 100px;
top margin is 25px
right margin is 50px
bottom margin is 75px

```
        left margin is 100px

If the margin property has three values:

    margin: 25px 50px 75px;
        top margin is 25px
        right and left margins are 50px
        bottom margin is 75px

If the margin property has two values:

    margin: 25px 50px;
        top and bottom margins are 25px
        right and left margins are 50px

If the margin property has one value:

    margin: 25px;
        all four margins are 25px
-------------------------
The auto Value

You can set the margin property to auto to horizontally center the
element within its container.

The element will then take up the specified width, and the remaining
space will be split equally between the left and right margins:
EX:
p{
width:auto;
height:auto;
border:2px solid green;
margin:auto;
}
----------------
The inherit Value

This example lets the left margin be inherited from the parent element:
Ex:
<!DOCTYPE html>
<html>
<head>
<style>
div.container {
    border: 1px solid red;
    margin-left: 100px;
}

p.one {
    margin-left: inherit;
}
</style>
</head>
<body>

<h2>Use of the inherit value</h2>
<p>Let the left margin be inherited from the parent element:</p>

<div class="container">
```

```
<p class="one">This is a paragraph with an inherited left margin (from
the div element).</p>
</div>

</body>
</html>
--------------
Margin Collapse

Top and bottom margins of elements are sometimes collapsed into a single
margin that is equal to the largest of the two margins.

This does not happen on horizontal margins (left and right)! Only
vertical margins (top and bottom)!

Look at the following example:
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
    margin: 0 0 50px 0;
}

h2 {
    margin: 20px 0 0 0;
}
</style>
</head>
<body>

<p>In this example the h1 element has a bottom margin of 50px and the h2
element has a top margin of 20px. Then, the vertical margin between h1
and h2 should have been 70px (50px + 20px). However, due to margin
collapse, the actual margin ends up being 50px.</p>

<h1>Heading 1</h1>
<h2>Heading 2</h2>

</body>
</html>
----------------------------
ll CSS Margin Properties
Property   Description
margin     A shorthand property for setting the margin properties in one
declaration
margin-bottom    Sets the bottom margin of an element
margin-left      Sets the left margin of an element
margin-right     Sets the right margin of an element
margin-top Sets the top margin of an element
----------------------
```

CSS Padding

CSS Padding propeties are used to generate space around content.

The padding clears an area around the Content (inside the border) of an element.

With CSS, you have full control over the padding. There are CSS Properties for setting the padding for each side of an element(top,right,bottom,and left).

Padding-Individual Sides

CSS has properties for specifing the padding for each side of an element.

1)padding-top
2)padding-right
3)padding-bottom
4)padding-left

All the padding properties can have the following values:

    length - specifies a padding in px, pt, cm, etc.
    % - specifies a padding in % of the width of the containing element
    inherit - specifies that the padding should be inherited from the parent element

---
The following example sets different padding for all four sides of a <p> element:

```
div {
    border: 1px solid black;
    background-color: lightblue;
    padding-top: 50px;
    padding-right: 30px;
    padding-bottom: 50px;
    padding-left: 80px;
}
<div>This div element has a top padding of 50px, a right padding of 50px, a bottom padding of 50px, and a left padding of 80px.</div>
```

Padding - Shorthand Property

To shorten the code, it is possible to specify all the padding properties in one property.

The padding property is a shorthand property for the following individual padding properties:

    padding-top
    padding-right
    padding-bottom
    padding-left

Ex:

```
p {
    padding: 50px 30px 50px 80px;
}
```

So, here is how it works:

```
>>If the padding property has four values:

    padding: 25px 50px 75px 100px;
        top padding is 25px
        right padding is 50px
        bottom padding is 75px
        left padding is 100px

>>If the padding property has three values:

    padding: 25px 50px 75px;
        top padding is 25px
        right and left paddings are 50px
        bottom padding is 75px

>>If the padding property has two values:

    padding: 25px 50px;
        top and bottom paddings are 25px
        right and left paddings are 50px

>>If the padding property has one value:

    padding: 25px;
        all four paddings are 25px
---------------
div.ex1 {
    padding: 25px 50px 75px 100px;
}

div.ex2 {
    padding: 25px 50px 75px;
}

div.ex3 {
    padding: 25px 50px;
}

div.ex4 {
    padding: 25px;
}

------------------
All CSS Padding Properties
Property   Description
padding    A shorthand property for setting all the padding properties in
one declaration
padding-bottom   Sets the bottom padding of an element
padding-left     Sets the left padding of an element
padding-right    Sets the right padding of an element
padding-top      Sets the top padding of an elemen
```

CSS Height and Width

Setting height and Width

The height and width properties are used to set the height and width of an element.

The height and width can be set to auto (this is default.Means that the browser calculates the height and width),or be specified in length values,like px,cm,etc., or in
Percent(%) of the containing block.

CSS Box Model::

The CSS Box Model

All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model:

Explanation of the different parts:

    Content - The content of the box, where text and images appear
    Padding - Clears an area around the content. The padding is transparent
    Border - A border that goes around the padding and content
    Margin - Clears an area outside the border. The margin is transparent

The box model allows us to add a border around elements, and to define space between elements.

Ex:
```
.box{
background-color:lightred;
width:300px;
border:25px solid blue;
padding:30px;
margin:24px;
}
div class="box">
This text is the actual content of the box. We have added a 25px padding, 25px margin and a 25px green border. Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
 Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
 Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum
      </div>
```

Width and Height of an Element

In order to set the width and height of an element correctly in all browsers, you need to know how the box model works.

Important: When you set the width and height properties of an element with CSS, you just set the width and height of the content area. To calculate the full size of an element, you must also add padding, borders and margins.

Assume we want to style a element to have a total width of 350px:

Ex:
```
<!DOCTYPE html>
<html>
<head>
<style>
div {
```

```
    width: 320px;
    padding: 10px;
    border: 5px solid gray;
    margin: 0;
}
</style>
</head>
<body>

<h2>Calculate the total width:</h2>

<img src="klematis4_big.jpg" width="350" height="263" alt="Klematis">
<div>The picture above is 350px wide. The total width of this element is
also 350px.</div>

</body>
</html>
--------------------------
Here is the math:
320px (width)
+ 20px (left + right padding)
+ 10px (left + right border)
+ 0px (left + right margin)
= 350px

The total width of an element should be calculated like this:

Total element width = width + left padding + right padding + left border
+ right border + left margin + right margin

The total height of an element should be calculated like this:

Total element height = height + top padding + bottom padding + top border
+ bottom border + top margin + bottom margin

Note for old IE: Internet Explorer 8 and earlier versions, include
padding and border in the width property. To fix this problem, add a
<!DOCTYPE html> to the HTML page.

---------------------------------
```

CSS Outline:

CSS Outline

The CSS outline property specify the style,color and width of an outline.

An outline is a line that is drawn around elements (outside the borders)to make the element "stand out".

However , the outline property is different from the border property-the outline is NOT a part of an elements dimensions;
the elements total width and height is not affected by the width of the outline.

Outline Style

The outline-style property specifies the style of the outline.

The outline-style property can have one of the following values:

- dotted - Defines a dotted outline
- dashed - Defines a dashed outline
- solid - Defines a solid outline
- double - Defines a double outline
- groove - Defines a 3D grooved outline. The effect depends on the outline-color value
- ridge - Defines a 3D ridged outline. The effect depends on the outline-color value
- inset - Defines a 3D inset outline. The effect depends on the outline-color value
- outset - Defines a 3D outset outline. The effect depends on the outline-color value
- none - Defines no outline.
- hidden - Defines a hidden outline.

The Following Example first sets a thin black border around each <p> element,then it shows diff outline-style values:

Ex:
```
<!DOCTYPE html>
<html>
<head>
<style>
p{
border:1px solid black;
outline-color:blue;
}
p.dotted{outline-style:dotted;}
p.dashed{outline-style:dashed}
p.solid{outline-style:solid;}
p.double{outline-style:double;}
p.groove{outline-style:groove;}
p.ridge{outline-style:ridge;}
p.inset{outline-style:inset;}
p.outset{outline-style:outset;}
</style>
</head>
<body>
<p class="dotted"> A Dotted Outline</p>
<p class="dashed"> A Dashed Outline</p>
```

```
<p class="solid"> A Solid Outline</p>
<p class="double"> A DOUBLE Outline</p>
<p class="groove"> A GROOVE Outline</p>
<p class="ridge"> A ridge Outline</p>
<p class="inset"> A INSET Outline</p>
<p class="outset"> A OUT Outline</p>
</body>
</html>
```

Note: None of the OTHER CSS outline properties described below will have ANY effect unless the outline-style property is set!

Outline Color:

The outline-color property is used to set the color of the outline.

The color can be set by:

- name - specify a color name, like "red"
- RGB - specify a RGB value, like "rgb(255,0,0)"
- Hex - specify a hex value, like "#ff0000"
- invert - performs a color inversion (which ensures that the outline is visible, regardless of color background)

Ex:

```
p {
    border: 1px solid black;
    outline-style: double;
    outline-color: red;
}
```

Outline Width

The outline-width property specifies the width of the outline.

The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: thin, medium, or thick.

Ex:

```
p {border: 1px solid black;}

p.one {
    outline-style: double;
    outline-color: red;
    outline-width: thick;
}

p.two {
    outline-style: double;
    outline-color: green;
    outline-width: 3px;
}
```

Outline - Shorthand property

To shorten the code, it is also possible to specify all the individual outline properties in one property.

The outline property is a shorthand property for the following individual outline properties:

```
    outline-width
    outline-style (required)
    outline-color
p {
    border: 1px solid black;
    outline: 5px dotted red;
}
```

All CSS Outline Properties

Property	Description
outline	Sets all the outline properties in one declaration
outline-color	Sets the color of an outline
outline-offset	Specifies the space between an outline and the edge or border of an element
outline-style	Sets the style of an outline
outline-width	Sets the width of an outline

CSS Text

Text Color:

The Color property is used to set the color of the text.

With CSS, a color is used to set the color of the text.

Ex:

```
body {
    color: blue;
}

h1 {
    color: green;
}
</style>
</head>
<body>

<h1>This is heading 1</h1>
<p>This is an ordinary paragraph. Notice that this text is blue. The
default text color for a page is defined in the body selector.</p>
```

Note: For W3C compliant CSS: If you define the color property, you must also define the background-color

---
Text Alignment

The text-align property is used to set the horizontal alignment of a text.

A text can be left or right aligned, centered, or justified.

The following example shows center aligned, and left and right aligned text (left alignment is default if text direction is left-to-right, and right alignment is default if text direction is right-to-left):

EX:

```
h1 {
    text-align: center;
}

h2 {
    text-align: left;
}

h3 {
    text-align: right;
}
```

When the text-align property is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazines and newspapers):

Example:

```
div {
    text-align: justify;
```

```
}
```

Text Decoration

The text-decoration property is used to set or remove decorations from text.

The value text-decoration: none; is often used to remove underlines from links:

```
a {
    text-decoration: none;
}
```

The other text-decoration values are used to decorate text:

Example

```
h1 {
    text-decoration: overline;
}

h2 {
    text-decoration: line-through;
}

h3 {
    text-decoration: underline;
}
```

Note: It is not recommended to underline text that is not a link, as this often confuses the reader.

Text Transformation

The text-transform property is used to specify uppercase and lowercase letters in a text.

It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word:

eX:

```
p.uppercase {
    text-transform: uppercase;
}

p.lowercase {
    text-transform: lowercase;
}

p.capitalize {
    text-transform: capitalize;
}
```

Text Indentation

The text-indent property is used to specify the indentation of the first line of a text:

Ex:
```
p {
    text-indent: 50px;
}
```

Letter Spacing

The letter-spacing property is used to specify the space between the characters in a text.

The following example demonstrates how to increase or decrease the space between characters:

Ex:
```
h1 {
    letter-spacing: 3px;
}

h2 {
    letter-spacing: -3px;
}
```

Line Height

The line-height property is used to specify the space between lines:
Ex:
```
p.small {
    line-height: 0.8;
}

p.big {
    line-height: 1.8;
}
```

Text Direction

The direction property is used to change the text direction of an element:

Ex:
```
div {
    direction: rtl;
}
```

Word Spacing

The word-spacing property is used to specify the space between the words in a text.

The following example demonstrates how to increase or decrease the space between words:

Ex:
Example
```
h1 {
    word-spacing: 10px;
```

```
}

h2 {
    word-spacing: -5px;
}
```

All CSS Text Properties

Property	Description
color	Sets the color of text
direction	Specifies the text direction/writing direction
letter-spacing	Increases or decreases the space between characters in a text
line-height	Sets the line height
text-align	Specifies the horizontal alignment of text
text-decoration	Specifies the decoration added to text
text-indent	Specifies the indentation of the first line in a text-block
text-shadow	Specifies the shadow effect added to text
text-transform	Controls the capitalization of text
unicode-bidi	Used together with the direction property to set or return whether the text should be overridden to support multiple languages in the same document
vertical-align	Sets the vertical alignment of an element
white-space	Specifies how white-space inside an element is handled
word-spacing	Increases or decreases the space between words in a text

---

CSS Fonts:

The CSS font properties define the font family,boldness,size,and the style of a text.

Diff Between Serif and Sans-serif Fonts (F-Sans-serif)(F-serif Times of Roman)

CSS Font Families

In CSS, there are two types of font family names:

1. generic family - a group of font families with a similar look (like "Serif" or "Monospace")
2. font family - a specific font family (like "Times New Roman" or "Arial")

Generic family	Font family	Description
Serif	Times New Roman Georgia	Serif fonts have small lines at the ends on some charcters
Sans-Serif	Arial	"Sans" means Without - these fonts do not have the lines at the ends of characters
Monospace	Courier New Lucida Console	All monospace characters have the same width

Note: On computer screens, sans-serif fonts are considered easier to read than serif fonts.

Font Family

The font family of a text is set with the font-family property.

The font-family property should hold several font names as a "fallback" system. If the browser does not support the first font, it tries the next font, and so on.

Start with the font you want, and end with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available.

Note: If the name of a font family is more than one word, it must be in quotation marks, like: "Times New Roman".

More than one font family is specified in a comma-separated list:

```
h2{
      font-family:"Arial";
}
```

Font Style

The font-style property is mostly used to specify italic text.

This property has three values:

    normal - The text is shown normally
    italic - The text is shown in italics
    oblique - The text is "leaning" (oblique is very similar to italic, but less supported)

```
h2{
      font-family:"Arial";
      font-style:"italic";
}
```

Font Size

The font-size property sets the size of the text.

Being able to manage the text size is important in web design. However, you should not use font size adjustments to make paragraphs look like headings, or headings look like paragraphs.

Always use the proper HTML tags, like <h1> - <h6> for headings and <p> for paragraphs.

The font-size value can be an absolute, or relative size.

Absolute size:

 1   Sets the text to a specified size
 2   Does not allow a user to change the text size in all browsers (bad for accessibility reasons)
 3 Absolute size is useful when the physical size of the output is known

Relative size:

  1 Sets the size relative to surrounding elements
  2 Allows a user to change the text size in browsers

Note: If you do not specify a font size, the default size for normal text, like paragraphs, is 16px (16px=1em).

Set Font Size With Pixels

Setting the text size with pixels gives you full control over the text size:

```
h1 {
    font-size: 40px;
}

h2 {
    font-size: 30px;
}

p {
    font-size: 14px;
}
```

Set Font Size With Em

To allow users to resize the text (in the browser menu), many developers use em instead of pixels.

The em size unit is recommended by the W3C.

1em is equal to the current font size. The default text size in browsers is 16px. So, the default size of 1em is 16px.

The size can be calculated from pixels to em using this formula: pixels/16=em

Ex:

```
h1 {
    font-size: 2.5em; /* 40px/16=2.5em */
}

h2 {
    font-size: 1.875em; /* 30px/16=1.875em */
}

p {
    font-size: 0.875em; /* 14px/16=0.875em */
}
```

In the example above, the text size in em is the same as the previous example in pixels. However, with the em size, it is possible to adjust the text size in all browsers.

Unfortunately, there is still a problem with older versions of IE. The text becomes larger than it should when made larger, and smaller than it should when made smaller.

Use a Combination of Percent and Em

The solution that works in all browsers, is to set a default font-size in percent for the <body> element:

Example

```
body {
    font-size: 100%;
}

h1 {
    font-size: 2.5em;
}

h2 {
    font-size: 1.875em;
}

p {
    font-size: 0.875em;
}
```

Our code now works great! It shows the same text size in all browsers, and allows all browsers to zoom or resize the text!

---

ont Weight

The font-weight property specifies the weight of a font:

Example

```
p.normal {
    font-weight: normal;
}

p.thick {
    font-weight: bold;
}
```

Font Variant

The font-variant property specifies whether or not a text should be displayed in a small-caps font.

In a small-caps font, all lowercase letters are converted to uppercase letters. However, the converted uppercase letters appears in a smaller font size than the original uppercase letters in the text.

Example

```
p.normal {
    font-variant: normal;
}

p.small {
    font-variant: small-caps;
}
```

All CSS Font Properties

Property	Description
font	Sets all the font properties in one declaration
font-family	Specifies the font family for text
font-size	Specifies the font size of text
font-style	Specifies the font style for text
font-variant	Specifies whether or not a text should be displayed in a small-caps font
font-weight	Specifies the weight of a font

CSS Links:

With Css, links can be styled in diff ways.

Styling Links:

Links can be styled with any CSS property(E.g Color,Font-Family,Background,etc..)
Ex:

```
.link{
color:lightgreen;
font-family:arial;
background:red;}

<a href="" class="link" target="default">LInk cLick mE</a>
```

In addition, links can be styled differently depending on what state they are in.

The four links states are:

- a:link - a normal, unvisited link
- a:visited - a link the user has visited
- a:hover - a link when the user mouses over it
- a:active - a link the moment it is clicked

```
<!DOCTYPE html>
<html>
<head>
<style>
/* unvisited link */
a:link {
    color: red;
}

/* visited link */
a:visited {
    color: green;
}

/* mouse over link */
a:hover {
    color: hotpink;
}

/* selected link */
a:active {
    color: blue;
}
</style>
</head>
<body>

<p><b><a href="default.asp" target="_blank">This is a link</a></b></p>
<p><b>Note:</b> a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.</p>
<p><b>Note:</b> a:active MUST come after a:hover in the CSS definition in order to be effective.</p>
```

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

When setting the style for several link states, there are some order rules:

a:hover MUST come after a:link and a:visited
a:active MUST come after a:hover.

Text Decoration::
The text-decoration property is mostly used to remove underlines from links.

```
a:visited{
   text-decoration:none;
      color:green;

}
a:hover{
      color:hotpink;
text-decoration:underline;
      }
a:active{
      color:blue;
      text-decoration:underline;
}
<a href ="" target="">Link</a>
```

Background Color:
The background-color property can be used to specify a background color for links:

Ex:

```
a:link{
text-decoration:none;
background-color:yellow;
      color;#b37700

}
a:visited{
background-color:cyan;
   text-decoration:none;
      color:green;

}
a:hover{
background-color:green;
      color:hotpink;
text-decoration:underline;
      }
a:active{
background-color:hotpink;
      color:blue;
      text-decoration:underline;
}


<a href ="" target="">Link</a>
```

Advanced - Link Buttons:

This Example demonstrates a more advanced example where we combine several CSS propertys to display like as boxes/buttons:

```
Ex:a:link,a:visited{
background-color:#3333ff;
padding:30px;
text-decoration:none;
test-align:center;
display:inline-block;
border-radius: 13px 13px 0px 0px; }
a:hover,a:active{
background-color:#ffe6f3;
text-decoration:underline;}

<a href ="" target="">Link</a>
```

CSS Lists:

HTML Lists and CSS List Properties.

In HTML, there are two main types of lists
 unordered lists- the list items are marked with bullets
ordered lists- the list items are marked with numbers or letters

The CSS list properties allow you to:
1) Set diff list item markers for ordered lists
2) Set diff list item markers for unordered lists
3) Set an Image as The list marker
4) Add background colors to lists and list items.

Diff List Items Markers:

The list type property specifies the type of list item marker.

The following ex shows some of the available list item markers:

Note:Some of the values are for unordered lists, and some for ordered lists.

Ex;

```
<!DOCTYPE html>
<html>
<head><title>Lists</title>
<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-greek;
}
</style>
</head>
<body>
<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>
<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>
      <li>Coffee</li>
      <li>Tea</li>
      <li>Coca Cola</li>
      </ol>
      </body>
</html>
-========================
```

An Image as The List Item Marker

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

```
ul {
    list-style-image: url('sqpurple.gif');
}
ul.a{
list-style-image:
url('file:///C:/Users/danda.satish/Desktop/Alvin/1.jpg')
}
==========================
```

Position The List Item Markers

The list-style-position property specifies whether the list-item markers should appear inside or outside the content flow:

Ex:

```
<!DOCTYPE html>
<html>
<head>
<style>
ul.a {list-style-position:inside;}
ul.b {list-style-position:outside;}
</style>
</head>

<body>
<p>The following list has list-style-position: inside:</p>
<ul class="a">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>

<p>The following list has list-style-position: outside:</p>
<ul class="b">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>

<p>"list-style-position: outside" is the default setting.</p>
</body>
</html>
---------------
```

List - Shorthand property

The list-style property is a shorthand property. It is used to set all the list properties in one declaration:

Example

```
ul {
    list-style: square inside url("sqpurple.gif");
}
```

When using the shorthand property, the order of the property values are:

    list-style-type (if a list-style-image is specified, the value of this property will be displayed if the image for some reason cannot be displayed)
    list-style-position (specifies whether the list-item markers should appear inside or outside the content flow)
    list-style-image (specifies an image as the list item marker)

If one of the property values above are missing, the default value for the missing property will be inserted, if any.

Styling List With Colors

We can also style lists with colors, to make them look a little more interesting.

Anything added to the <ol> or <ul> tag, affects the entire list, while properties added to the <li> tag will affect the individual list items:

Example:

```
ol {
    background: #ff9999;
    padding: 20px;
}

ul {
    background: #3399ff;
    padding: 20px;
}

ol li {
    background: #ffe5e5;
    padding: 5px;
    margin-left: 35px;
}

ul li {
    background: #cce5ff;
    margin: 5px;
}
```

CSS Tables:

The look of an HTML table can be greatly improved with CSS:

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Berglunds snabbköp	Christina Berglund	Sweden
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Königlich Essen	Philip Cramer	Germany
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy

Table Borders:

To specify table borders in CSS,use the border property.

The Example below specifies a black border for <table>,<th>,and <td> elements:

```
<!DOCTYPE html>
<html>
<head>
<style>
table,th,td{
border:1px solid green;
}
</style>
</head>
<body>
<table>
<tr>
      <th>First name</th>
      <th>Last Name</th>
</tr>
<tr>
      <td>Peter</td>
      <td>Griffin</td>
</tr>
<tr>
      <td>Peter</td>
      <td>Griffin</td>
</tr>
</table>
</body>
</html>
```

Notice that the table in the example above has double borders.
This is because both the table and the <th> and <td> elements have separate borders.

Collapse Table Borders

The border-collapse property sets whether the table borders should be collapsed into a single border:

Ex:

```
table {
    border-collapse: collapse;
}
```

```
table, th, td {
    border: 1px solid black;
}
```

>>If you only want a border around the table, only specify the border property for <table>:

```
table {
    border: 1px solid black;
}
```

Table Width and Height

Width and height of a table are defined by the width and height properties.

The example below sets the width of the table to 100%, and the height of the <th> elements to 50px:

```
<!DOCTYPE html>
<html>
<head>
<style>
table,td,th{
border:1px solid black;
}
table{
border-collapse:collapse;
width:100%
}
th{
height:50px;
}
</style>
</head>
<body>
<table>
      <tr>
            <th>First Name</th>
            <th>Last Name</th>
            <th>Saving</th>
      </tr>
      <tr>
            <td>Peter</td>
            <td>Giffin</td>
            <td>$100</td>
      </tr>
</table>
</body>
</html>
```

Horizontal Alignment

The text-align property sets the horizontal alignment (like left, right, or center) of the content in <th> or <td>.

By default, the content of <th> elements are center-aligned and the content of <td> elements are left-aligned.

The following example left-aligns the text in <th> elements:

```
<!DOCTYPE html>
<html>
<head>
<style>
table, td, th {
    border: 1px solid black;
}

table {
    border-collapse: collapse;
    width: 100%;
}

th {
    text-align: left;
}
</style>
</head>
<body>

<h2>The text-align Property</h2>
<p>This property sets the horizontal alignment (like left, right, or
center) of the content in th or td:</p>

<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>Peter</td>
    <td>Griffin</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>Lois</td>
    <td>Griffin</td>
    <td>$150</td>
  </tr>
  <tr>
    <td>Joe</td>
    <td>Swanson</td>
    <td>$300</td>
  </tr>
  <tr>
    <td>Cleveland</td>
    <td>Brown</td>
    <td>$250</td>
</tr>
</table>

</body>
</html>
```

Vertical Alignment

The vertical-align property sets the vertical alignment (like top, bottom, or middle) of the content in <th> or <td>.

By default, the vertical alignment of the content in a table is middle (for both <th> and <td> elements).

The following example sets the vertical text alignment to bottom for <td> elements:

Ex:

```
<!DOCTYPE html>
<html>
<head>
<style>
table, td, th {
    border: 1px solid black;
}

table {
    border-collapse: collapse;
    width: 100%;
}

td {
    height: 50px;
    vertical-align: bottom;
}
</style>
</head>
<body>

<h2>The vertical-align Property</h2>
<p>This property sets the vertical alignment (like top, bottom, or middle) of the content in th or td.</p>
<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>Peter</td>
    <td>Griffin</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>Lois</td>
    <td>Griffin</td>
    <td>$150</td>
  </tr>
  <tr>
    <td>Joe</td>
    <td>Swanson</td>
    <td>$300</td>
  </tr>
  <tr>
    <td>Cleveland</td>
    <td>Brown</td>
    <td>$250</td>
</tr>
</table>
```

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

Table Padding

To control the space between the border and the content in a table, use the padding property on <td> and <th> elements:

Example

```
th, td {
    padding: 15px;
    text-align: left;
}
```

Horizontal Dividers

First Name	Last Name	Savings
Peter	Griffin	$100
Lois	Griffin	$150
Joe	Swanson	$300

Add the border-bottom property to <th> and <td> for horizontal dividers:

Example

```
th, td {
    border-bottom: 1px solid #ddd;
}
```

Hoverable Table

Use the :hover selector on <tr> to highlight table rows on mouse over:

First Name	Last Name	Savings
Peter	Griffin	$100
Lois	Griffin	$150
Joe	Swanson	$300

Example

```
tr:hover {background-color: #f5f5f5}
```

Striped Tables

First Name	Last Name	Savings
Peter	Griffin	$100
Lois	Griffin	$150
Joe	Swanson	$300

For zebra-striped tables, use the nth-child() selector and add a background-color to all even (or odd) table rows:

Example

```
tr:nth-child(even) {background-color: #f2f2f2}
```

Table Color

The example below specifies the background color and text color of <th> elements:

First Name	Last Name	Savings
Peter	Griffin	$100
Lois	Griffin	$150
Joe	Swanson	$300

Example

```
th {
    background-color: #4CAF50;
    color: white;
}
```

Responsive Table

A responsive table will display a horizontal scroll bar if the screen is too small to display the full content:

```
First Name Last Name  Points     Points     Points     Points
      Points     Points     Points     Points     Points     Points
      Points     Points
Jill  Smith      50    50    50    50    50    50    50    50    50    50
      50    50
Eve   Jackson    94    94    94    94    94    94    94    94    94    94
      94    94
Adam  Johnson    67    67    67    67    67    67    67    67    67    67
      67    67
```

Add a container element (like) with overflow-x:auto around the <table> element to make it responsive:

```
<!DOCTYPE html>
<html>
<head>
<style>
table {
    border-collapse: collapse;
    width: 100%;
}

th, td {
    text-align: left;
    padding: 8px;
}

tr:nth-child(even){background-color: #f2f2f2}
</style>
</head>
<body>

<h2>Responsive Table</h2>
<p>A responsive table will display a horizontal scroll bar if the screen is too
small to display the full content. Resize the browser window to see the effect:</p>
<p>To create a responsive table, add a container element (like div) with <strong>overflow-x:auto</strong> around the table element:</p>

<div style="overflow-x:auto;">
  <table>
    <tr>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Points</th>
      <th>Points</th>
      <th>Points</th>
      <th>Points</th>
      <th>Points</th>
      <th>Points</th>
      <th>Points</th>
      <th>Points</th>
      <th>Points</th>
      <th>Points</th>
    </tr>
    <tr>
      <td>Jill</td>
      <td>Smith</td>
      <td>50</td>
      <td>50</td>
```

```
      <td>50</td>
      <td>50</td>
      <td>50</td>
      <td>50</td>
      <td>50</td>
      <td>50</td>
      <td>50</td>
      <td>50</td>
    </tr>
    <tr>
      <td>Eve</td>
      <td>Jackson</td>
      <td>94</td>
      <td>94</td>
      <td>94</td>
      <td>94</td>
      <td>94</td>
      <td>94</td>
      <td>94</td>
      <td>94</td>
      <td>94</td>
      <td>94</td>
    </tr>
    <tr>
      <td>Adam</td>
      <td>Johnson</td>
      <td>67</td>
      <td>67</td>
      <td>67</td>
      <td>67</td>
      <td>67</td>
      <td>67</td>
      <td>67</td>
      <td>67</td>
      <td>67</td>
      <td>67</td>
    </tr>
  </table>
</div>

</body>
</html>
------------------------
CSS Table Properties
Property   Description
border           Sets all the border properties in one declaration
border-collapse Specifies whether or not table borders should be
collapsed
border-spacing   Specifies the distance between the borders of adjacent
cells
caption-side     Specifies the placement of a table caption
empty-cells      Specifies whether or not to display borders and
background on empty cells in a table
table-layout     Sets the layout algorithm to be used for a tabl
```

CSS Layout - The display Property
The display property is the most important CSS property for controlling layout.

The display Property

The display Property specifes if/how an element is displayed.

Every HTML element has a default display value depending on what type of element it is. The default display value for most element is block or inline.

Block-level Elements

A block-level element always startes on a new line and takes up the full width available(stretches out to the left and right as far as it can).

Examples of block-level elements:

```
<div>
<h1> - <h6>
<p>
<form>
<header>
<footer>
<section>
```

Inline Elements

An inline element does not start on a new line and only takes up as much width as necessary.

This is an inline element inside a paragraph.

Examples of inline elements:

```
<span>
<a>
<img>
```

Display: none;

display: none; is commonly used with JavaScript to hide and show elements without deleting and recreating them. Take a look at our last example on this page if you want to know how this can be achieved.

The <script> element use display: none; as its default.

Override THe Default Display Value

As mentioned,every element has a default display value.However,you can override this.

Changing an inline element to a block element,or viceversa,can be useful for making the page look a specific eay, and still follow the web standards.

A Common example is making inline,<li>elements for horizantal menus:

Ex:

```
<style>
li {
    display: inline;
}
</style>
```

Note: Setting the display property of an element only changes how the element is displayed, NOT what kind of element it is. So, an inline element with display: block; is not allowed to have other block elements inside it.

Example

```
span {
    display: block;
}
```

Hide an Element - display:none or visibility:hidden?

Hiding an element can be done by setting the display property to none. The element will be hidden, and the page will be displayed as if the element is not there:

```
h1.hidden {
    display: none;
}
```

visibility:hidden; also hides an element.

However, the element will still take up the same space as before. The element will be hidden, but still affect the layout:

Example

```
h1.hidden {
    visibility: hidden;
}
```

CSS Display/Visibility Properties

Property	Description
display	Specifies how an element should be displayed
visibility	Specifies whether or not an element should be visible

CSS Layout - width and max-width

Using width, max-width and margin: auto;

As mentioned in the previous chapter; a block-level element always takes up the full width available (stretches out to the left and right as far as it can).

Setting the width of a block-level element will prevent it from stretching out to the edges of its container. Then, you can set the margins to auto, to horizontally center the element within its container. The element will take up the specified width, and the remaining space will be split equally between the two margins:
This element has a width of 500px, and margin set to auto.

Note: The problem with the above occurs when the browser window is smaller than the width of the element. The browser then adds a horizontal scrollbar to the page.

Using max-width instead, in this situation, will improve the browser's handling of small windows. This is important when making a site usable on small devices:
This element has a max-width of 500px, and margin set to auto.

Tip: Resize the browser window to less than 500px wide, to see the difference between the two divs!

Here is an example of the two divs above:
Example
```
<!DOCTYPE html>
<html>
<head>
<style>
div.ex1 {
    width:500px;
    margin: auto;
    border: 3px solid #73AD21;
}

div.ex2 {
    max-width:500px;
    margin: auto;
    border: 3px solid #73AD21;
}
</style>
</head>
<body>

<div class="ex1">This div element has width: 500px;</div>
<br>

<div class="ex2">This div element has max-width: 500px;</div>

<p><strong>Tip:</strong> Drag the browser window to smaller than 500px wide, to see the difference between
the two divs!</p>

</body>
</html>
```

CSS Layout - The position Property::::

The position property specifies the type of positioning method used for an element(static,relative,fixed or absolute).

The Position Property :

The Position Property specifies the type of positioning method used for an element.

There are four diff posistion values:

- static
- relative
- fixed
- absolute

Elements are then positioned using the top,bottom,left and right properties.However,these properties
will not work unless the position property is set first.They also work differently depending on the position value.

---#>Position::Static;
HTMl elements are positioned static by default.

Static positioned elements are not affected by the top,bottom,left,and right properties.

An element with position:static; is not positioned in any special way;itis always positioneed according to the normal flow of the page;

This element has position:static;

Herrrrrrre is the CSS that is USED:

```
<!doctype HTML>
<html>
<head>
<title>new</title>
<style>
div.st{
position:static;
border:3px solid red
padding:10px;
background-color:red;
color:yellow;
}
</style>
</head>
<body>
<div class="st">
This div element has position:static;
</div>
</body>
</html>
```

---
position: relative;

An element with position: relative; is positioned relative to its normal position.

Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position.
 Other content will not be adjusted to fit into any gap left by the element.

```
div.relative {
    position: relative;
    left: 30px;
    border: 3px solid #73AD21;
}
```

---

position: fixed;

An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.

A fixed element does not leave a gap in the page where it would normally have been located.

Notice the fixed element in the lower-right corner of the page. Here is the CSS that is used:

Ex:

```
div.st{
position:fixed;
border:3px solid red
padding:10px;
background-color:red;
color:yellow;
}
```

---

position: absolute;

An Element with position:absolute;is positioned relative to the nearest positioned ancestor(instead of positioned relative to the viewport,like fixed).

However; if an absolute positioned element has no positioned ancestors,it uses the document body,and moves along with page scrolling.

Note:A "positioned " element is one whose position is anything except static.

Here is an Ex:

```
div.relative {
    position: relative;
    width: 400px;
    height: 200px;
    border: 3px solid #73AD21;
}

div.absolute {
    position: absolute;
    top: 80px;
```

```
    right: 0;
    width: 200px;
    height: 100px;
    border: 3px solid #73AD21;
}
<div class="relative">This div element has position: relative;
  <div class="absolute">This div element has position: absolute;</div>
</div>
```

Overlapping An Elements:
When elements are positioned,they can overlap other elements.

The Z-index property specifies the stack order of an elemnt(which element should be placed in front of,or behind,the others).

An element can have a positive or negative stack order.

Ex

```
<!DOCTYPE html>
<html>
<head>
<style>
img{
postion:absolute;
left:0px;
top:0px;
z-index:-1;}

</style>
</head>
<body>
<h1>This is heading</h1>
<img src="file:///C:/Users/danda.satish/Desktop/Alvin/1.jpg">
<p>BEcause the image has a z index of -1, it will be placed behind the text</p>
</body>
</html>
```

An element with greater stack order is always in front of an element with a lower stack order.

Note: If two positioned elements overlap without a z-index specified, the element positioned last in the HTML code will be shown on top.

Positioning Text in an Image

How to position text over an image:

```
<!DOCTYPE html>
<html>
<head>
<style>
.con{
position:relative;
}
.top{
position:absolute;
top:25px;
```

```
left:106px;
font-size:28px;
color:blue;
font-family:Times of new Roman;
}
.middle{
postion:absolute;
left:0%;
top:50%;
width:100%;
text-align:center;
color:red;
font-size:28px;
font-family:Times of new Roman;
}
.right{
postion:absolute;
top:8pxpx;
right:16px;
font-size:times of new roman;
color:green;
font-size:28px;}
img{
width:100%;
height:auto;
opacity:0.3;}

</style>
</head>
<body>
<h2>Image  heading</h2>
<div class="con">
<img src="file:///C:/Users/danda.satish/Desktop/Alvin/1.jpg" alt =
"alvin">
<div class="top">Simon</div>
<div class="middle">ALVIN</div>
<div class="right">Theodare</div>
</div>
</body>
</html>
---------------------------------
```