II UNIT

CSS(CASCADING STYLE SHEET)

INTRODUTION:

To apply the style and lay outs to the web pages — for example, to alter the font, colour, size and spacing of your content, split it into multiple columns, or add animations and other decorative features.For that purpose we can use "Cascading Style Sheets(CSS)".

CSS Versions

Cascading Style Sheets, level 1 (CSS1) was 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 was became a W3C recommendation in May 1998 and builds on CSS1.This version adds support for media-specific style sheets e.g. printers and aural devices, downloadable fonts, element positioning and tables.

CSS3 was became a W3C recommendation in June 1999 and builds on older versions CSS. It has divided into documentations is called as Modules and here each module having new extension features defined in CSS2.

Advantages of CSS

- **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 comparisonto 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 it is a good idea to start using CSS inall the HTML pages to make them compatible to future browsers.

- **Offline Browsing** – CSS can store web applications locally with the help of an offline catche. Using of this, we can view offline websites. The cache also ensures faster loading and better overall performance of the website.
- **Platform Independence** –The Script offer consistent platform independence and can support latest browsers as well.

Styling HTML with CSS

- ➢ 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 allat once.

->CSS can be added to HTML elements in 3 ways:

1. **Inline CSS** -

Def: Using the <style> attribute with in HTML elements or HTMLtagsis called

"Inline CSS".

- An Inline CSS is used to apply a unique style to a single HTML element.
- This example sets the text color of the <h1> element to blue:

Example:

```
<!DOCTYPE html>

<html>

<body>

<h1 style="color:blue;">This is a Blue Heading</h1>

</body>

</html>
```

2. **Internal** CSS

Def: Using a <style> element in the <head> section is called "InternalCSS".

- An Internal CSS is used to define a style for "a single HTML page".
- An internal CSS is defined in the <head> section of an HTML page, withina <style> element.

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
        body {background-color: powderblue;}h1
             {color: blue;}
        p  {color: red;}
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

3. External CSS- Using a <style> element in an external CSS file is called“External

CSS”.

- An external style sheet is used to define the style for “many HTML pages”.
- **With an external style sheet, you can change the look of an entire website, by changing one file.**

The most common way to add CSS, is to keep the styles in separate CSS files. However, here we will use inline and internal styling, because this is easier to demonstrate, and easier for you to try it yourself.

Example:

To use an external style sheet, add a link to it in the <head> section of theHTML page:

```
<!DOCTYPE html>
<html>
<head>
       <link rel="Stylesheet" href="Styles.css">
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

- An external style sheet can be written in any text editor. The file must notcontain any HTML code, and must be saved with a .css extension.

Here is how the **"Styles.css"** looks:

```
body {
    background-color: red;
}
h1 {
    color: blue;
}
p {
    color: red;
}
```

Style Specification Formats:

- Format depends on the level of the style sheet
- Inline:
  - Style sheet appears as the value of the style attribute
  - General form:

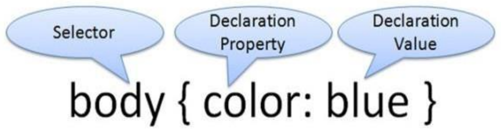
```
Style="Property1:Value1;

        Property2:Value2; .........

Propertyn:Valuen;
```

- Document-level (or) Internal-level:
  - Style sheet appears as a list of rules that are the content ofa <style> tag
  - The <style> tag must include the type attribute, set to "text/css"
  - General form:

```
<style type="text/css">

            Rule list

</style>
```

- External style sheets
  - Form is a list of style rules, as in the content of a <style> tag fordocument-level style sheets
  - General Form:
    selector {list of property/values}
    - Each property/value pair has the form:
      property: value
    - Pairs are separated by *semicolons*, just as in the value ofa <style> tag
  - Comments in the rule list must have a different form - use Comments (/*...*/)

Example:



CSS Selectors

A CSS selector is the part of a CSS rule set that actually selects the content to which we want to apply the style. Let’s look at all the different kinds of selectorsavailable, with a brief description of each.

(i) Universal Selector

The “*universal selector”* works like a wild card character, selecting all elements ona page. Every HTML page is built on HTML tags. Each set of tags represents an element on the page.

-> CSS example, which uses the universal selector:

```
<style>

        * {

        color: green; font-

        size: 20px;

        line-height: 25px;

        }

</style>
```

Note: 1)The three lines of code inside the curly braces (color, font-size,and line-

height) will apply to all elements on the HTML page.

2) As seen here, the universal selector is declared using an “asterisk(*)”.

3) You can also use the universal selector in combination with otherselectors.

(ii) Element Type Selector

Also referred to simply as a “type selector,” this selector must match one or moreHTML elements of the same name.

Example: A selector of <ul>would match all HTML unordered lists,or

<ul>elements. (i.e) The following example uses an element type selector to match

all <ul>elements:

```
ul {

list-style: none; border:

solid 1px #fff;

}
```

To put this in some context, here’s a section of HTML to which we’ll apply theabove CSS:

```
<ul>

<li>Fish</li>

<li>Apples</li>

<li>Cheese</li>

</ul>

<div class="example">

<p>Example paragraph text.</p>

</div>
<ul>

<li>Water</li>

<li>Juice</li>

<li>Maple Syrup</li>

</ul>
```

There are three main elements making up this part of the page:

Two <ul> elements and a <div>. The CSS will apply only to the two <ul> elements, and not to the <div>. Were we to change the element type selector to use <div> instead of <ul>, then the styles would apply to the <div> and not to the two <ul> elements.

Also note that the styles will not apply to the elements inside the <ul> or <div>elements. That being said, some of the styles may be inheritedby those inner elements.

(iii) ID Selector

An ID selector is declared using **"a hash, or pound symbol (#)"** precedinga string of characters. The string of characters is defined by the developer.

This selector matches any HTML element that has an ID attribute with thesame value as that of the selector, but minus the hash symbol.

Here's an example:

```
#container {
  width: 960px;
  margin: 0 auto;
}
```

This CSS uses an ID selector to match an HTML element such as:

```
<div id="container"></div>
```

In this case, the fact that this is a <div>element doesn't matter—it couldbe any kind of HTML element. As long as it has an ID attribute with a value
of container, the styles will apply.

An ID element on a web page should be unique. That is, there should onlybe a single element on any given page with an ID of container.

(iv) Class Selector

The class selector is the most useful of all CSS selectors. **It's declared witha dot preceding a string of one or more characters.**

The class selector also matches all elements on the page that have their classattribute set to the same value as the class, minus the dot.

Take the following rule set:

```
.box {
  padding: 20px;
  margin: 10px;
  width: 240px;
}
```

These styles will apply to the following HTML element:

```
<div class="box"></div>
```

The same styles will also apply to any other HTML elements that have a class attribute with a value of box. Having multiple elements on a single page withthe same class attribute is beneficial, because it allows you to reuse styles, and avoid needless repetition.

Another reason the class selector is a valuable is that HTML allows multipleclasses to be added to a single element. This is done by separating the classes in the HTML class attribute using spaces. Here’s an example:

```
<div class=”box box-more box-extended”></div>
```

(v) Descendant Combinator

The descendant selector or,the descendant combinator combine two or moreselectors. so we could be more specific in selection method.

For example:

```
#container .box {
  float: left;
  padding-bottom: 15px;
}
```

This declaration block will apply to all elements that have a class of box thatare inside an element with an ID of container. It’s worth noting that
the .boxelement doesn’t have to be an immediate child: there could be anotherelement wrapping .box, and the styles would still apply.

Look at the following HTML:

```
<div id="container">
<div class="box"></div>


<div class="box-2"></div>
</div>


<div class="box"></div>
```

If we apply the CSS in the previous example to this section of HTML, the
only element that’ll be affected by those styles is the first <div>element that hasa class of box.

The <div> element that has a class of box-2 won’t be affected by the styles. Similarly, the second <div>element with a class of boxwon’t be affectedbecause it’s not inside an element with an ID of container.

(vi) <u>Child Combinator</u>

A selector that uses the childcombinator is similar to a selector that uses adescendant combinator, except it only targets immediate childelements:

```
#container > .box {
   float: left;
   padding-bottom: 15px;
}
```

This is the same code from the descendant combinator example, but instead of a space character, we’re using the greater-than symbol (or right angle bracket.)

In this example, the selector will match all elements that have a class of box andthat are immediate children of the #containerelement. That means, unlike thedescendant combinator, there can’t be another element wrapping .box—it has tobe a direct child element.

Here’s an HTML example:

```
<div id="container">
<div class="box"></div>

<div>
<div class="box"></div>
</div>
</div>
```

In this example, the CSS from the previous code example will apply only to the first <div>element that has a class of box. As you can see, the second <div>element with a class of boxis inside another <div>element. As aresult, the styles will not apply to that element, even though it too has a class of box.

Again, selectors that use this combinator can be somewhat restricting, but they cancome in handy—for example, when styling nested lists.

(vii) General Sibling Combinator

A selector that uses a general sibling combinator matches elements based onsibling relationships. That is to say, the selected elements are beside each other in the HTML.

```
h2 ~ p {
  margin-bottom: 20px;
}
```

->This selector is declared using the tilde character (~).

In this example, all paragraph elements (<p>) will be styled with the specified rules, but only if they are siblings of <h2> elements. There could beother elements in between the <h2>and <p>, and the styles would still apply.

Let’s apply the CSS from above to the following HTML:

```
<h2>Title</h2>
<p>Paragraph example.</p>
<p>Paragraph example.</p>
<p>Paragraph example.</p>
<div class="box">
<p>Paragraph example.</p>
</div>
```

In this example, the styles will apply only to the first three paragraph elements. The last paragraph element is not a sibling of the <h2>element becauseit sits inside the <div>element.

(viii) Adjacent Sibling Combinator

A selector that uses the adjacent sibling combinator uses the plus symbol(+), and is almost the same as the general sibling selector.

The difference is that the targeted element must be an immediate sibling, notjust a general sibling.

Example:

```
p + p {
  text-indent: 1.5em;
  margin-bottom: 0;
  }
```

This example will apply the specified styles only to paragraph elements thatimmediately follow other paragraph elements. This means the first paragraph element on a page would not receive these styles. Also, if another element appeared between two paragraphs, the second paragraph of the two wouldn't havethe styles applied.

So, if we apply this selector to the following HTML:

```
<h2>Title</h2>
<p>Paragraph example.</p>
<p>Paragraph example.</p>
<p>Paragraph example.</p>

<div class="box">
<p>Paragraph example.</p>
<p>Paragraph example.</p>
</div>
```

…the styles will apply only to the second, third, and fifth paragraphs in this sectionof HTML.

(ix) <u>Attribute Selector</u>

The attribute selector targets elements based on the presence and/or value ofHTML attributes, and is declared using square brackets:

```
input[type="text"] {
  background-color: #444;
  width: 200px;
}
```

There should not be a space before the opening square bracket unless you intend to use it along with a descendant combinator.

<u>Example:</u>

The above CSS would match the following element:

```
<input type="text">
```

- But it wouldn't match to the following code:

```
<input type="submit">
```

The attribute selector can also be declared using just the attribute itself, with novalue, like this:

```
input[type] {
  background-color: #444;
  width: 200px;
}
```

This will match all input elements with an attribute of type, regardless of the value.

(x) <u>Pseudo-class</u>

A pseudo-class uses **"a colon"** character to identify a pseudo-state that anelement in Html page.

<u>Example:</u>

```
a:hover {
  color: red;
}
```

In this case, the pseudo-class portion of the selector is the :hover(Select and style a link when you mouse over it) part. Here we've attached this pseudo-class to all anchor elements (elements). This means that whenthe user hovers their mouse over an element, the color property for that element will change to red.

This type of pseudo-class is a dynamic pseudo-class, because it occurs onlyin response to user interaction—in this case, the mouse moving over the targeted element.

<u>FONT PROPERTIES IN CSS3:</u>

The font property in CSS is a shorthand property that combines all the followingsub-properties in a single declaration.

```
body {
font: normal small-caps  16px 1.4 Georgia;
}
/* is the same as:

body {
font-family: Georgia;
line-height: 1.4;
font-weight: normal; font-
variant: small-caps;font-
size: 16px;
}
 */
```

There are Six Font sub-properties. They are:

- **font-style**: It makes the text appear italicised or oblique.

Syntax: font-style: <value>;

Possible Values:

- normal
- italic
- oblique
- inherit

- **font-variant**: It changes target text to small caps.

Syntax: font-variant: <value>;

Possible Values:

- normal
- small-caps
- inherit

- **font-weight**: It sets the weight or the thickness of the font.

Syntax: font-weight: <value>;

Possible Values:

- normal
- bold
- bolder
- lighter
- 100, 200, 300, 400, 500, 600, 700, 800, 900
- inherit

- **font-size:**It sets the height of the font.

Syntax: font-size: <value>;

Possible Values:

- xx-small
- x-small
- small
- medium
- large
- x-large
- xx-large
- smaller, larger
- percentage
- inherit

- **line-height:** It defines the amount of space above and below inlineelements.
  Syntax:line-height: <value>;

Possible Values:

- normal
- number (font-size multiplier)
- percentage

- **font-family:** It definies the font that is applied to the element. **Syntax:** font-family: [[<family-name> | <generic-family>]]* [<family-name> | <generic-family>];

<generic-family> names:

- sans-serif
- serif
- monospace
- cursive
- fantasy

<font-family> names:

- caption
- icon
- menu
  message-box
- small-caption
- status-bar
- string

Example:

```
p.normal {

font-style: normal;

}
```

```
p.italic {

font-style: italic;

font-size: 40px;

font-weight: bold;

line-height: 90%;

font-family:Copperplate Gothic;

}


p.oblique {

font-style: oblique;

font-variant:small-caps;

}

</style>

</head>

<body>

<p class="normal">This is a paragraph in normal style.</p>

<p class="italic">This is a paragraph in italic style.</p>

<p class="oblique">This is a paragraph in oblique style.</p>

</body>

</html>
```

CSS List Properties

In CSS,there are two main types of lists:

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

The CSS list properties allow you to:

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

1,2)Different List Item Markers for Ordered and UnorderedLists:

We can use "**list-style-type"** property to specify the type of list itemmarker.

Example:

```
<!DOCTYPE html>

<html>

<head>

<style>
ul.a {

list-style-type: circle;

}

ul.b {

list-style-type: square;

}

ol.c {

list-style-type: upper-roman;

}

ol.d {

list-style-type: lower-alpha;

}
```

```
</style>

</head>

<body>

<p>Example of unordered lists:</p>

<ul class="a">

<li>Coffee</li>

<li>Tea</li>

<li>Coca Cola</li>

</ul>
<ul class="b">

<li>Coffee</li>

<li>Tea</li>

<li>Coca Cola</li>
</ul>

<p>Example of ordered lists:</p>

<ol class="c">

<li>Coffee</li>

<li>Tea</li>

<li>Coca Cola</li>

</ol>

<ol class="d">

<li>Coffee</li>

<li>Tea</li>

<li>Coca Cola</li>

</ol>

</body>

</html>
```

3) Set an image as the list item marker:

We can use the **“**list-style-image”property to specify an image as thelist item marker.

Example:

```
<!DOCTYPE html>

<html>

<head>

<style>

ul {

list-style-image: url('Imagename');

}

</style>

</head>

<body>

<ul>

<li>Coffee</li>

<li>Tea</li>

<li>Coca Cola</li>

</ul>

</body>

</html>
```

Position The List Item Markers:

We can use **“**list-style-position” property to specify whether the list-item markers should appear inside or outside the content flow.

Example:

```
<!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>
```

4) Styling List With Colors:

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

Anything added to the <ol> or <ul> tag, affects the entire list, whileproperties added to the <li> tag will affect the individual list items.

Example:

```
<!DOCTYPE html>

<html>

<head>

<style>

ol {

background: green;

padding: 20px;

}


ul {

background: yellow;

padding: 20px;

}

ol li { background:

red; padding: 5px;

margin-left: 35px;

}


ul li { background:

pink;margin: 5px;

}

</style>
</head>
```

```
<body>
<h1>Styling Lists With Colors:</h1>
<ol>
<li>Coffee</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>


<ul>
<li>Coffee</li>
<li>Tea</li>
<li>Coca Cola</li>
</ul>
</body>
</html>
```

Color Properties in CSS:

CSS color properties allows us to color the "**Background and Foreground Color"** on a Web Page. We can set CSS color on text, backgrounds, borders, and other parts of elements in a document.

I. CSS Background Color

(i) Set CSS body background color:

You can define the background color of a webpage by specifying its body "**background-color**" property.

Example:

```
body {background-color:#C0C0C0;}
```

(ii) Set CSS Paragraph background color :

Example:

```
body {background-color:#C0C0C0;}
```

a. p {background-color:#FFFFFF;}

(iii) Set CSS div back color :

```
body {background-color:#C0C0C0;}
p    {background-color:#FFFFFF;}
div  {background-color:#00FFFF;}
```

2. CSS Foreground Color

(i) Change CSS text color :

When we want to change the color of a text (foreground color) in an HTMLdocument the term color is used to specify the CSS property.

Example:

```
body { color: #800000 }
```

(ii) Change the Font Color with CSS :

When you set foreground color , actually the font color will change.

Example:

```
h1 { color: green; }
```

The above code changes the font color inside the h1 tag.

3. BorderColor

(i) Set CSS border-color:

The **"border-color"** property specifies the border color for each side of the box.

Example:

```
P{
border-width: 2px;
border-color:red;
border-style: solid;or
border:2px solid red;
                    }
```

You can specify border color to each side specifically.

<u>Example:</u>

```
<html>
<head>
<style type="text/css">
                    p{
                              border-width: 8px;
                              border-style: solid;
                              border-top-color: red;
                              border-right-color: green;
                              border-bottom-color: purple;
                              border-left-color: blue;
                    }
          </style>
</head>
<body>
```

<u>Color Keywords</u>

The first and easiest way to specify a color is using one of the 17 predefinedcolor ***keywords*** specified in CSS2.1.

Color	Keyword	Hex Value
⬛	Black	#000000
▩	Gray	#808080
▨	Silver	#c0c0c0
⬜	White	#ffffff
🟥	Maroon	#800000

🟥	Red	#ff0000
🟪	Purple	#800080
🟪	Fuchsia	#ff00ff
🟩	Green	#008000
Color	**Keyword**	**Hex Value**
🟩	Lime	#00ff00
🟫	Olive	#808000
🟨	Yellow	#ffff00
🟦	Navy	#000080
🟦	Blue	#0000ff
🟦	Teal	#008080
🟦	Aqua	#0000ff
🟧	Orange	#ffa500

Allignment of Text:

 The “text-align”property in CSS is used for aligning the innercontent of a Html element.

Example:

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

 These are the traditional values for text-align:

- left - The default value.Text aligns from the left side.
- right – Content or Text aligns from the right side.
- center - Content centers between the left and right edges. White space onthe left and right sides of each line should be equal.
- justify - Content spaces out such that as many blocks fit onto one line aspossible and the first word on that line is along the left edge and the last word is along the right edge.
- inherit - The value will be whatever the parent element's is.

Background Images

The “background-image”property specifies an image to use as thebackground of an element.

 By default, the image is repeated so it covers the entire element.

->The background image for a page can be set like this:

Example:

```
<!DOCTYPE html>

<html>

<head>

<style>

body {

background-image: url("paper.gif");

}

</style></head>
```

```
<body>

<h1>Hello World!</h1>

<p>This page has an image as the background!</p>

</body>

</html>
```

Div and Span Tags

Div tags and span tags are very common **HTML** elements that are growingin popularity due to their flexibility.

HTML div tags are more specific for organizational tasks like setting up thelayout of your page, which are preferred over tables because of their fluid like nature. -

Span tags are used more to permit customization of text and are often used inside other HTML elements to customize a certain piece of content from the rest.

The internal layout of this page is created with div elements.

HTML Div Element

The div element is a block level element, much like the paragraph tag.

Example

```
<div> I am a div!</div>
<div> Me too!</div>
```

HTML Span Element

- The HTML span element is an inline element, which means that it canbe used inside a block element and not create a new line.
- If we want to highlight some text inside a paragraph, wrap that text ina span tag.
- Span tags are often used to incorporate a specific CSS style to differentiate certain parts of content.

Example

```
<p> Something here is <span style="color:#900;"> special</span> , but whichone?</p>
```

Example:

```
<html>

<head>

<title>Span and Div Tags</title>

<style>

div

   {

background-color:yellow;border:2px

solid red;

   }

p

   {

color:green;

   }

</style>

</head>

<body>

<div>

<h1>Hi,Good Morning</h2>

<p> Welcome<span style="color:red;text-decoration:underline"> to</span>
Paragraph</p>

</div>

</body>

</html>
```

CSS Gradient:

CSS gradient is used to display smooth transition within two or more specified colors.

Why CSS Gradient

These are the following reasons to use CSS gradient.

- You don't have to use images to display transition effects.
- The download time and bandwidth usage can also be reduced.
- It provides better look to the element when zoomed, because the gradient is generated by the browser.

There are two types of gradient in CSS3.

1. Linear gradients
2. Radial gradients

1) CSS Linear Gradient

The CSS3 linear gradient goes up/down/left/right and diagonally. To create a CSS3 linear gradient, you must have to define two or more color stops. The color stops are the colors which are used to create a smooth transition. Starting point and direction can also be added along with the gradient effect.

1. background: linear-gradient (direction, color-stop1, color-stop2 ....);

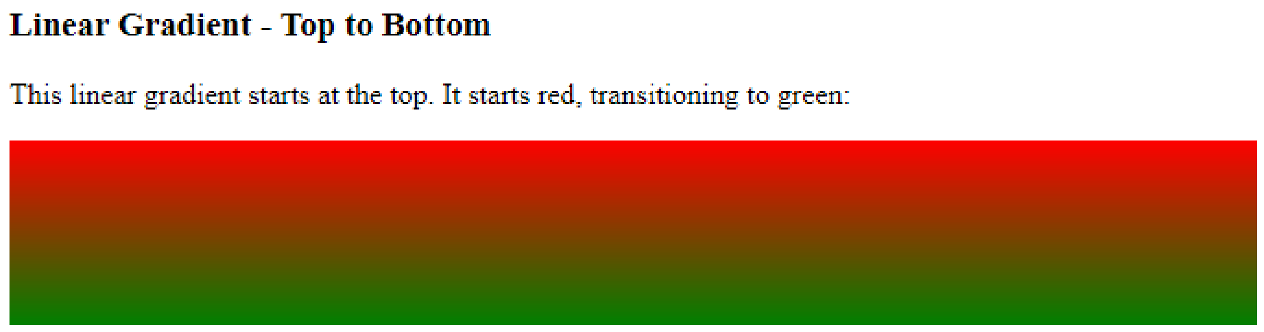
(i) CSS Linear Gradient: (Top to Bottom)

Top to Bottom Linear Gradient is the default linear gradient. Let's take an example of linear gradient that starts from top. It starts red and transitions to green.

```
<!DOCTYPE html>
<html>

<head>

<style>

#grad1 {

   height: 100px;

   background: -webkit-linear-gradient(red, green); /* For Safari 5.1 to 6.0 */
```

```
    background: -o-linear-gradient(red, green); /* For Opera 11.1 to 12.0 */
    background: -moz-linear-gradient(red, green); /* For Firefox 3.6 to 15 */
    background: linear-gradient(red, green); /* Standard syntax (must be last) */
}
</style>
</head>
<body>
<h3>Linear Gradient - Top to Bottom</h3>
<p>This linear gradient starts at the top. It starts red, transitioning to green:</p>
<div id="grad1"></div>
</body>
</html>
```

Output:



```
Linear Gradient - Top to Bottom

This linear gradient starts at the top. It starts red, transitioning to green:
```

(ii) CSS Linear Gradient: (Left to Right)

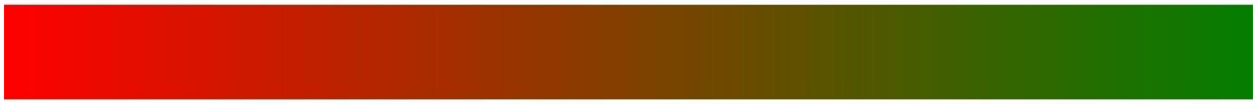
The following example shows the linear gradient that starts from left and goes to right. It starts red from left side and transitioning to green.

```
<!DOCTYPE html>
<html>
<head>
<style>
#grad1 {
   height: 100px;
   background: -webkit-linear-gradient(left, red, green); /* For Safari 5.1 to 6.0 */

   background: -o-linear-gradient(right, red, green); /* For Opera 11.1 to 12.0 */

   background: -moz-linear-gradient(right, red, green); /* For Firefox 3.6 to 15 */

   background: linear-gradient(to right, red , green); /* Standard syntax (must be last) */

}

</style>
</head>

<body>

<h3>Linear Gradient - Left to Right</h3>

<p>This linear gradient starts at the left. It starts red, transitioning to green:</p>

<div id="grad1"></div>

</body>

</html>
```

Output:

Linear Gradient - Left to Right

This linear gradient starts at the left. It starts red, transitioning to green:



(iii) CSS Linear Gradient: (Diagonal)

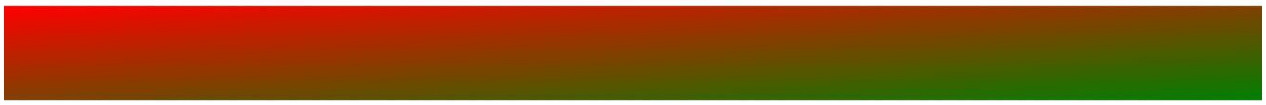
If you specify both the horizontal and vertical starting positions, you can make a linear gradient diagonal.

```
<!DOCTYPE html>

<html>

<head>

<style>

#grad1 {

   height: 100px;

   background: -webkit-linear-gradient(left top, red , green); /* For Safari 5.1 to 6.0 */

   background: -o-linear-gradient(bottom right, red, green); /* For Opera 11.1 to 12.0 */

   background: -moz-linear-gradient(bottom right, red, green); /* For Firefox 3.6 to 15 */

   background: linear-gradient(to bottom right, red , green); /* Standard syntax (must be last) */

}

</style>

</head>

<body>

<h3>Linear Gradient - Diagonal</h3>

<p>This linear gradient starts at top left. It starts red, transitioning to green:</p>

<div id="grad1"></div>
</body></html>
```

Output:

Linear Gradient - Diagonal

This linear gradient starts at top left. It starts red, transitioning to green:



2) CSS Radial Gradient

You must have to define at least two color stops to create a radial gradient. It is defined by its center.

1. background: radial-gradient(shape size at position, start-color, ..., last-color);

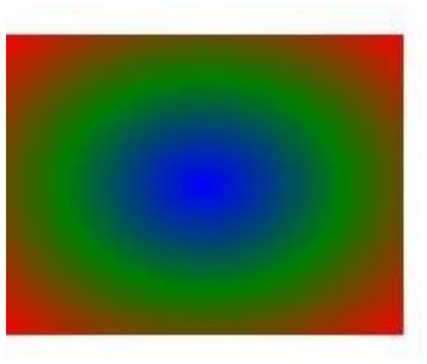
(i) CSS Radial Gradient: (Evenly Spaced Color Stops)

Evenly spaced color stops is a by default radial gradient. Its by default shape is eclipse, size is farthest- carner, and position is center.

```
<!DOCTYPE html>

<html>

<head>

<style>

#grad1 {

   height: 150px;

   width: 200px;

   background: -webkit-radial-gradient(blue, green, red); /* Safari 5.1 to 6.0 */

   background: -o-radial-gradient(blue, green, red); /* For Opera 11.6 to 12.0 */

   background: -moz-radial-gradient(blue, green, red); /* For Firefox 3.6 to 15 */

   background: radial-gradient(blue, green, red); /* Standard syntax (must be last) */

}
</style>

</head>

<body>

<h3>Radial Gradient - Evenly Spaced Color Stops</h3>

<div id="grad1"></div>

</body>

</html>
```

Output:

Radial Gradient - Evenly Spaced Color Stops



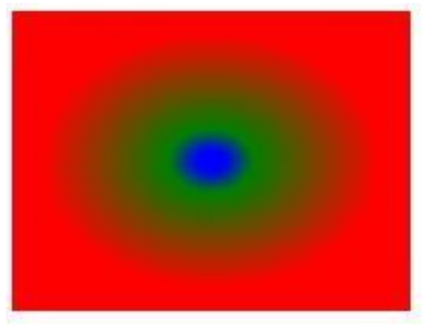
(ii) Radial Gradient: (Differently Spaced Color Stops)

```
<!DOCTYPE html>
<html>
<head>
<style>
#grad1 {
   height: 150px;
   width: 200px;
   background: -webkit-radial-gradient(blue 5%, green 15%, red 60%); /* Safari 5.1 to 6.0 */
   background: -o-radial-gradient(blue 5%, green 15%, red 60%); /* For Opera 11.6 to 12.0 */
   background: -moz-radial-gradient(blue 5%, green 15%, red 60%); /* For Firefox 3.6 to 15 */
   background: radial-
gradient(blue 5%, green 15%, red 60%); /* Standard syntax (must be last) */
}
</style>
</head>
```

```
<body>

<h3>Radial Gradient - Differently Spaced Color Stops</h3>

<div id="grad1"></div>

</body>

</html>
```

Output:

Radial Gradient - Differently Spaced Color Stops



CSS SHADOWS:

With CSS you can add shadow to text and to elements.

In these chapters you will learn about the following properties:

- text-shadow
- box-shadow

CSS Text-shadow:

As its name implies, this CSS property adds shadows to the text. It accepts the comma-separated list of shadows that applied to the text. It's default property is none. It applies one or more than one text-shadow effect on the element's text content.

Let's see the syntax of text-shadow property.

Syntax

1. text-shadow: h-shadow v-shadow blur-radius color| none | initial | inherit;

Values

h-shadow: It is the required value. It specifies the position of the horizontal shadow and allows negative values.

v-shadow: It is also the required value that specifies the position of the vertical shadow. It does not allow negative values.

blur-radius: It is the blur-radius, which is an optional value. Its default value is 0.

color: It is the color of the shadow and also an optional value.

none: It is the default value, which means no shadow.

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

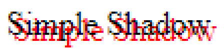
inherit: It simply inherits the property from its parent element.

Let's understand it by using some illustrations.

Example- Simple shadow

```
<!DOCTYPE html>
  <html>
<head>
  <title> font-weight property </title>
  <style>
    p.simple{
      text-shadow: 3px 3px red;
    }
  </style>
</head>
<body>
  <p class="simple">
    Simple Shadow
  </p>
</body>
</html>
```

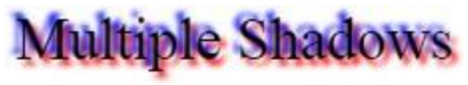
OUTPUT:


Simple Shadow

Example- Multiple Shadows:

```
<!DOCTYPE html>
 <html>
<head>
   <title> font-weight property </title>
   <style>
     p.multi{
       text-shadow: -3px -3px 3px blue, 3px 3px 3px red;
       font-size:30px;
       text-align:center;
     }
   </style>
</head>

<body>
   <p class="multi">
   Multiple Shadows
   </p>

</body>
</html>
```

OUTPUT:



Box-shadow CSS:

It is used to add shadow-like effects around the frame of an element.

Syntax:

box-shadow: h-offset v-offset blur spread color

|inset|inherit|initial|none;Let's understand property values.

h-offset: It horizontally sets the shadow position. Its positive value will set the shadow to the right side of the box. Its negative value is used to set the shadow on the left side of the box.

v-offset: Unlike the **h-offset**, it is used to set the shadow position vertically. The positive value in it sets the shadow below the box, and the negative value sets the shadow above of the box

blur: As its name implies, it is used to blur the box-shadow. This attribute is optional.

spread: It sets the shadow size. The spread size depends upon the spread value.

color: As its name implies, this attribute is used to set the color of the shadow. It is an optional attribute.

inset: Normally, the shadow generates outside of the box, but by using inset, the shadow can be created within the box.

initial: It is used to set the property of the box-shadow to its default value.

inherit: it is inherited from its parent.

none: It is the default value that does not include any shadow property.

EXAMPLE:

```
<!DOCTYPE html>

<html>

<head>

<title>Example of CSS3 Multiple Box Shadow Effects</title>

<style>

   .box{

                   width: 200px;

                   height: 150px;
```

```
                    background: #000;

                    box-shadow: 10px 10px 10px red, 20px 20px 30px yellow;

          }

</style>

</head>

<body>

   <div class="box"></div>

</body>

</html>
```

OUTPUT: