II SWAMI SHREEJI II

ATMIYA

YOGIDIVINE SOCIETY PRERIT,
SARVODAY KELAVNI SAMAJ SANCHALIT
INSTITUTE OF TECHNOLOGY & SCIENCE, RAJKOT

SUBJECT: WEB TECHNOLOGY
Subject Code - 2160708
6th Semester Computer Engineering

Faculty: Chhaya Patel, Jaydeep Tadhani

Unit 4 : Style sheets

References:

Book: Developing Web Applications, Ralph Moseley and M. T. Savaliya, Wiley-India

Web link: https://www.tutorialspoint.com/css/index.htm

Web Link: https://www.w3schools.com/css/css3_intro.asp

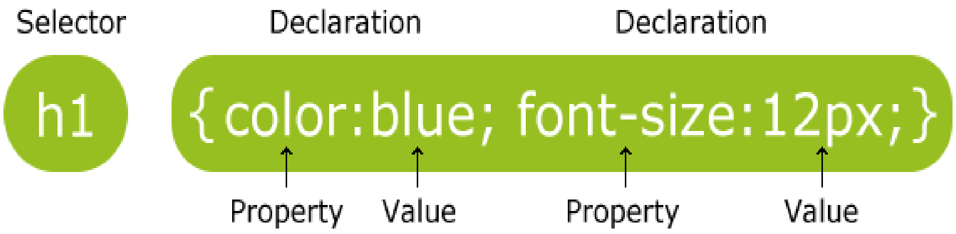
➢ Introduction to CSS OR Need of CSS.

- CSS is the acronym for "Cascading Style Sheet".
- CSS is a markup language used in the web development for presentation purpose.
- CSS handles the look and feel part of a web page.
- The primary goal of CSS was to separate out the web content from the web presentation.
- Various elements such as text, fonts, and color are used in CSS for presentation purpose.
- CSS is combined with the markup languages HTML or XHTML.
- **Advantages of CSS or Need for CSS**
  1. **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.

2. **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.
3. **Easy maintenance** – To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
4. **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.
5. **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.
6. **Global web standards** – Now HTML attributes are being deprecated and it is being recommended to use CSS. So it's a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.
7. **Offline Browsing** – CSS can store web applications locally with the help of an offline cache. Using of this, we can view offline websites. The cache also ensures faster loading and better overall performance of the website.
8. **Platform Independence** – The Script offer consistent platform independence and can support latest browsers as well.

➢ Basic syntax and structure of CSS

- When a browser reads a style sheet, it will format the HTML document according to the information in the style sheet.
- A CSS rule-set consists of a selector and a declaration block.



- 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.
- In the following example all <p> elements will be center-aligned, with a red text color.

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

➢ **What are different ways to write CSS?**

- **There are three ways of inserting a style sheet:**
  1. **Inline style sheet : by using the style attribute in HTML elements**
     - An inline CSS is used to apply a unique style to a single HTML element.
     - To use inline styles, **add the style attribute to the relevant element**.
     - The style attribute can contain any CSS property.
     - **Example:**

```
<!DOCTYPE html>
<html>
<head>
       <title>Inline Cascading Style Sheet</title>
</head>
<body>
       <p>This is simple text</p>
       <p style="font-size: 30pt; font-family: script">This text
with inline style</p>
       <p style="font-size: 40pt; color: red">This text is
colored</p>
</body>
</html>
```

- **Advantages:** using inline style sheet we can apply the uniform style on tags for the whole document.
- **Disadvantages:** An inline style not much suitable for web page designing because of mixing content with presentation.

2. **Internal style sheet(Embedded style sheet) : by using a <style> tag in the <head> section**
    - 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, within a <style> element.
    - Also appears in the body section newly defined selector tags are used with the actual contents.
    - We should mention the **<style type="text/css">** in the head section, by this the browser will come to know that the program is making use of CSS.
    - **Advantages:** it is useful when we want to apply the unique style for the webpage.
    - **Disadvantages:** when we what to apply same style to all WebPages that time internal style sheet is not used.
    - **Example:**

```
<!DOCTYPE html>
<html>
<head>
      <title>Internal Cascading Style Sheet</title>
      <style type="text/css">
            body {background-color: yellow}
            h1
            {
            color:black;
            font-family: Arial;
            }
            p
            {
            color: red;
            font-family: Arial;
```

```
        }
    </style>
</head>
<body>
    <h1>This is heading</h1>
    <p>This is paragraph</p>
</body>
</html>
```

3. External style sheet: by using an external CSS file

- 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 web site, by changing one file.
- Desired style is stored in one **.css** file.
- Add this **.css** file link in the <head> section of the HTML page using <link> element of HTML.
- The <link> tag defines relationships between a current document and external style sheets.
- **Example:**

externalstyle.html

```
<!DOCTYPE html>
<html>
<head>
    <title>External Style Sheet</title>
    <link rel="stylesheet" type="text/css" href="external.css">
</head>
<body>
<h1>This is heading</h1>
<p>This is paragraph</p>
</body>
</html>
```

external.css

```
body {background-color: yellow;}
        h1
        {
        color:black;
        font-family: Arial;
        }
        p
        {
        color: red;
        font-family: Arial;
        }
```

- **Advantages:** With an external style sheet, you can change the look of an entire web site, by changing one file.

Explain different type of selectors.

- A CSS selector is the part of a CSS rule set that actually selects the content you want to style.
- **Types of selectors:**

1. HTML/Simple selector

- To define how a particular HTML tag should be styled across your Web page, use its HTML selector.
- For example, if you say that all header level 2 tags should be red: We can apply same style to more than one selector.

```
h2,h3 { color: red; }
```

- The styles will be applied to any content tagged with:

```
<h2>...</h2>
<h3>...</h3>
```

- All header level 2 and level 3 tags on the page will be red, unless you override its declarations with other declarations.

2. Class selector

- Using class selector we can assign different styles to the same element.
- These different styles appear on different occurrences of that element.

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

```
<!DOCTYPE html>
<html>
<head>
       <title>Internal Cascading Style Sheet</title>
       <style type="text/css">
             body {background-color: yellow}
             p.center
             {
                    text-align: center;
                    color: red;
             }
             p.left
             {
                    text-align: left;
                    color: blue;
             }
       </style>
</head>
<body>
       <p class="center">This is paragraph is center align</p>
       <p class="left">This paragraph is left align</p>

</body>
</html>
```

3. Generic selector

- Using generic selector, particular class can be applied to any tag.
- 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:

```
<!DOCTYPE html>
<html>
<head>
   <title>Internal Cascading Style Sheet</title>
   <style type="text/css">
          body {background-color: yellow}
          p.center
          {
                 text-align: center;
```

```
                color: red;
          }
          .left
          {
                text-align: left;
                color: blue;
          }
   </style>
</head>
<body>
   <p class="center">This is paragraph is center align</p>
   <h1 class="left">This is heading 1</h1>
   <h2 class="left">This is heading 2</h2>
   <p class="left">This is paragraph</p>

</body>
</html>
```

4. id selector

- ID selector is similar to class selector but only one difference is that class selector can be applied to more than one element where as using ID selector style can be applied to the one specific element.
- ID selector denoted by # sign.
- Example:

```
<!DOCTYPE html>
<html>
<head>
   <title>ID Selector</title>
   <style type="text/css">
          #top
          {
            color: blue;
            font-size: 30pt;
          }
   </style>
</head>
<body>
   <div id="top">ID selector style can be applied to the one
specific element.</div>
</body>
</html>
```

5. Universal selector

- Universal selector can be applied to all the elements in the documents.
- This selector is denoted by * sign.
- Example:

```
<!DOCTYPE html>
<html>
<head>
   <title>Universal Selector</title>
   <style type="text/css">
          *{
                color: green;
          }
   </style>
</head>
<body>
   <h1>This is univesal selector for h1</h1>
    <h2>This is univesal selector for h2</h2>
    <h3>This is univesal selector for h3</h3>
</body>
</html>
```

➢ Background images, colors and properties

- The CSS background properties are used to define background effects for elements.
- CSS background properties are
  - background-color
  - background-image
  - background-repeat
  - background-position
  - background-attachment

- **Background Color**
  - The background-color property specifies the background color of an element.
  - 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)"
  - The background color of a page is set like this:

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

```
   <title></title>
   <style type="text/css">
          body
          {
          background-color: lightblue;
          }
          p{background-color: pink;}
          h1
          {
          background-color: white;
          }

   </style>
</head>
<body>
<p>This is Background Color Testing</p>
<h1>This is Background Color Testing</h1>
</body>
</html>
```

- **Background Image**
    - The background-image property specifies an image to use as the background 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:

```
<!DOCTYPE html>
<html>
<head>
   <title></title>
   <style type="text/css">
          body
          {
          background-image: url(image.jpg);
          }
   </style>
</head>
<body>
</body>
</html>
```

- **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.
    - Image repeated only horizontally by using (background-repeat: repeat-x;),
    - To repeat an image vertically, set (background-repeat: repeat-y);
    - Example:

```
<!DOCTYPE html>
<html>
<head>
   <title></title>
   <style type="text/css">
          body
          {
          background-image: url("image.jpg");
          background-repeat: repeat-y;
          }
   </style>
</head>
<body>
</body>
</html>
```

- **Background Image – Attachment - 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.
  - Example

```
<!DOCTYPE html>
<html>
<head>
   <title></title>
   <style type="text/css">
          body
          {
          background-image: url("image.jpg");
          background-repeat: no-repeat;
        background-position: right top;
        background-attachment: fixed;
          }
   </style>
</head>
<body>
</body>
</html>
```

- **Background Image - Set position and no-repeat**
  - Background image only once is specified by the background-repeat: no-repeat; property.
  - The position of the image is specified by the background-position property: background-position: right top;
  - Example

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

```
<head>
   <title></title>
   <style type="text/css">
          body
          {
          background-image: url("image.jpg");
          background-repeat: no-repeat;
          background-position: right top;
          }
   </style>
</head>
<body>
</body>
</html>
```

- **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:
  - 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.
  - Example

```
<!DOCTYPE html>
<html>
<head>
   <title></title>
   <style type="text/css">
   Body
   {
    background: red url("image.jpg") no-repeat left top;
   }
   </style>
</head>
<body>
</body>
</html>
```

- **Describe CSS texts with all its attributes.**
  - CSS text is styled with some of the text formatting properties.
  - The heading uses the text-align, text-transform, and color properties.
  - The paragraph is indented, aligned, and the space between characters is specified.
  - **Text Color**
    - The color property is used to set the color of the text.
    - The color is specified by:
      - a color name - like "red"
      - a HEX value - like "#ff0000"
      - an RGB value - like "rgb(255,0,0)"
      - Example

        ```
        body
        {
            color: blue;
        }
        ```

  - **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):
    - Example

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

      h2
      {
          text-align: left;
      }

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

```
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.
    - Example

      ```
      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;
      }
      ```

- **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.
    - Example

      ```
      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.
  - Example

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

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

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

- **Line-height**
  - The line-height property is used to specify the space between lines
  - Example

```
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.
  - Example

```
p
```

```
{
    direction: rtl;
}
```

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

```
h1
{
    word-spacing: 10px;
}
```

- **Text Shadow**
  - The text-shadow property adds shadow to text.
  - text-shadow: 3px (right), 2px(bottom), red(color).
  - Example

```
h1
{
    text-shadow: 3px 2px red;
}
```

➢ Describe CSS Fonts with all its attributes.

- 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.

- **Font Family**
  - In CSS, there are two types of font family names:
  - **generic family** - a group of font families with a similar look (like "Serif" or "Monospace").
  - **font family** - a specific font family (like "Times New Roman" or "Arial").
  - 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.
    - 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.
    - **Example:**

    ```
    p {
        font-family: "Times New Roman", Times, serif;
      }
    ```

- **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)
    - **Example:**

    ```
    <html>
       <head>
       </head>
       <body>
          <p style="font-style:italic;">
          This text will be rendered in italic style
          </p>
       </body>
    </html>
    ```

- **Font Size**
    - The font-size property is used to control the size of fonts.
    - Possible values could be xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, size in pixels or in %.
    - **Example**

    ```
    <html>
       <head>
       </head>
       <body>
          <p style="font-size:20px;">This font size is 20 pixels</p>
          <p style="font-size:small;">This font size is small</p>
    ```

```
      <p style="font-size:large;">This font size is large</p>
   </body>
</html>
```

- **Font Weight**
  - The font-weight property specifies the weight of a font: Bold or Normal.
  - **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.
  - **Example:**

```
<html>
   <head>
   </head>
   <body>
      <p style="font-variant:small-caps;">
      This text will be rendered as small caps
      </p>
   </body>
</html>
```

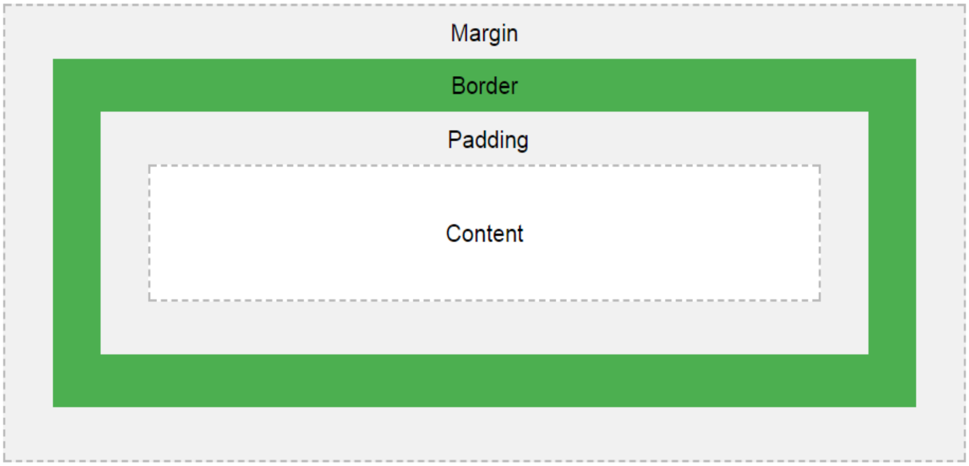
- **Shorthand Property**
  - You can use the font property to set all the font properties at once.
  - **Example**

```
<html>
   <head>
   </head>
   <body>
      <
      Applying all the properties on the text at once.
      </p>
   </body>
</html>
```

➢ Describe CSS Boxes or Box Model.

- The box comes in picture when we put the contents within some rectangle.

- 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:



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

```
<!DOCTYPE html>
<html>
<head>
    <title>Boxes</title>
    <style type="text/css">
            p{
                    background-color: lightgray;
                    width:300px;
                    border:25x solid green;
                    padding: 25px;
                    margin: 25px;
            }
    </style>
</head>
<body>
<p>Box demonstration</p>
```

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

➢ Describe CSS Margins.

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

```
<!DOCTYPE html>
<html>
<head>
   <title>Boxes</title>
   <style type="text/css">
         p{
               background-color: lightgray;
               width:300px;
               border:25x solid green;
               margin-left: 15px;
               margin-right: 30px;
         }
   </style>
</head>
<body>
<p>Box demonstration</p>
</body>
</html>
```

➢ Describe 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.

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

```
<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>
```

- **border-style**
    - The border-style property allows you to select one of the following styles of border.
        - **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.
  - You can individually change the style of the bottom, left, top, and right borders of an element using the following properties −
    - **border-bottom-style** changes the style of bottom border.
    - **border-top-style** changes the style of top border.
    - **border-left-style** changes the style of left border.
    - **border-right-style** changes the style of right border.
    - **Example**

```
<!DOCTYPE html>
<html>
<head>
   <title>Brorder Implementation</title>
   <style type="text/css">
          p.example
          {
                 border-style: groove;
                 border-left-style: groove;
                 border-top: dotted;
                 border-top-color: red;
          }
   </style>
</head>
<body>
<p class="example">I have borders on all sides.</p>
</body>
</html>
```

- **border – width**
  - The border-width property allows you to set the width of an element borders. The value of this property could be either a length in px, pt or cm or it should be set to thin, medium or thick.
  - You can individually change the width of the bottom, top, left, and right borders of an element using the following properties –

        - **border-bottom-width** changes the width of bottom border.
        - **border-top-width** changes the width of top border.
        - **border-left-width** changes the width of left border.
        - **border-right-width** changes the width of right border.
        - **Example**

```
<!DOCTYPE html>
<html>
<head>
   <title>Brorder Implementation</title>
   <style type="text/css">
          p.example
          {
                border-style: groove;
                border-left-style: groove;
                border-top: dotted;
                border-top-color: red;
                border-width: thick;
                border-top-width: thin;
          }


   </style>
</head>
<body>
<p class="example">I have borders on all sides.</p>
</body>
</html>
```

- **Border Properties Using Shorthand**
    - The border property allows you to specify color, style, and width of lines in one property.
    - The following example shows how to use all the three properties into a single property.
    - This is the most frequently used property to set border around any element.

```
<html>
   <head>
   </head>
   <body>
      <p style="border:10px solid red;">
      This example is showing shorthand property for border.
      </p>
   </body>
</html>
```

Describe CSS Padding.

- 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 inherited, 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** serves as shorthand for the preceding 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.
    - **Example**

```
<!DOCTYPE html>
<html>
<head>
   <title>Boxes</title>
   <style type="text/css">
          p{
                background-color: lightgray;
                width:300px;
                border:25x solid green;
                padding: inherit;
                padding-left: 30px;
                padding-bottom: 45px;
          }
   </style>
</head>
<body>
<p>Box demonstration</p>
</body>
</html>
```

Describe CSS Lists.

- Lists are very helpful in conveying a set of either numbered or bullet points.
- We have the following CSS properties, which can be used to control lists:

- The **list-style-type** allows you to control the shape or appearance of the marker.
    - Values for unordered list: none, disc (default), circle, square.
    - Values for ordered list: decimal, decimal-leading-zero, lower-alpha, upper-alpha, lower-roman, upper-roman, lower-greek etc…
- 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.
    - Values for lists: inside, outside.
- The **list-style-image** specifies an image for the marker rather than a bullet point or number.
    - Values for lists: none, url.
- The **list-style** serves as shorthand for the preceding properties.

```
<style type="text/css">
ul { list-style: square inside url("image.jpg");  }
ol{list-style-type: decimal-leading-zero;list-style-position:inside;}
</style>
```

➢ Describe CSS Positioning.

- CSS Position is used to give custom position to particular element.
- We can use position property with four different values: static (default), relative, and absolute, fixed.
    - **static:** static is default value and default position of elements.

      ```
      h1{position: static;}
      ```

    - **relative:** An element with **position: relative** is positioned at the specified coordinates relative to its normal position. We have to use four different values for relative position: left, right, top, bottom.

      ```
      h2{position:relative; top:200px; }
      ```

    - **absolute:** An element with **position: absolute** is positioned at the specified coordinates relative to your screen top-left corner.

      ```
      p{position: absolute;top: 200px;}
      ```

- **fixed:** Fixed positioning allows you to fix the position of an element to a particular spot on the page, regardless of scrolling. Specified coordinates will be relative to the browser window.

```
img{position: fixed; right:0px;top: 0px;}
```

➢ Describe Overview and features of CSS3.

- CSS3 is the latest version of CSS and contains a number of exciting, new features that make it easier for web developers to create great styling for websites.
- CSS3 is completely backwards-compatible with earlier versions of CSS.
- CSS3 has been split into "modules". It contains the "old CSS specification" (which has been split into smaller pieces). In addition, new modules are added.
- Following are new features of CSS3
  - Provide various text effects
  - Selectors including attribute selector.
  - Support more colors and custom fonts.
  - Improvements in borders and backgrounds.
  - Supports rounded corners for the box and shadow effects.
  - Support for multi-column text.
  - Animation and transitions
  - Gradients
  - 2-D/3-D graphics etc…
  - **For example:**
    1. CSS3 provides text effects with property text-shadow.

       We can add blur-radius and negative position values in text.

       ```
       p{text-shadow: 12px 12px 5px, -12px -12px 5px red;}
       ```

    2. CSS3 provides rounded borders**.**

       ```
       h1{
       margin: 20px;border: 2px solid blue; padding: 20px;text-align:
       center; border-radius: 10px 50px 10px 50px;}
       ```

3. CSS3 provide more than one image as background.

   ```
   body{background-image:url(wt.jpg),url(css.jpg);background-position:right top,left top;
   background-repeat: no-repeat,repeat;}
   ```

*****************************END********************************