UNIT-3
Cascading Style Sheet(CSS)

Q) What is CSS ?

- ➔ **CSS** is the acronym for **"Cascading Style Sheet"**. It's a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS helps the web developers to control the layout and other visual aspects of the web pages.
- ➔ CSS plays a crucial role in modern web development by providing the tools necessary to create visually appealing, accessible, and responsive websites.
- ➔ The current version of CSS is CSS3, and all major browsers support it. In 2009, the W3C started work on CSS4. But, CSS4 is not fully supported by the major browsers yet.
- ➔ **Advantages of Using CSS :**
  - ★ Responsive design
  - ★ Flexibility and Control
  - ★ Consistency and Reusability
  - ★ Search Engine Optimization (SEO)
  - ★ Ease of Maintenance
  - ★ Faster Page Loading
- ➔ **CSS Rules :** CSS works by associating rules with HTML elements.A **CSS rule** is a fundamental component in CSS that defines how HTML elements should be styled. Each rule specifies the styling instructions for targeted elements.
  - ◆ ***Components css rule:*** A CSS rule contains two main parts

    1. a **selector** which specifies the HTML element(s) to style.
       - ○ ***Example :*** h1, .class-name, #id-name, *

    2. a **declaration block** enclosed within curly braces {} and which contains one or more declarations separated by semicolons.
       - ○ Each declaration includes a **property** and a **value**, specifying the aspect of the element's presentation to control.
       - ○ ***Example :*** color: blue; font-size: 16px;

Example of CSS Rule:

```
h1 {
  color: blue;
  font-size: 24px;
}
```

Q)Write about selectors in css??

- ➔ **Selectors :** CSS use selectors used to style HTML Elements.CSS selectors are patterns used to select and style HTML elements. They are fundamental in applying styles to specific parts of a webpage.
- ➔ **Basic Selectors**

1. Type Selector : A **type selector** (also known as an element selector) targets all instances of a specific HTML element type. It is the most straightforward way to apply styles to elements of a particular type across a webpage.

Syntax:	***Example:***
element {	p {
property: value;	color: blue;
}	}

2. **Class Selector :** Selects elements with a ***specific class attribute(.) .***

Example:

.highlight { background-color: yellow; } -> apply yellow background to all elements with the class highlight.

3. **ID Selector :** Targets a unique element with a ***specific ID(#)*** .

#header {font-size: 24px;}-> apply 24px font size to element in header.

4. **Universal Selector :** The universal selector (*) targets all elements in

the HTML document. It is a powerful tool for applying global styles but should be used judiciously due to performance considerations.

Syntax:

```
* {
  property: value;
  }
```

Example :

```
* {
 margin: 0;
padding: 0; }
```

- This is often used to reset default browser styles.

Q)Create a webpage with Type Selector and the Universal Selector

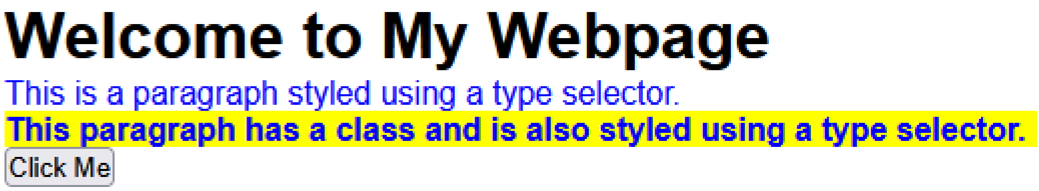
a. ***HTML (index.html)***

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>CSS Selectors Example</title>

  <link rel="stylesheet" href="styles.css">

</head>

<body>

  <h1>Welcome to My Webpage</h1>

  <p>This is a paragraph styled using a type selector.</p>
  <p class="highlight">This paragraph has a class and is also styled using a type selector.</p>

  <button>Click Me</button>

</body>

</html>
```

b. CSS (styles.css):

```
/* Universal Selector: Applies to all elements */

* {

  margin: 0;

  padding: 0;

  font-family: Arial, sans-serif;

}

/* Type Selector: Targets all <p> elements */

p {

  color: blue;

  font-size: 16px;

 font-color: red;

}

/* Class Selector: Targets elements with class "highlight" */

.highlight {
  background-color: yellow;

  font-weight: bold;

}
```

<u>Output :</u>



Welcome to My Webpage

This is a paragraph styled using a type selector.

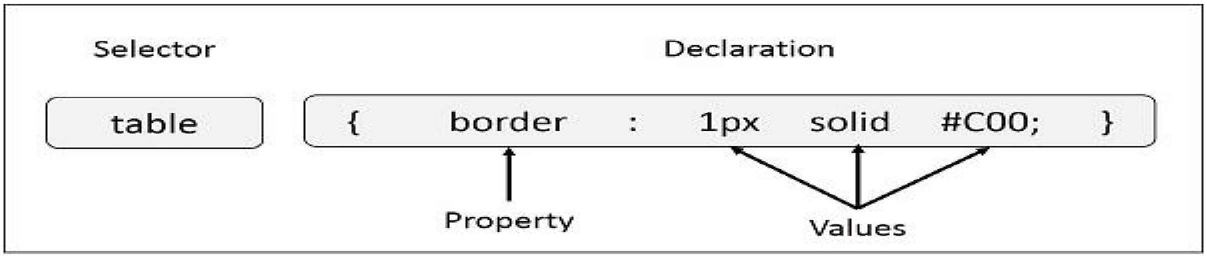
This paragraph has a class and is also styled using a type selector.

Q) Explain about css syntax and style

- ➔ CSS stands for Cascade Style Sheet is a popular stylesheet language used to design an interactive webpage.
- ➔ ***<u>CSS Syntax :</u>***

  selector { property: value; }

  - **Selector:** CSS **selectors** are used to select the HTML element or groups of elements to style on a web page.
  - **Property:** A CSS property is an aspect or characteristic of an HTML element that can be styled or modified using CSS, such as color, font-size, or margin.
  - **Value:** Values are assigned to properties. For example, color property can have value like red, green etc.



➔ <u>CSS Style:</u>

- CSS rules are enclosed in a style container. And the style container positioned at the bottom of the web page's head container.In the style start tag, it's legal to include a type attribute with a value of "text/css", like this:

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

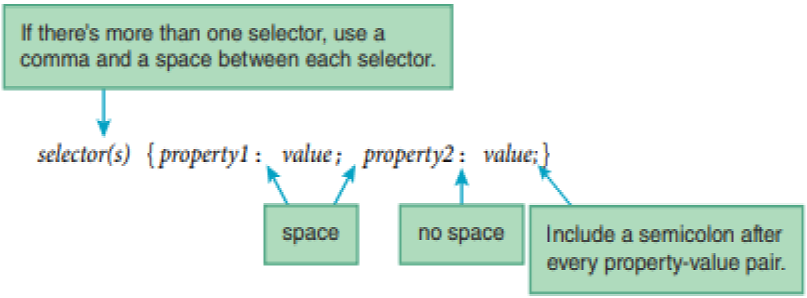
- Here "text/css" is the only legal value for the type attribute, and it's the default value for the type attribute.
- ***the syntax for the style container is :***

  ```
  <style> css rules </style>
  ```

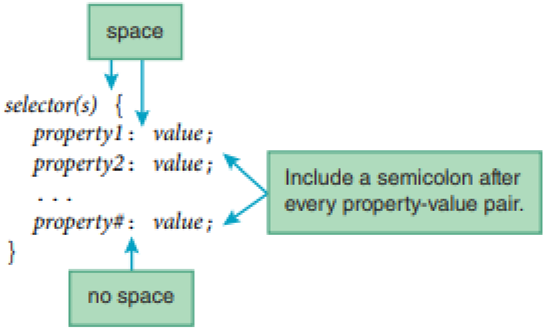
  <u>Example:</u>

  ```
  <style>
   * {text-align: center;}
   hr {width: 50%;}
   h2, p {font-style: italic; color: blue;}
  </style>
  ```

- They are style guidelines to follow to understand and maintain code easily. For short CSS rules, use this format:


If there's more than one selector, use a comma and a space between each selector.
selector(s) {property1 : value ; property2 : value; }
space
no space
Include a semicolon after every property-value pair.

- For long (at least two or three property-value pairs),then use block formatting like this:



Q)Write about Class selector ?

➔ The **CSS class selector** is a fundamental tool in web development, allowing users to apply styles to HTML elements based on their class attribute. This selector is versatile and widely used for styling multiple elements simultaneously.

➔ CSS class selectors are used to select elements with a specific class attribute. The class selector is defined using a **period (.)** followed by the **class name**.

➔ ***Syntax :***

```
.classname {
   property: value;
  }
```

➔ ***Example:***

```
.center {
  text-align: center;
color: red;
} ->This rule will apply the specified styles to all elements with class="center"
```

➔ ***Variations in class selector :***

1. **class Selectors with Element Type Prefixes**

- In CSS, combining **element type prefixes** with **class selectors** allows to target specific HTML elements that possess a particular class. This approach enhances the precision of styling, ensuring that only the desired elements are affected.

- More importantly, using a class selector with an element prefix makes the code more maintainable.
- **Syntax :** To select an element of a specific type that also has a particular class, the syntax is :

  ```
  element.class {
    /* CSS properties */
  }
  ```

  Example: Improved style container for Mark Twain Quotes web page

  ```
  <style>
          h1.blue {color: blue;}
          q.red {background-color: tomato;}
          q.white {background-color: white;}
          q.blue {background-color: skyblue;}
          q {font-family: Impact;}
  </style>
  ```

a) Class Selectors with * Prefixes :

Instead of prefacing a class selector with an element type, as an alternative, user can preface a class selector with an *. Because * is the universal selector, it matches all elements.

Syntax: for standard class selector rule (with no prefix):

```
 *.class-value  {                          Ex:  *.big-warning {
       property1:  value;                          font-size: x-large;
       property2:  value;                          color: red;
  }                                                }
```

Q) Explain about ID selectors in CSS ?

- ➔ An ID selector is similar to class selector but instead using element attribute values(class selector) ,it uses an element's id attribute.
- ➔ An id attribute value must be unique within a particular web page.
- ➔ The ID selector's unique-value feature means that an ID selector's CSS rule matches only one element on a web page.

- ➔ The ID selector mechanism is particularly helpful with links and with JavaScript.
- ➔ ***syntax:***

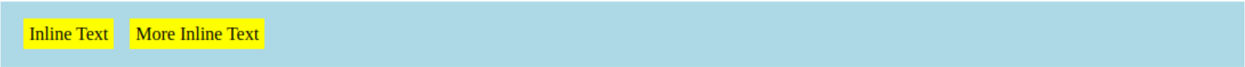
```
 #id-value {                        Ex:   #header {
          property1: value;               background-color: blue;
          property2: value;                color: white;
  }                                        }
```

- ➔ The syntax is the same as for a class selector rule, except pound sign (#) instead of a dot (.), and use id attribute value instead of class attribute value.

Q) Explain about Span and Div elements with an example ?

```
          <html>
           <head>
             <style>
                .block {
                           background-color: lightblue;
                           padding: 20px;
                           margin: 10px;
                }
                .inline {
                            background-color: yellow;
                            padding: 5px;
                           margin-right: 10px;
                }
        </style>
  </head>
  <body>
          <div class="block">inline
           <span class="inline">Inline Text</span>
           <span class="inline">More Inline Text</span>
          </div>  </body> </html>
```

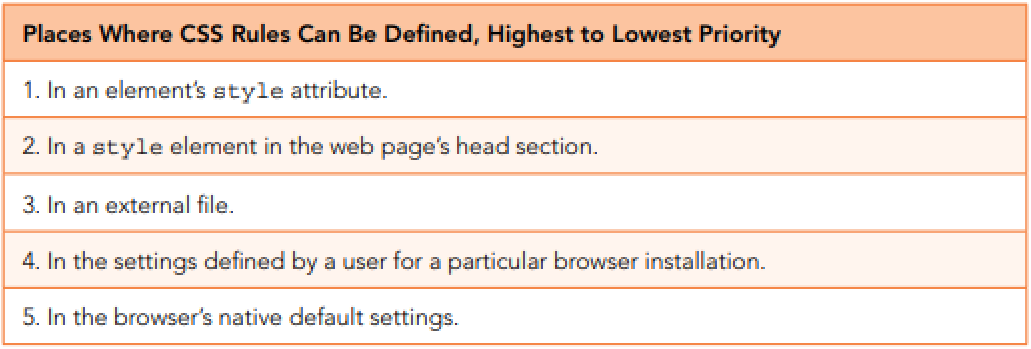
Output:



Inline Text More Inline Text

Cascading:

- ➔ The “cascading”means “a series of stages in a process.” Likewise, CSS uses a series of stages.
- ➔ More specifically, there are different stages/places where CSS rules can be defined. Each stage/place has its own set of rules, and each set of rules is referred to as a style sheet.
- ➔ With multiple style sheets organized in a staged structure, together it’s referred to as Cascading Style Sheets.
- ➔ To handle the possibility of conflicting rules at different places, different priorities are given to the different places.



Places Where CSS Rules Can Be Defined, Highest to Lowest Priority
1. In an element’s `style` attribute.
2. In a `style` element in the web page’s head section.
3. In an external file.
4. In the settings defined by a user for a particular browser installation.
5. In the browser’s native default settings.

1. The higher priority places is in an ***element’s style attribute***
2. The second place for CSS rules is in a ***style container***.
3. The next place for CSS rules is in an ***external file***.
4. The next place for CSS rules is in the settings defined by a user for a ***particular browser installation.***
5. The last place for CSS rules, and the place with the lowest priority, is in the ***native default settings*** for the browser that’s being used.

Q)difference between style attribute vs style container

(or)

Q)Explain about different types of css rules

1) ***style Attribute / inline styles :*** The style attribute is at the top of cascading CSS rules list;

- style attribute insert CSS property-value pairs directly in the code for an individual element.

- The style attribute is a global attribute i.e., it can be used with any element. Using the style attribute is used to be referred to as “**inline styles”** .
- To use inline CSS, add the style attribute to an HTML tag with the desired CSS properties.

  Syntax :

  ```
          <tag style="property: value;">Content</tag>
  ```

  Example :

  ```
          <p style="color: blue; font-size: 18px;">
          This is a blue paragraph with 18px font size. </p>
  ```

2) ***style Container / Internal Styles :***

- ➔ The style element is a container for CSS rules that apply to the entire current web page.The browser applies the CSS rules’ property values by matching the CSS rules’ selectors with elements in the web page.
- ➔ Normally there will be one style container per page, and that should be in the web page’s head container.
- ➔ Using the style element in the head section used to be referred to as “internal styles”.
- ➔ Internal or Embedded CSS is defined within the HTML document’s **<style> element**. It applies styles to specific HTML elements. The CSS rule set should be within the HTML file in the head section, i.e. the CSS is embedded within the <style> tag inside the head section of the HTML file.
- ➔ ***Syntax***

```
 <style>
        /* code*/
</style>
```

Example:

```
<html>
<head>
        <title>Internal CSS </title>
        <style>
                h1 { color : green ; }
```

```
        </style>
    </head>
    <body>
        <h1> welcome to internal </h1>
    </body>
</html>
```

3) __External css:__

- <u>External CSS</u> contains separate CSS files that contain only style properties with the help of tag attributes (For example class, id, heading, … etc).
- CSS property is written in a separate file with a .css extension and should be linked to the HTML document using a **link** tag. It means that, for each element, style can be set only once and will be applied across web pages.
- Syntax to link external css file:

  <link rel="stylesheet" href="name-of-external-file" **>**
- Here, href stands for “hypertext reference.” The href attribute’s value specifies the name of the file that holds the CSS rules.
- rel stands for “relationship,” and rel=”stylesheet” tells the browser engine to look for CSS rules in the href file and apply them to the current web page.
- **Example:**

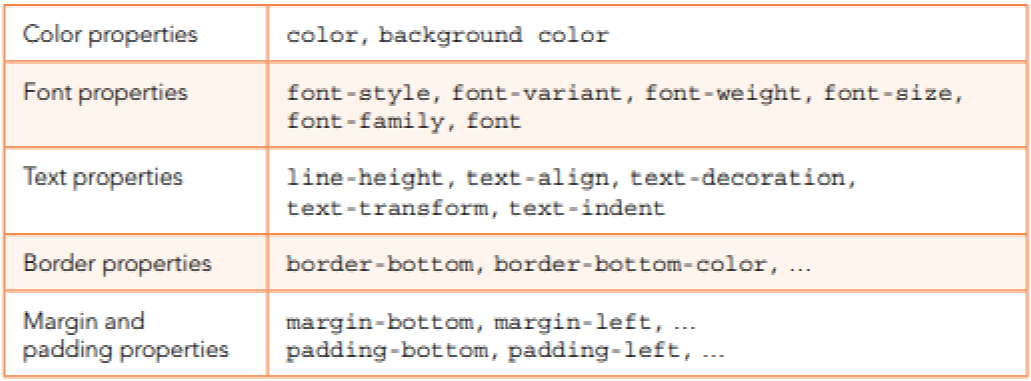
  ```
  <!DOCTYPE html>
  <html>
      <head>
          <link rel="stylesheet" href="styles.css">
      </head>
      <body>
          <h1>This is a heading</h1>
          <p>This is a paragraph.</p>
      </body>
  </html>
  ```

  Css file(with .css)

  ```
      body {  background-color: powderblue; }
  ```

Q) What are the different types of properties available in css ?

- ➔ A CSS property specifies one aspect of an HTML element’s appearance. The W3C’s CSS3 standard provides many CSS properties;all those properties have types of values associated with them.
- ➔ Those properties fall into five property groups — color, font, text, border, and margin/padding.



Color properties	`color, background color`
Font properties	`font-style, font-variant, font-weight, font-size, font-family, font`
Text properties	`line-height, text-align, text-decoration, text-transform, text-indent`
Border properties	`border-bottom, border-bottom-color, ...`

Margin and padding properties	`margin-bottom, margin-left, ...`
	`padding-bottom, padding-left, ...`

a) Color Properties

- There are two color properties—***color*** and ***background-color.***
- The color property specifies the color of an element’s text.
- The background-color property specifies the background color of an element.
- The color properties are straightforward but the values for the color properties require more attention. Color value can be specified using one of five different formats :
  1. color name—for example, red
  2. RGB value—specifies amounts of red, green, and blue
  3. RGBA value—specifies red, green, and blue, plus amount of opacity
  4. HSL value—specifies amounts of hue, saturation, and lightness
  5. SLA value—specifies hue, saturation, and lightness, plus amount of opacity

1. **Color Names:** The CSS3 specification defines 147 color names, and the major browsers support all those colors. Examples of color names are orange, darkorange, darkslategray.

Example:

```
<style>
 .roofColor {color: darkslategray;}
</style>
```

2. RGB Values for Color : RGB stands for red, green, and blue.

- An RGB value specifies the amounts of red, green, and blue that mix together to form the displayed color.
- To specify an amount of a color, use a ***percentage***,an ***integer***,or a ***hexadecimal*** number .
- The allowable ranges for each technique:
    I. percentage—0% to 100% for each color
    II. integer—0 to 255 for each color
    III. hexadecimal—00 to ff for each color

I. ***RGB Values with Percentages*** : To specify an RGB value with percentages, use this format: rgb (red-percent, green-percent, blue-percent) .Each percent value must be between 0% and 100%.

Example: class selector rule that uses an RGB value with percentages:

```
<style>
 .eggplant { background-color: rgb (52%,20%,45%) ; }
</style>
```

- To specify black use the least intensity (a value of 0%) for each of the three colors. I.e., **.black { background-color : rgb(0%,0%,0%); }**
- To specify white,use the greatest intensity (a value of 100%) for each of the three colors. I.e., **.white { color: rgb(100%,100%,100%); }**

II. ***RGB Values with Integers :*** To specify an RGB value with integers, use the format: rgb(red-integer,green-integer,blue-integer)

- Each integer value must be between 0 and 255, with 0 providing the least intensity and 255 providing the most.

- ***Example:***

  ```
  <style>
   .favorite1 {color: rgb(144,238,144);}
  .favorite2 {color: rgb(127,127,127);}
  </style>
  ```

III. ***RGB Values with Hexadecimal :*** With many programming languages, including HTML and CSS, numbers can be represented not only with base-10 decimal numbers, but also with base-16 hexadecimal (hex) numbers.

- For hexadecimal RGB values, use the format #rrggbb where:
  rr = two hexadecimal digits that specify the amount of red
  gg = two hexadecimal digits that specify the amount of green
  bb = two hexadecimal digits that specify the amount of blue
- The smallest hexadecimal digit is 0, so 00 represents the absence of a particular color. If all colors are absent, that's black. Therefore, #000000 (00 for each of the three colors) indicates black.
- The largest hexadecimal digit is f, so ff represents the greatest intensity of a particular color. If all colors are maximally intense, that's white. Therefore, #ffffff (ff for each of the three colors) indicates white.
- ***Example:***

  ```
  <style>
   .sapphire { background-color : #0f42ba ; }
  </style>
  ```

- Note : RGB hexadecimal digits are case insensitive.

3. RGBA Value for color :

- The A in RGBA stands for alpha. To specify an RGBA value:
  -rgba(red-integer,green-integer,blue-integer,opacity-number-between-0-and-1)
  -rgba(red-percent,green-percent,blue-percent,opacity-number-between-0-and-1)
- For the first format, each integer value must be between 0 and 255, with 0 providing the least intensity and 255 providing the most.
- For the second format, each percent value must be between 0% and 100%.

- For both formats, the fourth value specifies the opacity. The opacity value must be in the form of a decimal number between 0 and 1, with 0 being completely transparent, 1 being completely opaque, and .5 in between .

  <u>Opacity value for color :</u>

  - ➔ The **opacity** in CSS is the property of an element that describes the transparency of the element. It determines how opaque the color is, where opaque refers to the inability to see through something. It’s the opposite of transparency.
  - ➔ If the opacity value is 100%, that means the color is completely opaque, and if there is content behind the color, that content gets covered up.
  - ➔ If the opacity value is 0%, that means the color is completely transparent.
- ***Example :*** (A transparent yellow color is placed on top of a red background so that orange is formed)***.***

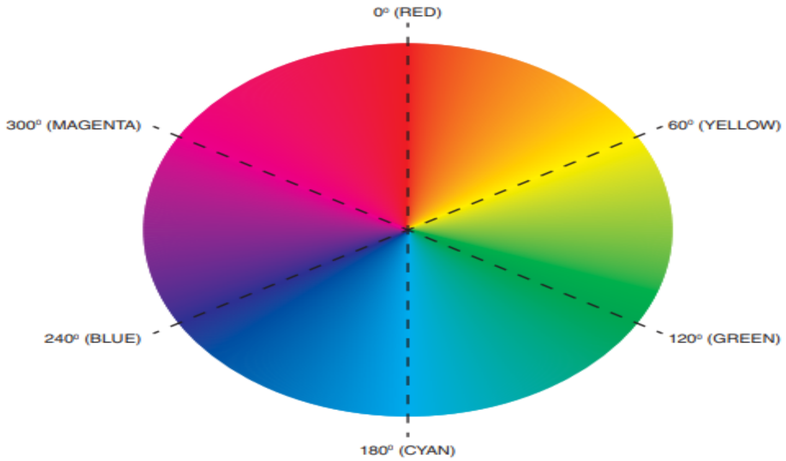
```
<html lang="en">
<head>
        <style>
           .red {background-color: red;}
          .yellow-bg2 {background-color: rgba( 255, 255, 0, .4);}
        </style>
</head>
<body class="red">
     <p class="yellow-bg">This line uses background-color: rgb(255,255,0);</p>
     <p class="yellow-bg2">
                This line uses background-color: rgba(255,255,0,.4);
        </p>
</body>
</html>
```

4. **HSL and HSLA Values for Color :**

- HSL stands for hue, saturation, and lightness.

  syntax: hsl(hue-integer,saturation-percent,lightness-percent) .

- Hue is a degree on the color wheel, so the wheel's degrees go from 0 to 360. 0 degrees is for red, 120 degrees is for green, and 240 degrees is for blue. For a circle, 0 degrees is equivalent to 360 degrees. So, to specify red, you can use 360 as an alternative to 0.



3URE 3.15 **Color wheel for the `hsl` construct's hue value**

- The second value in the hsl construct is the color's percentage of saturation. 0% means a shade of gray, and 100% is the full color.
- The third value in the hsl construct is the color's percentage of lightness. A lightness value of 0% generates black, regardless of the values for hue and saturation. A lightness value of 100% generates white, regardless of the values for hue and saturation. A lightness value of 50% generates a "normal" color.
- ***Example:*** (It forms a light shade of grayish green)

```
<style>
 p {background-color: hsl(120,50%,75%);}
</style>
```

- Likewise rgba construct, to add transparency to an HSL value, use the hsla construct.
  - ➔ ***syntax:***

    hsla(hue-integer,saturation-percent,lightness-percent,opacity-number-between-0-and-1)
  - ➔ ***Example:*** (a lighter shade of grayish green)

```
<style>
 .background {background-color: hsla(120,50%,75%,.5);} </style>
```

Q)Explain about CSS Font Properties ?

- These properties describe how to display text with different font characteristics.Font refers to the characteristics of text characters—height, width, thickness, slantedness,body curvatures, and endpoint decorations.
- There are many CSS font properties available like font-style, font-variant, font-weight, font-size, font-family, and font shorthand properties.

1. ***font-style property :*** It specifies whether the text is to be displayed normally or slanted. Valid values for the font-style property :

Font-style values	**Description**
normal	Upright characters (not slanted)
oblique	Use the same font as the current font, but slant the characters.
italic	Use a cursive font (which tends to be slanted and is supposed to look like handwriting).

   - ➔ To generate italic text use : {font-style: italic;}
   - ➔ To generate normal (not italicized) text use : { font-style: normal;}
   - ➔ The W3C provides default values for all CSS properties, and to force the default value to be used for a particular property, then specify ***initial*** for that property.

     Ex: .not-italicized {font-style: initial;}

2. ***font-variant Property :*** It specifies how lowercase letters are displayed. The valid values for the font-variant property:

Font-variant Values	**Description**
Normal	Display lowercase letters normally.
small-caps	Display lowercase letters with smaller-font uppercase letters.

   - ➔ ***Example*** : .title {font-variant: small-caps;} ...

3. ***font-weight Property :*** It specifies the boldness of the text characters. The valid values for the font-weight property:

font-weight Values	**Description**
normal, bold	It’s up to the browser to determine a font weight that can be described as normal or bold.
bolder, lighter	Using a value of bolder causes its targeted text to have thicker characters than the text that surrounds it. Using a value of lighter causes its targeted text to have thinner characters than the text that surrounds it.
100, 200, 300, 400, 500, 600, 700, 800, 900	100 generates the thinnest characters and 900 the thickest characters. 400 is the same as normal, and 700 is the same as bold.

Ex: technique for making something bold

.bold {font-weight: bold;}

4. ***font-size Property :*** It specifies the size of the text characters.The values allowed for the font-size property :

Font-size values	**Description**
xx-small, x-small, small, medium, large, x-large, xx-large	It’s up to the browser to determine a font size that can be reasonably described as ***xx-small, x-small, small***, etc.
smaller, larger	- Using a value of ***smaller*** causes its targeted text to have smaller characters than the text that surrounds it. - Using a value of ***larger*** causes its targeted text to have larger characters than the text that surrounds it.
number of em units	One em unit is the height of the element’s normal font size.

➔ ***Ex: (*** font-size property with em values)

.disclaimer {font-size: .5em;} ⇒ ***.5em*** value displays text that is half the size of normal text

.advertisement {font-size: 3em;} ⇒ ***3em*** value displays text that is three times the size of normal text.

➔ ***Note:*** For each Font-size value, there is no blank space separating the number from em.

5. ***font-family Property :*** It allows the web developer to choose the set of characters that the browser uses when displaying the element's text.
   - ➔ font-family properties normally have a comma-separated list of fonts. In applying the precedence rule with ***`asci-art`*** `class attribute`, the browser processes the font stack from left to right.i.e.,it uses the first font in the list available on the user's system and skips the others if the first one is found.
   - ➔ The font names(title case) that use all lowercase letters are special, and they are known as generic fonts. A generic font is a name that represents a group of fonts that are similar in appearance. For example, monospace is a generic font, and it represents all the fonts where each character's width is uniform.
     - Whenever using a font-family CSS rule , include a generic font at the end of the rule's list of font names.
       Ex: .ascii-art {font-family: Courier, Prestige, monospace;}
     - The ***monospace*** font is one of five generic fonts that all browsers recognize and support. List of generic fonts with descriptions :

Generic Font Names	**Description**
monospace	All characters have the same width.
serif	Characters have decorative embellishments on their endpoints.
sans-serif	Characters do not have decorative embellishments on their endpoints.
cursive	Supposed to mimic cursive handwriting, such that the characters are partially or completely connected
fantasy	Supposed to be decorative and playful.

- ➔ When specifying a multiple-word font name, surround the name with quotes.
- ➔ ***Example:***

```
<style>
 blockquote {font-family: "New Century Schoolbook", Times, serif;} </style>
```

6. ***font Shorthand Property :***

- ➔ The ***font*** property as a shorthand property because it’s a timesaving construct for handling multiple font characteristics.i.e., can be used to apply more than one of the prior font-related properties to an element.
- ➔ Syntax for a font property-value pair:
  font: [font-style-value] [font-variant-value] [font-weight-value] font-size-value[/line-height-value] font-family-value
  - ➢ The bracketed thing is optional,whereas non bracketed must be included whenever using the font property.
- ➔ Use spaces to separate property values and use commas(,) to separate ***font-family*** values.
- ➔ ***Example:***

```
blockquote {
font: italic large "Arial Black", Helvetica, sans-serif;
}
```

Q)Explain about line-height property ?

- ➔ ***line-height Property :*** It’s used to specify the vertical separation between each line of text in an element.
  Example : .sentence1 {line-height: 2em} ⇒ display its lines with a vertical separation equal to twice the height of a normal character
- ➔ Line-height property is also used with shorthand ***font*** property, the / separates a font-size value from a line-height value.
  Ex: .sentence2 {font: 1em/2em Times, serif;}
- ➔ **Example:**

```
<html>
<head>
        <style>
         .sentence1 {line-height: 2em; font-family: Times, serif;}
         .sentence2 {font: 1em/2em Times, serif;}
        </style>
</head>
```

```
<body>
<p>
 <span class="sentence1">We hold these truths to be self-evident
</span>
 <br>
 <span class="sentence2">Liberty and the pursuit of Happiness.</span>
</p>
</body>
</html>
```

➔ As an alternative to using an em value for the line-height property, you can use a percentage value. The percentage value is relative to the element's font size .

Example: .proof-reading {line-height: 300%;}

Q) Explain about css text properties ?

The font properties focused on appearance characteristics of individual characters. The text properties focus on appearance characteristics of groups of characters —text-align, textdecoration, text-transform, and text-indent.

a) ***text-align Property*** : The text-align property specifies the horizontal alignment for a block of text. The valid values for the text-align property are :

text-align Values	**Description**
left	Align the text at the left.
right	Align the text at the right.
center	Center the text.
justify	Stretch the lines so that each line extends to the left edge and the right edge.

b) ***text-decoration Property :*** It specifies something decorative that is added to text. The valid values for the text-decoration property are:

Example : .underlined {text-decoration: underline;}

text-decoration Values	**Description**
none	This displays normal text (no decoration added).
underline	Draw a line below the text.
overline	Draw a line above the text.
line-through	Draw a line through the text.
blink	This causes the text to blink.

c) ***text-transform Property :*** This property controls the text's capitalization. The valid values for the text-transform property are :

text-transform Values	**Description**
none	The text renders the same as the original text.
capitalize	Transform the first character of each word to uppercase.
uppercase	Transform all characters to uppercase.
lowercase	Transform all characters to lowercase

d) ***text-indent Property :*** This property specifies the size of the indentation of the first line in a block of text. The block's second and third lines (and so on) are unchanged; i.e., they do not get indented.

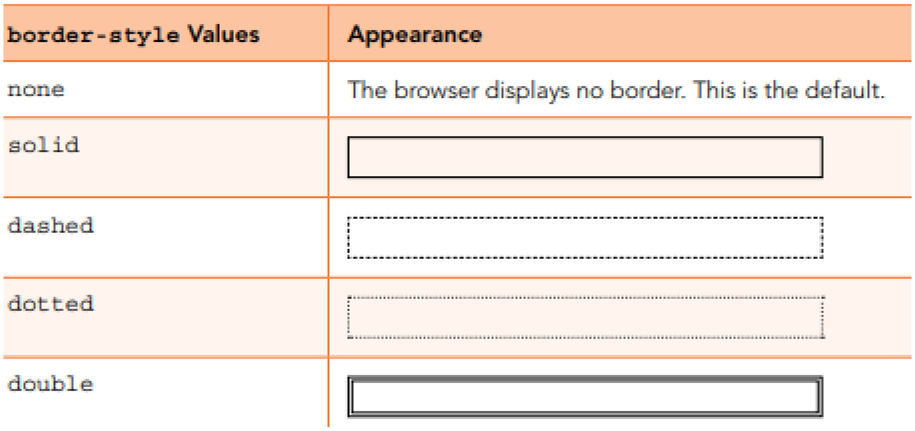
- ➔ If you want to adjust all the lines in a block of text, use the margin property, not the text-indent property.
- ➔ The most appropriate way to specify a value for the text-indent property is to use em units.
- ➔ ***Example :*** type selector rule that uses the text-indent property:

  p {text-indent: 4em;}

Q)Explain Border properties ?

Border Properties are used to specify the appearance of borders that surround elements. It includes many border properties—border-style, border-width, and border-color properties.

a) ***border-style Property :*** This property specifies the type of border that surrounds the matched element.The valid values for the border-style property:



border-style Values	Appearance
none	The browser displays no border. This is the default.
solid	
dashed	
dotted	
double	

Example:.coupon {border-style: dashed;} ⇒ to draw a dashed border

b) ***border-width Property :*** It specifies the width of the border that surrounds the matched element. There are few values allowed for the border-width property.

border-width Values	**Description**
thin, medium, thick	The browser determines a border width that can be reasonably described as thin, medium, or thick. The default is medium.
number of px units	A CSS pixel unit is the size of a single projected dot on a computer monitor when the monitor's zooming factor is at its default position of 100%.

- ➔ Use the border-width property with the border-style property. If border-style property is not specified , then the default border-style value none i.e, With a border-style value of none, no border will be displayed.
- ➔ CSS pixel values use px units. As with all the other CSS size values, CSS pixel values are relative to the monitor's resolution .
- ➔ If four values are specified for the border-width property,then the four values get applied to the border's four sides in ***clockwise*** order, starting at the top.

  Example: .boxed { border-style: solid; border-width: 4px 2px 0px 2px; }
- ➔ If three values are specified to the border-width,then the first value applies to the top side, the second value applies to the left and right sides, and the third value applies to the bottom side.

Example: .three { border-style : dashed; border-width : 4px 2px 0px; }

- ➔ If you specify just two values, then the first value applies to the top and bottom sides and the second value applies to the left and right sides.

c) ***border-color Property :*** It specifies the color of the border that surrounds the matched element.

- ➔ There's no syntax for the border-color property because it uses the same values as the color property and the background-color property. For the border-color property to work, use it in conjunction with a border-style property.
- ➔ In order to change the border's color or change the border's width,it must have a visible border, and that's done by using a border-style property.

d) ***border Shorthand Property :*** It is used to apply more than one of the prior border-related properties to an element.

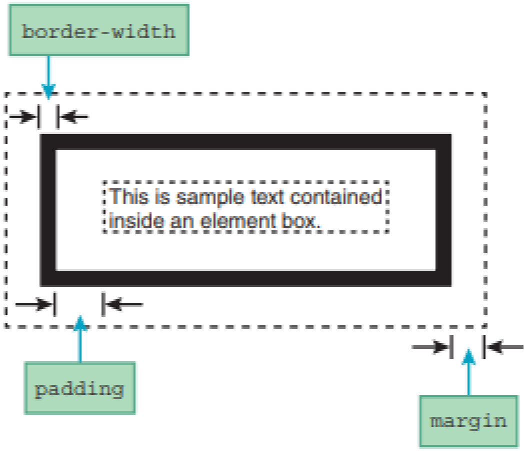
- ➔ The border property is a shorthand notation for specifying a border's width, style, and color in that order.
- ➔ Here are two examples:

  .understated-box {border: thin dotted blue;}
  .in-your-face-box {border: 10px solid;}

  Note : color can be omitted but must include style value.

q) **Element Box, padding Property, margin Property**

- Usually, borders have no gaps inside or outside of them. To introduce gaps around an element's border, then use the element box. Every web page element has an element box associated with it.
- An element box has a border, padding inside the border, and a margin outside the border. For most elements, but not all, the default border, padding, and margin widths are zero. We can adjust the widths with the border-width, padding, and margin properties.
- The dashed lines indicate the perimeters of the margin and padding areas. When a web page is displayed, only the border can be made visible;
- The padding property is for inside the border. Therefore, the margin property must be for outside the border.



An element box's margin, border, and padding

a) ***padding and margin Properties :***

- ➔ The padding property specifies the width of the area on the interior of an element's border, whereas the margin property specifies the width of the area on the exterior of an element's border.
- ➔ Usually, the most appropriate type of value for the padding and margin properties is a CSS pixel value.
- ➔ ***Example :***

```
 .label {border: solid; padding: 20px; margin: 20px;}
```

- ➔ Just as with the border-width property,different padding widths can be specified for the four different sides —***padding-top, padding-right, padding-bottom, and padding-left.***
- ➔ Likewise, different margin widths can be specified for the four different sides—***margin-top, margin-right, margin-bottom, and margin-left.***
- ➔ The margin and padding properties allow -values. While a +value forces two elements to be separated by a specified amount, a -value causes two elements to overlap by a specified amount.
- ➔ ***Example:***

```
<html lang="en">
<head>
<style>
 .hot {background-color: red;}
 .cold {background-color: blue;}
 .label {
          color: white;
           font: bold xx-large Lucida, monospace;
           border: solid black;
           border-radius: 50%;
           padding: 20px;
           margin: 20px;
           display: inline-block;  }
</style>
</head>
<body>
<span class="hot label">HOT</span>
<span class="cold label">COLD</span>
</body>
</html>
```