CSS

Cascading Style Sheets

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

Advantages of CSS

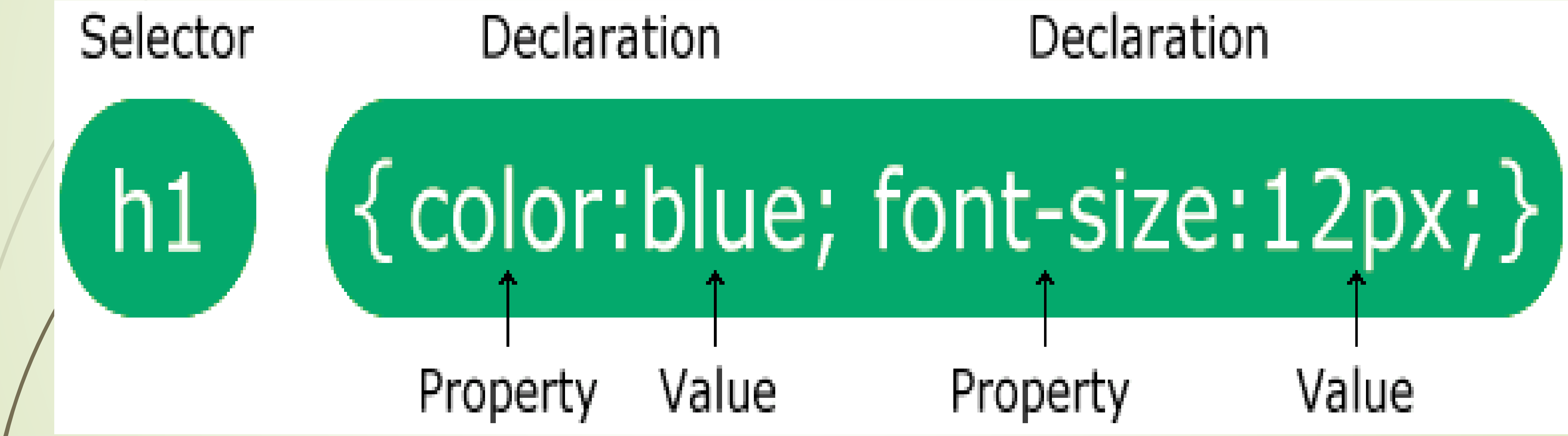
- **CSS saves time** - You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- **Pages load faster** - If you are using CSS, you do not need to write HTML tag attributes every time. Just write 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 comparison to HTML attributes.
- **Multiple Device Compatibility** - Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
- **Global web standards** - Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

CSS - Syntax

A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of three parts −

- **Selector** − A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.
- **Property** − A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be *color*, *border* etc.
- **Value** − Values are assigned to properties. For example, *color* property can have value either *red* or *#F1F1F1* etc.

CSS Syntax



```
h1 {color:blue; font-size:12px;}
```

CSS Syntax

- 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.
- Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

EXAMPLE

```
<html>
    <head>
        <style>
            .h1tag{color : red;}
            .h2tag{color : blue;}
        </style>
    </head>
    <body>
        <h1 class="h1tag">head 1</h1>
        <h2 class="h2tag">head 2</h2>
    </body>
</html>
```

CSS Selectors

CSS selectors are used to "find" (or select) the HTML elements you want to style.

CSS element Selector

The element selector selects HTML elements based on the element name.

Eg:

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

CSS Selectors

```
<html>
<head><style>
p {
        text-align: center;
        color: red;
      }
h1 {
        text-align: center;
        color: blue;
}
</style></head>
<body>
<p>para1</p>
<h1>heading1</h1>
</body><html>
```

CSS Universal Selector

The universal selector (*) selects all HTML elements on the page.

Eg:

```
* {
   text-align: center;
   color: blue;
}
```

CSS Descendant Selectors

Suppose you want to apply a style rule to a particular element only when it lies inside a particular element. As given in the following example, style rule will apply to <p> element only when it lies inside <div> tag.

Eg:

```
<html>
    <head>
        <style>
            div p{color : red;}
        </style>
    </head>
    <body>
        <div><p>Descendent selectors</p></div>
    </body>
</html>
```

CSS class Selector

The class selector selects HTML elements with a specific class attribute.

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

Eg:

```
<html>
    <head>
        <style>
            .h1tag{color : red;}
            .h2tag{color : blue;}
        </style>
    </head>
    <body>
        <h1 class="h1tag">head 1</h1>
        <h2 class="h2tag">head 2</h2>
    </body>
</html>
```

In this example only <p> elements with class="center" will be red and center-aligned:

Eg:

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

You can apply more than one class selectors to given element.

Eg:

```
<html>
    <head>
        <style>
            p.center {text-align: center;}
            p.large { font-size: 300%;}
        </style>
    </head>
    <body>
        <h1 class="center">This heading will not be affected</h1>
        <p class="center large">This paragraph will be red, center-aligned,
and in a large font-size.</p>
    </body>
</html>
```

CSS id Selector

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

Eg:

```
<html>
    <head>
        <style>
            #h1tag{color : red;}
        </style>
    </head>
    <body>
        <h1 id="h1tag">head 1</h1>
     </body>
</html>
```

CSS Attribute Selectors

You can also apply styles to HTML elements with particular attributes. The style rule below will match all the input elements having a type attribute with a value of *text* −

Eg:

```
input[type = "text"] {
    color: #FF0000;
}
```

The advantage to this method is that the <input type = "submit" /> element is unaffected, and the color applied only to the desired text fields.

Multiple Style Rules

You may need to define multiple style rules for a single element. You can define these rules to combine multiple properties and corresponding values into a single block .

Eg:

```
h1 {
        color: #36C;
        font-weight: normal;
        letter-spacing: .4em;
        margin-bottom: 1em;
        text-transform: lowercase;
}
```

Here all the property and value pairs are separated by a **semicolon (;)**. You can keep them in a single line or multiple lines. For better readability, we keep them in separate lines.

Grouping Selectors

You can apply a style to many selectors if you like. Just separate the selectors with a comma.

Eg:

```
h1, h2, h3 {
    color: #36C;
    font-weight: normal;
    letter-spacing: .4em;
    margin-bottom: 1em;
    text-transform: lowercase;
}
```

CSS - Inclusion

There are four ways to associate styles with your HTML document.

Inline CSS

Embedded CSS

External CSS

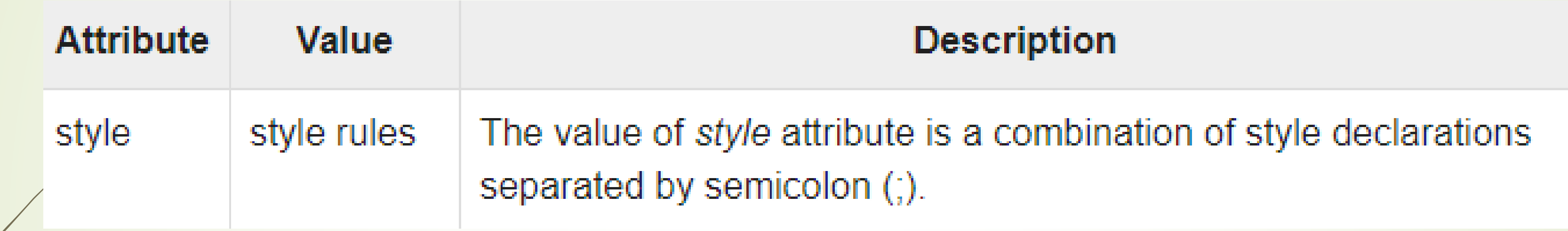
Imported CSS

Most commonly used methods are inline CSS and External CSS.

Inline CSS

You can use *style* attribute of any HTML element to define style rules. These rules will be applied to that element only. Here is the generic syntax –

<element style = "...style rules....">



Attribute	Value	Description
style	style rules	The value of *style* attribute is a combination of style declarations separated by semicolon (;).

Eg:

```
<html>
    <body>
        <h1 style="color : blue; text-align : center;">This is a heading</h1>
        <p style="color : red;">This is a paragraph.</p>
    </body>
</html>
```

Embedded CSS

You can put your CSS rules into an HTML document using the <style> element. This tag is placed inside the <head>...</head> tags. Rules defined using this syntax will be applied to all the elements available in the document.

Eg:

```
<html>
    <head>
        <style>
            body { background-color: linen;}
            h1 {color: maroon; margin-left: 40px;}
        </style>
    </head>
    <body>
        <h1>This is a heading</h1>
        <p>This is a paragraph.</p>
    </body>
</html>
```

External CSS

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

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

An external style sheet can be written in any text editor, and must be saved with a .css extension.

The external .css file should not contain any HTML tags.

Eg:

```
    <head>
        <link type = "text/css" href = "..." media = "..." />
    </head>
```

Example:

Consider a simple style sheet file with a name *mystyle.css* having the following rules −

```
h1, h2, h3 {
color: #36C;
font-weight: normal;
letter-spacing: .4em;
margin-bottom: 1em;
text-transform: lowercase;
}
```

Now you can include this file *mystyle.css* in any HTML document as follows −

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

Imported CSS

@import is used to import an external stylesheet in a manner similar to the <link> element. Here is the generic syntax of @import rule.

Eg:

```
<head>
    <style>
        @import "URL";
    </style>
</head>
```

Here URL is the URL of the style sheet file having style rules. You can use another syntax as well –

Eg:

```
<head>
    <style>
        @import url("URL");
    </style>
</head>
```

Example:

Following is the example showing you how to import a style sheet file into HTML document −

Eg:

```
<head>

    <style>

        @import "mystyle.css";
    </style>

</head>
```

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 is placed inside the <style> element, and starts with /* and ends with */:

Eg:

```
<html>
    <head>
        <style>
            p {color: red;} /* This is a single-line comment */
        </style>
    </head>
    <body>
        <p>Hello World!</p>
    </body>
</html>
```

CSS - Measurement Units

- CSS supports a number of measurements including absolute units such as inches, centimeters, points, and so on, as well as relative measures such as percentages and em units.

- You need these values while specifying various measurements in your Style rules e.g. **border : "1px solid red"**.

CSS - Measurement Units

Unit	Description	Example
%	Defines a measurement as a percentage relative to another value, typically an enclosing element.	p {font-size: 16pt; line-height: 125%;}
cm	Defines a measurement in centimeters.	div {margin-bottom: 2cm;}
em	A relative measurement for the height of a font in em spaces. Because an em unit is equivalent to the size of a given font, if you assign a font to 12pt, each "em" unit would be 12pt; thus, 2em would be 24pt.	p {letter-spacing: 7em;}
ex	This value defines a measurement relative to a font's x-height. The x-height is determined by the height of the font's lowercase letter x.	p {font-size: 24pt; line-height: 3ex;}
in	Defines a measurement in inches.	p {word-spacing: .15in;}
mm	Defines a measurement in millimeters.	p {word-spacing: 15mm;}
pc	Defines a measurement in picas. A pica is equivalent to 12 points; thus, there are 6 picas per inch.	p {font-size: 20pc;}
pt	Defines a measurement in points. A point is defined as 1/72nd of an inch.	body {font-size: 18pt;}
px	Defines a measurement in screen pixels.	p {padding: 25px;}

CSS - Colors

- CSS uses color values to specify a color. Typically, these are used to set a color either for the foreground of an element (i.e., its text) or else for the background of the element.

- They can also be used to affect the color of borders and other decorative effects.

CSS - Colors

Format	Syntax	Example
Hex Code	#RRGGBB	p{color:#FF0000;}
Short Hex Code	#RGB	p{color:#6A7;}
RGB %	rgb(rrr%,ggg%,bbb%)	p{color:rgb(50%,50%,50%);}
RGB Absolute	rgb(rrr,ggg,bbb)	p{color:rgb(0,0,255);}
keyword	aqua, black, etc.	p{color:teal;}

CSS Colors - Hex Codes

- A hexadecimal is a 6 digit representation of a color. The first two digits(RR) represent a red value, the next two are a green value(GG), and the last are the blue value(BB).

- A hexadecimal value can be taken from any graphics software like Adobe Photoshop, Jasc Paintshop Pro, or even using Advanced Paint Brush.

- Each hexadecimal code will be preceded by a pound or hash sign '#'. Following are the examples to use Hexadecimal notation.

CSS Colors - Hex Codes

Color	Color HEX
Black	#000000
Red	#FF0000
Green	#00FF00
Blue	#0000FF
Yellow	#FFFF00
cyan	#00FFFF
fuchsia	#FF00FF
Silver	#C0C0C0
White	#FFFFFF

CSS Colors - Short Hex Codes

- This is a shorter form of the six-digit notation. In this format, each digit is replicated to arrive at an equivalent six-digit value. For example: #6A7 becomes #66AA77.

- A hexadecimal value can be taken from any graphics software like Adobe Photoshop, Jasc Paintshop Pro, or even using Advanced Paint Brush.

- Each hexadecimal code will be preceded by a pound or hash sign '#'. Following are the examples to use Hexadecimal notation.

CSS Colors - Short Hex Codes

Color	Color HEX
Black	#000
Red	#F00
Green	#0F0
Blue	#00F
Yellow	#FF0
cyan	#0FF
fuchsia	#F0F
White	#FFF

CSS Colors - RGB Values

- This color value is specified using the **rgb( )** property. This property takes three values, one each for red, green, and blue. The value can be an integer between 0 and 255 or a percentage.

- **NOTE** – All the browsers does not support rgb() property of color so it is recommended not to use it.

- Following is the example to show few colors using RGB values.

CSS Colors - RGB Values

Color	Color RGB
Black	rgb(0,0,0)
Red	rgb(255,0,0)
Green	rgb(0,255,0)
Blue	rgb(0,0,255)
Yellow	rgb(255,255,0)
cyan	rgb(0,255,255)
fuchsia	rgb(255,0,255)
Silver	rgb(192,192,192)
White	rgb(255,255,255)

CSS - Backgrounds

You can set the following background properties of an element –

- The **background-color** property is used to set the background color of an element.
- The **background-image** property is used to set the background image of an element.
- The **background-position** property is used to control the position of an image in the background.
- The **background** property is used as a shorthand to specify a number of other background properties.

CSS background-color

Eg:

```
<html>
    <head>
        <style>
            body {background-color: blue;}
        </style>
    </head>
    <body>
        <h1>Hello World!</h1>
        <p>This page has a light blue background color!</p>
    </body>
</html>
```

Opacity

```
<html>
<head>
<style>
    div {background-color: green;}
    div.first {opacity: 0.3}
    div.second {opacity: 0.8;}
</style>
</head>
<body>
    <div class="first"><h1>opacity 0.3</h1></div>
    <div class="second"><h1>opacity 0.8</h1></div>
</body>
</html>
```

CSS background-Image

```
<html>
<head>
    <style>
        body {
            background-image: url(“image.jpg");
            background-color: #cccccc;
         }
    </style>
 </head>
<body>
    <h1>Hello World!</h1>
</body>
</html>
```

CSS background-position

```
<html>
<head>
<style>
body {
background-image: url("1.png");
background-position: right top;
margin-right: 200px;
}
</style>
</head>
<body>
<h1>Hello World!</h1>
<p>In this example we have also added a margin on the right side>
</body>
</html>
```

CSS - Fonts

You can set following font properties of an element −

- The **font-family** property is used to change the face of a font.
- The **font-style** property is used to make a font italic or oblique.
- The **font-variant** property is used to create a small-caps effect.
- The **font-weight** property is used to increase or decrease how bold or light a font appears.
- The **font-size** property is used to increase or decrease the size of a font.
- The **font** property is used as shorthand to specify a number of other font properties

CSS – font-family

```
<html>
<head>
</head>
 <body>
<p style = "font-family: georgia, garamond, serif;">
This text is rendered in either georgia, garamond, or the default serif font depending on which font you have at your system.
</p>
</body>
</html>
```

CSS – font-style

```
<html>
<head>
</head>
 <body>
        <p style = "font-style: italic;">
                Font style will be italic
        </p>
        <p style = "font-style: oblique;">
                Font style will be oblique
        </p>
</body>
</html>
```

CSS – font-variant

```
<html>
<head>
</head>
 <body>
        <p style = "font-variant: small-caps;">
                Font variant will be small-caps
        </p>
        <p style = "font-variant: normal;">
                Font variant will be normal
        </p>
</body>
</html>
```

CSS – font-weight

The font-weight property provides the functionality to specify how bold a font is. Possible values could be *normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900*.

```
<html>
<head> </head>
<body>
    <p style = "font-weight:bold;">
        This font is bold.
    </p>
    <p style = "font-weight:bolder;">
        This font is bolder.
    </p>
    <p style = "font-weight:500;">
        This font is 500 weight.
     </p>
</body>
</html>
```

CSS – font-size

```
<html>
<head> </head>
<body>
    <p style = "font-size:25px;">
         Font size will be 25px
    </p>
    <p style = "font-size: 10%;">
         Font size Will be 10%
    </p>
</body>
</html>
```

CSS – font

```
<html>

<head> </head>

<body>

    <p style="font: italic small-caps bold 12px/30px Georgia,
    serif;">shorthand</p>

</body>

</html>
```

CSS – Text

You can set following text properties of an element –

- The **color** property is used to set the color of a text.
- The **direction** property is used to set the text direction.
- The **letter-spacing** property is used to add or subtract space between the letters that make up a word.
- The **word-spacing** property is used to add or subtract space between the words of a sentence.
- The **text-indent** property is used to indent the text of a paragraph.

CSS – Text

- The **text-align** property is used to align the text of a document.
- The **text-decoration** property is used to underline, overline, and strikethrough text.
- The **text-transform** property is used to capitalize text or convert text to uppercase or lowercase letters.
- The **white-space** property is used to control the flow and formatting of text.
- The **text-shadow** property is used to set the text shadow around a text.

CSS – Text color

```
<html>
<head>
</head>
<body>
   <p style="color:blue">Text color</p>
   <p style="background-color: brown;">Text Background color</p>
</body>
</html>
```

CSS – Text Align and Direction

```
<html>
<head>
</head>
<body>
    <p style="text-align: center;">Text align center</p>
    <p style="direction: rtl;">Text align center</p>
</body>
</html>
```

CSS – Letter spacing and Word spacing

```
<html>
<head>
</head>
<body>
   <p style="letter-spacing: 20px;">Text letter spacing</p>
    <p style="word-spacing: 20px;">Text word spacing</p>
</body>
</html>
```

CSS – Text Decoration

```
<html>
<head>
</head>
<body>
    <p style="text-decoration: underline red;">Text decoration underline</p>
    <p style="text-decoration: overline red;">Text decoration overline</p>
    <p style="text-decoration: line-through red;">Text decoration line through</p>
</body>
</html>
```

CSS – Text Indent

```
<html>
<head>
</head>
<body>
    <p style="text-indent: 20px;">Text indent property
    <br>will be displayed</p>
</body>
</html>
```

CSS – Text Transform

```
<html>

<head>

</head>

<body>

    <p style="text-transform: capitalize;">text transform</p>

    <p style="text-transform: uppercase;">text transform</p>

    <p style="text-transform: lowercase;">TEXT TRANSFORM</p>
</body>

</html>
```

CSS – White Space

```
.a{
  white-space: nowrap;
}

.b{
  white-space: normal;
}

.c{
  white-space: pre;
}
```

CSS – Text Shadow

```
<html lang="en">
<head>
</head>
<body>
  <p style="text-shadow: 5px 5px brown;">Text shadow example</p>
</body>
</html>
```

CSS – Images

Images play an important role in any webpage. Though it is not recommended to include a lot of images, but it is still important to use good images wherever required.

CSS plays a good role to control image display. You can set the following image properties using CSS.

- The **border** property is used to set the width of an image border.
- The **height** property is used to set the height of an image.
- The **width** property is used to set the width of an image.
- The **-moz-opacity** property is used to set the opacity of an image.

CSS – Image Border Property

The *border* property of an image is used to set the width of an image border. This property can have a value in length or in %.

A width of zero pixels means no border.

Here is the example −

```
<html>
<head> </head>
<body>
       <img style = "border:0px;" src = “image1.png" />
       <br />
       <img style = "border:3px dashed red;" src = “image2.jpg" />
 </body>
</html>
```

CSS – Image Width Property

The *width* property of an image is used to set the width of an image. This property can have a value in length or in %. While giving value in %, it applies it in respect of the box in which an image is available.

Here is an example −

```
<html>
<head> </head>
<body>
    <img style = "border:1px solid red; width:150px;" src =“image1.png" />
    <br />
    <img style = "border:1px solid red; width:100%;" src = “image2.jpg" />
</body>
</html>
```

CSS – Image Height Property

The *width* property of an image is used to set the width of an image. This property can have a value in length or in %. While giving value in %, it applies it in respect of the box in which an image is available.

Here is an example −

```
<html>

<head> </head>

<body>

    <img style = "border:1px solid red; height:150px;" src =“image1.png" />

    <br />

    <img style = "border:1px solid red; height:100%;" src = “image2.jpg" />

</body>

</html>
```

CSS – Image Opacity Property

```
<html>
<head> </head>
<body>
    <img style = "opacity: .4;" src =“image1.PNG" />
    <img style = "opacity: .8;" src =“image2.PNG" />
    <img style = "opacity: .1;" src =“image3.PNG" />
</body>
</html>
```

CSS – Links

You can set following properties of a hyper link –

We will revisit the same properties when we will discuss Pseudo-Classes of CSS.

- The **:link** signifies unvisited hyperlinks.
- The **:visited** signifies visited hyperlinks.
- The **:hover** signifies an element that currently has the user's mouse pointer hovering over it.
- The **:active** signifies an element on which the user is currently clicking.

CSS – Links

Usually, all these properties are kept in the header part of the HTML document.

Remember a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective. Also, a:active MUST come after a:hover in the CSS definition as follows –

```
<html>
<head>
<style type = "text/css">
        a:link {color: #000000}
        a:visited {color: #006600}
        a:hover {color: #FFCC00}
        a:active {color: #FF00CC}
</style>
</head>
<body>
        <a href=“link.html”>Links Example</a>
</body>
</html>
```

CSS – Tables

You can set following properties of a table –

- The **border-collapse** specifies whether the browser should control the appearance of the adjacent borders that touch each other or whether each cell should maintain its style.
- The **border-spacing** specifies the width that should appear between table cells.
- The **caption-side** captions are presented in the <caption> element. By default, these are rendered above the table in the document. You use the *caption-side* property to control the placement of the table caption.
- The **empty-cells** specifies whether the border should be shown if a cell is empty.
- The **table-layout** allows browsers to speed up layout of a table by using the first width properties it comes across for the rest of a column rather than having to load the whole table before rendering it.(Values: auto and fixed)

CSS – Table Border Collapse Property

```
<html>
  <head>
   <style type="text/css">
      table.one {border-collapse:collapse;}
      table.two {border-collapse:separate;}
      td.a {
        border-style:dotted;
        border-width:3px;
        border-color:#000000;
        padding: 10px;
      }
      td.b {
        border-style:solid;
        border-width:3px;
        border-color:#333333;
        padding:10px;
      }
    </style>
</head>
  <body>
<table class="one">
      <caption>Collapse Border Example</caption>
      <tr><td class="a"> Cell A Collapse Example</td></tr>
      <tr><td class="b"> Cell B Collapse Example</td></tr>
    </table>
    <br />
    <table class="two">
      <caption>Separate Border Example</caption>
      <tr><td class="a"> Cell A Separate Example</td></tr>
      <tr><td class="b"> Cell B Separate Example</td></tr>
    </table>


  </body>
</html>
```

CSS – Table Empty-cells Property

```
<html>
<head>
<style>
    table.ex1 {empty-cells: show;}
</style>
</head>
<body>
<table class="ex1" border="1">
  <tr> <td>Peter</td> <td>Griffin</td> </tr>
  <tr> <td>Lois</td> <td></td> </tr>
</table>
```

CSS – Borders

The *border* properties allow you to specify how the border of the box representing an element should look. There are three properties of a border you can change –

- The **border-color** specifies the color of a border.
- The **border-style** specifies whether a border should be solid, dashed line, double line, or one of the other possible values.
- The **border-width** specifies the width of a border.

CSS – Border Color Property

The border-color property allows you to change the color of the border surrounding an element. You can individually change the color of the bottom, left, top and right sides of an element's border using the properties –

- **border-bottom-color** changes the color of bottom border.
- **border-top-color** changes the color of top border.
- **border-left-color** changes the color of left border.
- **border-right-color** changes the color of right border.

CSS – Border Color Property

```
<html>
  <head>
    <style type = "text/css">
      p.example1 {
        border:1px solid;
        border-bottom-color:#009900; /* Green */
        border-top-color:#FF0000;    /* Red */
        border-left-color:#330000;   /* Black */
        border-right-color:#0000CC;  /* Blue */
      }
      p.example2 {
        border:1px solid;
        border-color:#009900;        /* Green */
      }
    </style>
  </head>

  <body>
    <p class = "example1">
      This example is showing all borders in different colors.
    </p>

    <p class = "example2">
      This example is showing all borders in green color only.
    </p>
  </body>
</html>
```

CSS – Border Style Property

- **none** – No border. (Equivalent of border-width:0;)
- **solid** – Border is a single solid line.
- **dotted** – Border is a series of dots.
- **dashed** – Border is a series of short lines.
- **double** – Border is two solid lines.
- **groove** – Border looks as though it is carved into the page.
- **ridge** – Border looks the opposite of groove.
- **inset** – Border makes the box look like it is embedded in the page.
- **outset** – Border makes the box look like it is coming out of the canvas.
- **hidden** – Same as none, except in terms of border-conflict resolution for table elements.

CSS – Margins

The *margin* property defines the space around an HTML element. It is possible to use negative values to overlap content.

The values of the margin property are not inherited by the child elements. Remember that the adjacent vertical margins (top and bottom margins) will collapse into each other so that the distance between the blocks is not the sum of the margins, but only the greater of the two margins or the same size as one margin if both are equal.

We have the following properties to set an element margin.

- The **margin** specifies a shorthand property for setting the margin properties in one declaration.
- The **margin-bottom** specifies the bottom margin of an element.
- The **margin-top** specifies the top margin of an element.
- The **margin-left** specifies the left margin of an element.
- The **margin-right** specifies the right margin of an element.

CSS – Margin Property

```
<html>
  <head>
  </head>
  <body>
       <p style = "margin: 15px; border:1px solid black;">all four margins will be 15px</p>
      <p style = "margin:10px 2%; border:1px solid black;">top and bottom margin will be 10px, left and right margin will be 2% of the total width of the document.</p>
      <p style = "margin: 10px 2%  10px; border:1px  solid black;">top margin will be 10px, left and right margin will be 2% of the total width of the document, bottom margin will be 10px</p>
      <p style = "margin: 10px 2%  10px auto; border:1px solid black;">
        top margin will be 10px, right margin will be 2% of the total width of the document, bottom margin will be 10px, left margin will be set by the browser
      </p>
  </body>
</html>
```

CSS – Lists

Lists are very helpful in conveying a set of either numbered or bullet points. This chapter teaches you how to control list type, position, style, etc., using CSS.

We have the following five CSS properties, which can be used to control lists –

- The **list-style-type** allows you to control the shape or appearance of the marker.
- The **list-style-position** specifies whether a long point that wraps to a second line should align with the first line or start underneath the start of the marker.
- The **list-style-image** specifies an image for the marker rather than a bullet point or number.
- The **list-style** serves as shorthand for the preceding properties.

CSS – list-style-type

```
<head>
<style>
    ul.a {list-style-type: circle;}
    ul.b {list-style-type: disc;}
    ul.c {list-style-type: square;}
    ol.d {list-style-type: armenian;}
    ol.e {list-style-type: cjk-ideographic;}
    ol.f {list-style-type: decimal;}
    ol.g {list-style-type: decimal-leading-zero;}
    ol.h {list-style-type: georgian;}
    ol.i {list-style-type: hebrew;}
    ol.j {list-style-type: hiragana;}
    ol.k {list-style-type: hiragana-iroha;}
    ol.l {list-style-type: katakana;}
    ol.m {list-style-type: katakana-iroha;}
    ol.n {list-style-type: lower-alpha;}
    ol.o {list-style-type: lower-greek;}
    ol.p {list-style-type: lower-latin;}
    ol.q {list-style-type: lower-roman;}
    ol.r {list-style-type: upper-alpha;}
    ol.s {list-style-type: upper-greek;}
    ol.t {list-style-type: upper-latin;}
    ol.u {list-style-type: upper-roman;}
    ol.v {list-style-type: none;}
    ol.w {list-style-type: inherit;}
</style>
</head>
```

CSS – list-style-position

```
<head>
  <style>
      ul.a {list-style-position: outside;}
      ul.b {list-style-position: inside;}
   </style>
</head>
<body>
  <ul class="a">
    <li>Coffee <br> from coffee powder </li><li>Tea <br> from tea powder
    </li><li>Coca Cola <br> from chemical powder </li>
  </ul>
  <ul class="b">
    <li>Coffee <br> from coffee powder </li><li>Tea <br> from tea powder
    </li><li>Coca Cola <br> from chemical powder </li>
  </ul>
</body>
```

CSS – list-style-image

```
<head>
        <style>
                ul {
                        list-style-image: url('sqpurple.gif’);
                }
        </style>
</head>
<body>
        <ul>
                <li>Coffee</li>
                <li>Tea</li>
                <li>Coca Cola</li>
        </ul>
</body>
```

CSS – paddings

- The *padding* property allows you to specify how much space should appear between the content of an element and its border −
- The value of this attribute should be either a length, a percentage, or the word *inherit*. If the value is *inherit*, it will have the same padding as its parent element. If a percentage is used, the percentage is of the containing box.
- The following CSS properties can be used to control lists. You can also set different values for the padding on each side of the box using the following properties −
- **The padding-bottom specifies the bottom padding of an element.**
- **The padding-top specifies the top padding of an element.**
- **The padding-left specifies the left padding of an element.**
- **The padding-right specifies the right padding of an element.**
- The padding serves as shorthand for the preceding properties.

CSS – paddings

```
<body>
    <p style = "padding: 15px; border:1px solid black;">
        all four padding will be 15px
   </p>
   <p style = "padding:10px 2%; border:1px solid black;">
      top and bottom 10px, left and right 2% of the total width of the
      document.
   </p>
   <p style = "padding: 10px 2% 10px; border:1px solid black;">
     top 10px, left and right 2% of the total width of the document,
     bottom 10px.
   </p>
   <p style = "padding: 10px 2% 10px 10px; border:1px solid black;">
     top 10px, right 2% of the total width of the document, bottom and
     left 10px
   </p>
 </body>
```

CSS – Cursor

- The *cursor* property of CSS allows you to specify the type of cursor that should be displayed to the user.

- One good usage of this property is in using images for submit buttons on forms. By default, when a cursor hovers over a link, the cursor changes from a pointer to a hand. However, it does not change form for a submit button on a form. Therefore, whenever someone hovers over an image that is a submit button, it provides a visual clue that the image is clickable.

CSS – Cursor

- **auto**

  Shape of the cursor depends on the context area it is over. For example an I over text, a hand over a link, and so on...

- **crosshair**

  A crosshair or plus sign

- **default**

  An arrow

- **pointer**

  A pointing hand (in IE 4 this value is hand)

- **move**

  The I bar

- **e-resize**

  The cursor indicates that an edge of a box is to be moved right (east)

- **ne-resize**

  The cursor indicates that an edge of a box is to be moved up and right (north/east)

- **nw-resize**

  The cursor indicates that an edge of a box is to be moved up and left (north/west)

- **n-resize**

  The cursor indicates that an edge of a box is to be moved up (north)

CSS – Cursor

- **se-resize**

  The cursor indicates that an edge of a box is to be moved down and right (south/east)

- **sw-resize**

  The cursor indicates that an edge of a box is to be moved down and left (south/west)

- **s-resize**

  The cursor indicates that an edge of a box is to be moved down (south)

- **w-resize**

  The cursor indicates that an edge of a box is to be moved left (west)

- **text**

  The I bar

- **Wait**

  An hour glass

- **help**

  A question mark or balloon, ideal for use over help buttons

- **<url>**

  The source of a cursor image file

```
 <div style = "cursor:auto">Auto</div>
<div style = "cursor:crosshair">Crosshair</div>
<div style = "cursor:default">Default</div>
<div style = "cursor:pointer">Pointer</div>
<div style = "cursor:move">Move</div>
<div style = "cursor:e-resize">e-resize</div>
  <div style = "cursor:ne-resize">ne-resize</div>
 <div style = "cursor:nw-resize">nw-resize</div>
 <div style = "cursor:n-resize">n-resize</div>
<div style = "cursor:se-resize">se-resize</div>
<div style = "cursor:sw-resize">sw-resize</div>
<div style = "cursor:s-resize">s-resize</div>
<div style = "cursor:w-resize">w-resize</div>
<div style = "cursor:text">text</div>
<div style = "cursor:wait">wait</div>
<div style = "cursor:help">help</div>
```

CSS – Outline

Outlines are very similar to borders, but there are few major differences as well –

- An outline does not take up space.
- Outlines do not have to be rectangular.
- Outline is always the same on all sides; you cannot specify different values for different sides of an element.

You can set the following outline properties using CSS.

- The **outline-width** property is used to set the width of the outline.
- The **outline-style** property is used to set the line style for the outline.
- The **outline-color** property is used to set the color of the outline.
- The **outline** property is used to set all the above three properties in a single statement.

CSS – outline-style

The *outline-style* property specifies the style for the line (solid, dotted, or dashed) that goes around an element. It can take one of the following values –

- **none** – No border. (Equivalent of outline-width:0;)
- **solid** – Outline is a single solid line.
- **dotted** – Outline is a series of dots.
- **dashed** – Outline is a series of short lines.
- **double** – Outline is two solid lines.
- **groove** – Outline looks as though it is carved into the page.
- **ridge** – Outline looks the opposite of groove.
- **inset** – Outline makes the box look like it is embedded in the page.
- **outset** – Outline makes the box look like it is coming out of the canvas.
- **hidden** – Same as none.

CSS – Outline Properties

- The *outline* property is a shorthand property that allows you to specify values for any of the three properties discussed previously in any order but in a single statement.

```
<body>
    <p style = "outline:thin solid red;">
      This text is having thin solid red outline.
    </p>
    <br />
    <p style = "outline:thick dashed #009900;">
      This text is having thick dashed green outline.
    </p>
    <br />
    <p style = "outline:5px dotted rgb(13,33,232);">
      This text is having 5x dotted blue outline.
    </p>
 </body>
```

CSS – Dimensions

- You have seen the border that surrounds every box ie. element, the padding that can appear inside each box and the margin that can go around them.
- We have the following properties that allow you to control the dimensions of a box.
- The **height** property is used to set the height of a box.
- The **width** property is used to set the width of a box.
- The **line-height** property is used to set the height of a line of text.

CSS – Height and Width Properties

- The *height* and *width* properties allow you to set the height and width for boxes. They can take values of a length, a percentage, or the keyword auto.

```
<html>
  <head>
  </head>
     <body>
    <p style = "width:400px; height:100px; border:1px solid red; padding:5px; margin:10px;">
      This paragraph is 400pixels wide and 100 pixels high
    </p>
  </body>
</html>
```

CSS – line-height Properties

- The *line-height* property allows you to increase the space between lines of text. The value of the line-height property can be a number, a length, or a percentage.

```
<html>
  <head>
  </head>

  <body>
    <p style = "width:400px; height:100px; border:1px solid red; padding:5px; margin:10px; line-height:30px;">
      This paragraph is 400pixels wide and 100 pixels high and here line height is 30pixels.
      This paragraph is 400 pixels wide and 100 pixels high and here line height is 30pixels.
    </p>
  </body>
</html>
```

CSS – Scrollbars

- There may be a case when an element's content might be larger than the amount of space allocated to it. For example, given width and height properties do not allow enough room to accommodate the content of the element.
- CSS provides a property called *overflow* which tells the browser what to do if the box's contents is larger than the box itself. This property can take one of the following values −
- **visible**

  Allows the content to overflow the borders of its containing element.
- **hidden**

  The content of the nested element is simply cut off at the border of the containing element and no scrollbars is visible.
- **scroll**

  The size of the containing element does not change, but the scrollbars are added to allow the user to scroll to see the content.
- **auto**

  The purpose is the same as scroll, but the scrollbar will be shown only if the content does overflow.

```
<style type = "text/css">
      .scroll {
        display:block;
        border: 1px solid red;
        padding:5px;
        margin-top:5px;
        width:300px;
        height:50px;
        overflow:scroll;
      }
      .auto {
        display:block;
        border: 1px solid red;
        padding:5px;
        margin-top:5px;
        width:300px;
        height:50px;
        overflow:auto;
      }
</style>
```

```
<body>
    <p>Example of scroll value:</p>
    <div class = "scroll">
      I am going to keep lot of content here just to show
you how  scrollbars works if there is an      overflow in an
element box. This provides your horizontal as well as
vertical scrollbars.
    </div>
    <br />
    <p>Example of auto value:</p>
    <div class = "auto">
      I am going to keep lot of content here just to show
you how scrollbars works if there is an      overflow in an
element box. This provides your horizontal as well as
vertical scrollbars.
    </div>
  </body>
```

CSS – Navigation Bar

- A navigation bar needs standard HTML as a base.
- A navigation bar is basically a list of links, so using the <ul> and <li> elements makes perfect sense:

```
<html>
<body>
    <ul>
        <li><a href="#home">Home</a></li>
        <li><a href="#news">News</a></li>
        <li><a href="#contact">Contact</a></li>
        <li><a href="#about">About</a></li>
    </ul>
    <p>Note: We use href="#" for test links. In a real web site this would be URLs.</p>
</body>
</html>
```

CSS – Navigation Bar

```
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333;
}
li {float: left;}
li a {
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
 text-decoration: none;
}
li a:hover {
  background-color: #111;
}
</style>
</head>
<body>
<ul>
  <li><a class="active" href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>
</body>
```

Media Queries

- Media queries are a popular technique for delivering a tailored style sheet to different devices.

- Media queries can be used to check many things, such as:
- width and height of the viewport
- width and height of the device
- orientation (is the tablet/phone in landscape or portrait mode?)
- resolution

- To demonstrate a simple example, we can change the background color for different devices

Media Queries

```
<head>
<style>
body {
  background-color: tan;
  color: black;
}
/* On screens that are 992px wide or less, the background color is blue */
@media screen and (max-width: 992px) {
  body {
    background-color: blue;
    color: white;
  } }
/* On screens that are 600px wide or less, the background color is olive */
@media screen and (max-width: 600px) {
body {
    background-color: olive;
    color: white;
  } }
</style>
</head>
<body>
	<h1>Resize the browser window to see the effect!</h1>
	<p>By default, the background color of the document is "tan". If the screen size is 992px or less, the color will change to "blue". If it is 600px or less, it will change to "olive".</p>
</body>
</html>
```

Website Layout Example

```
<!DOCTYPE html>
<html>
<head>
    <style>
        * {
            box-sizing: border-box;
        }
        body {
            font-family: Arial;
            padding: 10px;
            background: #f1f1f1;
        }
/* Header/Blog Title */
        .header {
            padding: 30px;
            text-align: center;
            background: white;
        }
        .header h1 {
            font-size: 50px;
        }
/* Style the top navigation bar */
        .topnav {
            overflow: hidden;
            background-color: #333;
        }
/* Style the topnav links */
        .topnav a {
            float: left;
            display: block;
            color: #f2f2f2;
            text-align: center;
            padding: 14px 16px;
            text-decoration: none;
        }
```

```
/*Change color on hover */
        .topnav a:hover {
            background-color: #ddd;
            color: black;
        }
/* Create two unequal columns that floats next to each other */
/* Left column */
        .leftcolumn {
            float: left;
            width: 75%;
        }
/* Right column */
        .rightcolumn {
            float: left;
            width: 25%;
            background-color: #f1f1f1;
            padding-left: 20px;
        }
```

```
/* Fake image */
        .fakeimg {
            background-color: #aaa;
            width: 100%;
            padding: 20px;
        }
/* Add a card effect for articles */
        .card {
            background-color: white;
            padding: 20px;
            margin-top: 20px;
        }
/* Clear floats after the columns */
        .row:after {
            content: "";
            display: table;
            clear: both;
        }
```

```
/* Footer */
        .footer {
            padding: 20px;
            text-align: center;
            background: #ddd;
            margin-top: 20px;
        }
/* Responsive layout - when the screen is less than 800px wide, make the two columns stack
on top of each other instead of next to each other */
        @media screen and (max-width: 800px) {
            .leftcolumn, .rightcolumn {
                width: 100%;
                padding: 0;
            } \
        }
/* Responsive layout - when the screen is less than 400px wide, make the navigation links
stack on top of each other instead of next to each other */
```

```
        @media screen and (max-width: 400px) {
            .topnav a {
                float: none;
                width: 100%;
            }
        }
    </style>
</head>
<body>
    <div class="header">
        <h1>My Website</h1>
        <p>Resize the browser window to see the effect.</p>
    </div>
    <div class="topnav">
        <a href="#">Link</a>
        <a href="#">Link</a>
        <a href="#">Link</a>
        <a href="#" style="float:right">Link</a>
    </div>
```

```
<div class="row">
    <div class="leftcolumn">
        <div class="card">
            <h2>TITLE HEADING</h2>
            <h5>Title description, Dec 7, 2017</h5>
            <div class="fakeimg" style="height:200px;">Image</div>
            <p>Some text..</p>
            <p>Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur
                adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore
                magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation
                ullamco.
            </p>
        </div>
        <div class="card">
            <h2>TITLE HEADING</h2>
            <h5>Title description, Sep 2, 2017</h5>
            <div class="fakeimg" style="height:200px;">Image</div>
            <p>Some text..</p>
        <p>Sunt in culpa qui officia deserunt mollit anim id est laborum
           consectetur adipiscing elit, sed do eiusmod tempor incididunt ut
           labore etdolore magna aliqua. Ut enim ad minim veniam, quis
           nostrud exercitation ullamco.
        </p>
    </div>
</div>
<div class="rightcolumn">
    <div class="card">
    <h2>About Me</h2>
    <div class="fakeimg" style="height:100px;">Image</div>
    <p>Some text about me in culpa qui officia deserunt mollit anim..</p>
</div>
<div class="card">
    <h3>Popular Post</h3>
    <div class="fakeimg">
        <p>Image</p>
    </div>
```

```
                <div class="fakeimg">
                    <p>Image</p>
                </div>
                <div class="fakeimg">
                    <p>Image</p>
                </div>
            </div>
            <div class="card">
                <h3>Follow Me</h3>
                <p>Some text..</p>
            </div>
        </div>
    </div>
    <div class="footer">
        <h2>Footer</h2>
    </div>
</body>
</html>
```