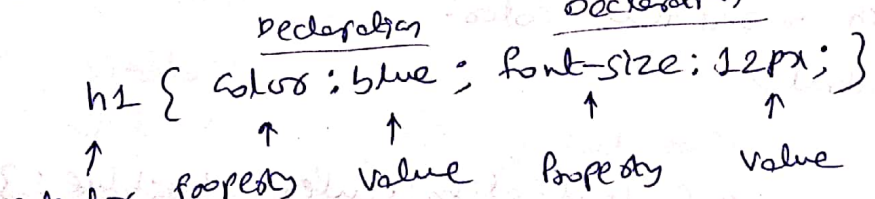
Cascading Style Sheet (CSS):- CSS describes how HTML elements are displayed on screen, paper, or in other media. CSS saves a lot of work and it can control the layout of multiple webpages all at once.

→ It is used to describe the look and formatting of a document written in markup language. We can add new looks to our old HTML documents.

→ It can also be used with HTML to change the style of web pages and user interfaces. We can completely change the look of our website with only a few changes in CSS code.

→ Before CSS, tags like font, color, background style, element alignments, border and size had to be repeated on every webpage. CSS was created to solve this problem. CSS provides more detailed attributes than plain HTML to define the look and feel of the website.

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



→ The selector indicates the HTML element you want to style. The declaration block can contain one or more declarations separated by a semi colon. Above example, there are two declarations.

→ Each declaration contains a property name and value, separated by a colon. A property is a type of attribute of HTML element. It could be color, border etc. Values are assigned to CSS property. Declaration blocks are surrounded by curly braces.

→ In this example all <P> elements will be center-aligned, with a red text color.

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

→ 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.

```
Ex:  P { color: red;
       /* This is a comment */
       text-align: center;
     }
```

→ CSS Selectors:- CSS selectors are used to select the content you want to style. Selectors are the part of CSS rule set. CSS selectors select HTML elements according to its id, class, type, attribute etc

→ There are several different types of selectors in CSS. They are
1) CSS Element Selector 2) CSS Id Selector 3) CSS class selector.
4) CSS Universal selector 5) CSS Grouping Selector

→ 1) CSS Element selector:- It selects HTML elements based on the element name. For example, all <P> elements on the page will be center-aligned, with a red text color.

```
Ex. <! DOCTYPE html>
    <HTML>
     <head> <style> P { text-align: center; color: blue; }
            </style> </head>
     <body> <p> This style will be applied on every paragraph </p>
            <p id = "para1"> Me too! </p>
            <p> And me! </p>
     </body> </html>
```

→ 2) 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.

```
<!DOCTYPE html>
<html> <head>
<style> #para1 { text-align:center; color:blue; } </style>
</head><body>
<p id="para1">Hello Java </p>
<p> This paragraph will not be affected </p>
</body>
</html>
```

3) 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.

```
<!DOCTYPE html>  <!-- class name should not start with number -->
<html> <head>
<style> .center { text-align: center; color:blue; } </style>
</head> <body>
<h1 class="center">This heading is blue and center aligned </h1>
<p class="center">This paragraph is blue and center-aligned</p>
</body>
</html>
```

→ If you want to specify that only one specific HTML element should be affected then you should use the element-name with class selector.

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

4) CSS Universal selector:- The universal selector is used as a wild character. It selects all the elements on the pages.

```
<!DOCTYPE html>
<html><head>
<style> * { color:green; font-size:20px; } </style>
</head><body>
<h2> This is heading </h2>
<p> This style will be applied on every paragraph </p>
<p id="para1"> me too! </p>
<p> and me! </p>
</body>
</html>
```

5) CSS Grouping selector:- It selects all the HTML elements with the same style definitions. It will be better to group the selectors, to minimize the code. To group selectors, seperate each selector with a comma.

```
<!DOCTYPE html>
<html><head>
<style> h1,h2,p { text-align:center; color:blue; } </style>
</head><body>
<h1> Hello </h1> <h2> Welcome </h2> <h3> HTML </h3>
<p> This is paragraph </p>
</body>                    <!-- h3 will not be effected -->
</html>
```

→ How to Add CSS:- CSS is added to HTML pages to format the document according to information in the style sheet. There are three ways to insert the style sheets in HTML. 1) Inline CSS 2) Internal CSS 3) External CSS.

→ 1) Inline CSS:- An inline style may be used to apply a unique style for a single element. To use inline styles, add the style

attribute to the relevant element. The style attribute can contain any CSS property.

Syntax: <htmltag style="cssproperty1:value; cssproperty2:value;"></htmltag>

Ex: <h2 style="color:red; margin-left:40px;">
Inline CSS is applied on this heading. </h2>
This paragraph is not affected.

Disadvantages of Inline CSS:- 1) You cannot use quotations within inline CSS. If you use quotations the browser will interpret this as an end of your style value. 2) These styles cannot be reused anywhere else. 3) These styles are tough to be edited because they are not stored at a single place. 4) It is not possible to style pseudo-codes and pseudo-classes with inline CSS. 5) Inline CSS does not provide browser cache advantages.

2) Internal CSS:- It is used to add a unique style for a single document. It is defined in <head> section of the HTML page inside the <style> tag.

```
<!DOCTYPE html>
<html> <head>
<style> body { background-color: linen; }
        h1 { color:red; margin-left: 80px; }
</style> </head> <body>
<h1> The internal style sheet is applied on this heading </h1>
<p> This paragraph will not be affected </p>
</body>
</html>
```

3) External CSS:- The external style sheet is generally used when you want to make changes on multiple pages. It is ideal for this condition because it facilitates you to change the look of the entire website by changing just one file.

→ It uses the <link> tag on every pages and the <link> tag should be put inside the head section.

Ex:- <head> <link rel="stylesheet" type="text/css" href="mystyle.css"> </head>

→ The external style sheet may be written in any text editor but must be saved with a .css extension. This file should not contain HTML elements. Name of the file is "myfile.css",

```
body { background-color : lightblue; }
h1 { color: navy; margin-left: 20px; }
```

→ Note: You should not use a space between the property value and the unit for example, it should be margin-left: 20px not margin-left: 20 px.

→ ④ @import Style sheet:- It is another way to loading a CSS file. @import CSS style define within <style type="text/css">... </style> element in your <head>...</head> of your webpage.

→ It consists of 3 steps. step1 is @import (keyword), step2 is url() and step3 is CSS file path.

→ Ex① /* style.css */

```
p { color: purple; margin-left: 20px; }
div { color: purple; font-size: 16px; background-color: #ff6633; }
```

→ <!-- HTML Program to import style.css file -->

```
<!DOCTYPE html>
<html><head> <title>Import CSS Style </title>
    <style> @import url("style.css"); </style>
</head> <body>
  <p> This is a first paragraph </p>
  <div> This is a second paragraph </div>
  <h1> This is a third paragraph </h1>
</body> </html>
```

A) CSS Background:- CSS background property is used to define the background effects on element. There are 5 CSS background properties that affects the HTML elements.

1) CSS background-color:- The CSS background-color property is used to specify the background color of the element

Ex:- h2, p { background-color: #b04dde; }

→ With CSS, a color is most often specified by: 1) A valid color name like "lightblue". 2) A HEX value like - "#ff0000". 3) An RGB value like rgb(255,0,0).

2) CSS background-image:- The CSS background-image property is used to set an image as a background of an element. By default, the image covers the entire element. You can set the background image for a page like this.

```
<!DOCTYPE html>
<html> <head>
<style> body { background-image: url("paper.gif");
                margin-left: 100px; } </style></head>
<body> <h1> Hello CSS </h1> </body>
</html>
```

→ Note: The background image should be chosen according to text color. The bad combination of text and background image may be a cause of poor designed and not readable webpage.

3) CSS background-repeat:- By default, the background-image property repeats the background image horizontally and vertically. The background looks better if the image repeated horizontally only.

```
body { background-image: url("pic1.png");
       background-repeat: repeat-x; } // To repeat horizontally.
```

To repeat image vertically, background-repeat: repeat-y;

4) CSS background-attachment:- This property is used to specify if the background image is fixed or scroll with the rest of the page in browser window. If you set fixed the background image then the image will not move during scrolling in the browser.

Ex: background: white url('bbb.gif');
background-repeat: no-repeat;
background-attachment: fixed;

5) CSS background-position:- This property is used to define the initial position of the background image. By default, the background image is placed on the top-left of the web page. You can set the following positions: center, top, bottom, left and right.

Ex:- Position the background image in the top-right corner:

```
body
{
    background-image: url("img_tree.png");
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-position: right top;
}
```

→ Note: It is possible to specify all the background properties in one single property. This is called a shorthand property.

Ex: body { background: #ffffff url("img.png") no-repeat fixed right top ~~fixed~~; }

→ When using the shorthand property the order of the property values is: background-color, background-image, background-repeat, background-attachment, background-position.

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

8. CSS Border :- These properties are used to specify the style, color and size of the border of an element. The CSS border properties are border-style, border-color, border-width, border-radius.

1) CSS Border-style :- It is used to specify the border type, which you want to display on the webpage

```
<!DOCTYPE html>
<html><head><style>
p.none {border-style: none;}
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.hidden {border-style: hidden;}
</style></head><body>
<p class="none">No border</p>
<p class="dotted">A dotted border</p>
<p class="dashed">A dashed border</p>
<p class="solid">A solid border</p>
<p class="double">A double border</p>
<p class="groove">A groove border</p>
<p class="ridge">A ridge border</p>
<p class="inset">An inset border</p>
<p class="outset">An outset border</p>
<p class="hidden">A hidden border</p>
</body></html>.
```

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

→ The border-width property is not used alone. It is always used with other border properties like "border-style" property to set the border first otherwise it will not work.

Ex:
```
<!DOCTYPE html>
<html>
<head><style>
p.one {border-style: solid; border-width: 5px; }
p.two { border-style: solid; border-width: medium; }
p.three { border-style: solid; border-width: 1px; }
</style></head><body>
<p class="one">Write your text here</p>
<p class="two">Write your text here</p>
<p class="three">Write your text here</p>
</body></html>
```

3) CSS Border Color:- The border-color property is used to set the color of the four borders. The color can be set by a) Name (Specify the colorname like "red"). b) HEX (specify a hex value like "#ff0000"). c) RGB (specify a RGB value, like "rgb(255, 0, 0)").

→ The border-color property can have from one to four values (for the top border, right border, bottom border, and the left border) If border-color is not set, it inherits the color of the element.

Ex:
```
p.one { border-style: solid; border-color: red; }
p.two {border-style: solid; border-color: "#ff0000"; }
p.three {border-style: solid; border-color: red green blue yellow; }
```

→ The border-color property is always used with border-style

The border property is a shorthand property for the following individual border properties: a) border-width b) border-style (required). c) border-color.

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

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

Ex: 1) P { border-left: 6px solid red; background-color: lightgrey; }

Ex: 2) P {border-bottom: 6px solid red; background-color: lightgrey; }

→ The border-radius property is used to add rounded borders to an element.

Ex: P {border: 2px solid red; border-radius: 5px; }

c. CSS Display:- It is used to control the layout of the element. It specifies how the element is displayed. Every element has a default display value according to its nature. Every element on the webpage is a rectangular box and the CSS property defines the behaviour of that rectangular box.

→ There are the following CSS display values which are commonly used 1) display: inline 2) display: inline-block 3) display: block 4) display: run-in 5) display: none

→ 1) display: inline :- The inline element takes the required width only. It does not force the line break, so the flow of text doesn't break in inline The inline elements are , , , etc.

Ex: <html> <head> <style> p { display: inline; } </style>
</head> <body>
 Hello Javatpoint.   Java Tutorial.
 SQL Tutorial.   CSS Tutorial.
</body> </HTML>

output: HelloJavatpoint. Java Tutorial. SQL Tutorial. CSS Tutorial.

2) display: inline-block :- The CSS display: inline-block element is very similar to inline element but the difference is that you are able to set the width and height.

3) display: block :- The CSS display block element takes as much as horizontal space as they can. Means the block element takes the full available width. They make a line break before and after them.

4) display: none :- The none value totally removes the element from the page. It will not take any space.

Ex: <html><head><style> h1.hidden { display: none; }</style>
</head><body> <h1> This heading is visible </h1>
<h1 class = "hidden"> This is not visible </h1>
 You can see that hidden heading does not contain any space
</body>
</html>

⑥ CSS Float :- The CSS float property is a positioning property. It is used to push an element to the left or right, allowing other element to wrap around it. It is generally used with images & layouts.

The clear property is used to avoid elements after the floating elements which flow around it.

→ The float property can have one of the following values:

a) left - The element floats to the left of its container.

b) right - The element floats to the right of its container

c) none - The element does not float (will be displayed just where it occurs in the text). This is default.

d) inherit :- The element inherits the float value of its parent.

Ex:- <html><head><style> img {float: right;} </style></head>
<body> The following paragraph contains an image with style
 float: right The result is that the image will float
to the right in the paragraph
Hello CSS </body>

(E) CSS Font:- CSS font property is used to control the look of texts. By the use of CSS font property you can change the text size, color, style and more.

1) Font color:- It is used to change the color of the text (stand-alone attribute). There are three different formats to define a color. a) By a color name b) By hexadecimal value c) By RGB.

Ex: <style> body { font-size: 100%; }
h1 { color: red; }
h2 { color: #9000A1; }
p { color: rgb(0, 220, 98); }
</style>

2) font size:- CSS font size property is used, to change the size of the font. There are the possible values that can be used to set the font size. font size values are xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, size in pixels or percentage.

Ex: a) <P style="font-size: xx-small;"> Extremely small. </P>
b) size is larger
c) size is set on 200%
d) size is 20 pixels.

(default size for normal text, like paragraphs, is 16 px (16 px = 1 em))

3) font-family:- CSS font family can be divided in two types:
a) Generic family (It includes serif, sans-serif, and Monospace)
b) Font family (It specifies the font family name like Arial, Times New Roman etc).

Ex: <style> body { font-size: 100%; }
h1 { font-family: sans-serif; }
h2 { font-family: monospace; }
</style>

4) CSS Font Style:- CSS font style property defines what type of font you want to display. It may be italic, normal, or oblique. (Oblique is very similar to italic, but less supported).

Ex: <style> body { font-size: 100%; }
h2 { font-style: italic; }
h3 { font-style: oblique; }
h4 { font-style: normal; }
</style>

5) font variant:- The font-variant property specifies whether or not a text should be displayed in a small-caps font. In small caps font, all lower case letters are converted to upper case letters. However, the converted uppercase letters appears in a smaller font size than the original uppercase letters in the text.

Ex: <style> p { font-variant: small-caps; }
h3 { font-variant: normal; }
</style>

6) font weight:- It defines the weight of the font and specify that how bold a font is. The possible values of font weight may be normal, bold, bolder, lighter, or number (100, 200, ... upto 900).

Ex:- Font is lighter
 Font is 700 weight

3) CSS Height and Width:- The height and width properties are used to set the height and width of an element. The height and width properties do not include padding, borders, or margins. It sets the height/width of the area inside the padding, border, and margin of the element.

→ The height and width properties may have the following values: a) auto (this is default. Browser calculates the height & width)

b) length (defines the height/width in px, cm etc). c) % (defines the height/width in percent of the containing block d) initial (sets the ~~default~~ height/width to its default value). e) inherit (The height/width will be inherited from its parent value).

Ex: div { height: 100px; width: 500px; background-color: blue; }

(g) CSS Margin:- It is used to define the space around elements. It is completely transparent and doesn't have any background color. It clears an area around the element.

→ Top, bottom, left and right margin can be changed independently using separate properties. You can also change all the properties at once by using shorthand margin property.

Ex.a) p { margin-top: 100px; margin-bottom: 100px; margin-right: 150px; margin-left: 80px; }

b) p { margin: 50px 100px 150px 200px; }

(It identifies that margin-top, margin-right, margin-bottom, margin-left).

c) p { margin: 50px 100px 150px; }

(It identifies that top margin, left & right margin, bottom margin).

d) p { margin: 50px 100px; }

(It identifies that top & bottom margin, left and right margin)

e) p { margin: 50px }

(It identifies that top right bottom and left margin values are 50px).

(h) CSS Opacity/Transparency:- It is used to specify the transparency of an element. Opacity is defined as degree in which light is allowed to travel through an object. In simple words, it specifies the clarity of the image.

→ Opacity setting is applied uniformly across the entire object and the opacity value is defined in term of digital value less than one. The lesser opacity value displays the greater opacity. Opacity is not inherited.

Ex: img { opacity: 0.5; filter: alpha(opacity=50); }

④ CSS Overflow:- It specifies how to handle the content when it overflows its block level containers.

→ CSS overflow property values are: 1) visible (Specifies that overflow is not clipped, it renders outside the element's box, this is a default value). 2) hidden (specifies that over flow is clipped, and rest of the content will be invisible). 3) scroll (Specifies that overflow is clipped, and a scrollbar is used to see the rest of the content). 4) auto (specifies that if overflow is clipped, a scrollbar is needed to see the rest of the content). 5) inherit (inherits the property from its parent element). 6) initial (set initial value).

Ex: div.scroll { background-color: #00ffff; width: 100px; height: 100px; overflow: scroll; }

 Scroll

⑤ CSS Padding:- It is used to define the space between the element content and the element border. It is different from CSS margin in the way that CSS margin defines the space around elements. CSS padding is affected by the background colors. It clears an area around the content.

→ CSS padding properties are: padding, padding-left, padding-right, padding-top, padding-bottom. We can specify the padding values using length (to define fixed padding pt, px, em) or in % (padding: %)

Ex: p { padding-top: 50px; padding-right: 100px;
padding-bottom: 150px; padding-left: 200px; }

(K) CSS Position:- It is used to set the position for an element, it is also used to place an element behind another and also useful for scripted animation effect.

→ You can position an element using top, bottom, left and right properties. These properties can be used only after position property is set first. A position element's computed position property is relative, absolute, fixed or sticky.

a) CSS Static Positioning:- This is a by default position for HTML elements. It always positions an element according to the normal flow of the page. It is not affected by the top, bottom, left and right properties.

Ex: div { position: static; border: 3px solid #34FF23; }

b) CSS Fixed Positioning:- It helps to put the text field on the browser. This fixed text is positioned relative to the browser window, and doesnot move even you scroll the window

Ex: <style> p.pos_fixed { position: fixed; top: 50px;
right: 5px; color: blue; } </style>

<body>
 Some text   Some text
 This is the fix positioned text
</body>

c) CSS Relative Positioning: It is used to set the element relative to its normal position. Setting the top, right, bottom and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element

4) CSS Absolute Positioning:- An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).

→ However, if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling. A "positioned" element is one whose position is anything except "static".

Ex: div.relative { position: relative; border: 3px solid #734341;
width: 300px; height: 200px; }

div.absolute { position: absolute; top: 80px; right: 0;
width: 200px; height: 200px; border: 3px; }

5) CSS Sticky:- An element with position: sticky; is positioned based on the user's scroll position. A sticky element toggles between relative and fixed, depending on the scroll position. It is positioned relative until a given offset position is met in the view port then it "sticks" in place (like position: fixed).

Ex: div.sticky { position: -webkit-sticky; /* Safari */
position: sticky; top: 0; background-color: green; }

6) CSS White Space:- This property is used to specify how to display the content within an element. It is used to handle the white spaces inside an element.

Ex. <style> P { white-space: nowrap; } </style>
<body> write some text   write some text </body>.

→ nowrap:- Sequences of whitespace will collapse into a single whitespace. In this value, text will never wrap to the next line and only break when
 tag is used.

→ normal:- default value, text is wrapped when necessary. Sequences of whitespace will be collapse into a single white space

(iii) CSS Word Wrap:- It is used to break the long words and wrap on to the next line. This property is used to prevent overflow when an unbreakable string is too long to fit in the containing box.

→ CSS word wrap values are:- 1) Normal (It is used to break words only at allowed break points). 2) break-word (used to break unbreakable words). 3) initial (Used to set this property to its default value). 4) inherit (it inherits this property from its parent element)

Ex. <html>
<style> p.test { width: 50px; background-color: green;
border: 1px solid #0000ff; padding: 10px;
word-wrap: break-word; } </style>
<body> This is a long word & it will be break
 </body> </html>

(iv) CSS outline:- CSS outline is just like CSS borders property. It facilitates you to draw an extra border around an element to get visual attention. It is as easy as to apply as a border.

Ex. <html> <style> .box { background-color: #eee;
outline: 3px solid red; border: 3px solid lightgreen;
padding: 5px 10px; } </style>
<body> Welcome </body> </html>

(v) CSS Visibility:- It is used to specify whether an element is visible or not. An invisible element also take up the space on the page. By using display property we can create invisible elements that don't take up space

Ex. <html> <style> h1.visible { visibility: visible; }
h1.hidden { visibility: hidden; } </style>
<body> <h1 class="visible"> I am visible </h1>
<h1 class="hidden"> I am hidden </h1> </body> </html>

(16) CSS Counters:- These are similar to variables. These are maintained by CSS and those values can be incremented by CSS rules to track how many times they are used. CSS counters facilitate simple CSS based incrementing and display of a number of the generated content.

⟶ Counter properties are: 1) counter-reset (used to create or reset a counter), 2) counter-increment (used to increment the counter value), 3) content (used to insert generated content)
4) counter(): Used to add value of a counter to an element.

Ex. <html><head><style> body { counter-reset: section; }

h2::before { counter-increment: section;
content: "Section" counters(section)": "; }

</style></head><body> <h1> Example of CSS counters </h1>
<h2> Java </h2> <h2> Java Script </h2> <h2> MySQL </h2>
<h2> HTML </h2> </body> </html>

(17) CSS ToolTips:- CSS tooltips are a great way to display extra information about something when the user moves the mouse cursor over an element.

Ex: <html> <style> .tooltip { position: relative;
display: inline-block; border-bottom: 1px dotted black; }
.tooltip .tooltiptext { visibility: hidden; width: 120px;
background-color: red; color: #fff;
text-align: center; border-radius: 6px; padding: 5px 0;
position: absolute; z-index: 1; }
.tooltip:hover .tooltiptext { visibility: visible; } </style>
<body style="text-align: center"> Move the mouse
over the text below
 Hover over me
 This is tooltiptext
 </body> </html>