**CSS – cascaded style sheet**

CSS stands for **Cascading style sheet**.

HTML provides structure or skeleton to the Web page.

CSS provide presentation to that structure or to the element in HTML.

**Element = <opentag>Content</closetag>**

CSS apply presentation to the above element.

Presentation means applying colors, background color, margins padding, etc. to make your web page looks beautiful or presentable.

**Types of CSS**:

* **Inline CSS** : This css is applied in a line with the element.

Property and its value are written in the opentag of HTML element with the help of style attribute. syntax- <opentag style="property:value;">content</closetag>

Ex. <h2 style="color:red;">This is first line</h2>

* **Internal CSS**: It is written in the head section of the HTML document.

syntax:

<style>

selector{

property1:value1;

property2:value2;

.

.

propertyN:valueN;

}

</style>

Ex. <style>

h2{

color:red;

background-color:yellow;

}

</style>

* **External CSS**: To apply properties to elements, we need to create separate style.css file which links to html file in head section using –

<link href="relative path of .css file " rel="stylesheet" />

**Types of selectors**:

1. **Tag selector**: ex- h2{

color:red;

}

1. **Id selector**: ex- #one{

color:red;

}

1. **Class selector**: ex- .one{

color:yellow;

}

1. **Group selector**: ex- h2,p,b{

color:blue;

}

1. **Universal selector** – ex- \*{

background-color:yellow;

font-family:arial;

}

1. **Combinator selector** – **descendent selctor** – ex. div p{color:red;}

It will apply property to all child and grandchild tags inside div

**Child selector** – ex. div>p{ color:red; }

It will apply properties to all p tags inside div tag except p tag of section tag

**Adjacent sibling** – ex. div+p{ color:red; }

It will apply property to adjacent p tag outside of div tag.

**All sibbling selector** – ex. div~p{ color:red; }

It will apply properties to all p tags outside div tag.

1. **Pseudo class selector** - syntax- selector:pseudo-class{ property:value; }

Ex- a:hover{

color:white;

}

**Hover**- mouse over link

**Link**- default

**Visited**- once visited

**Active**- clicked

1. **Attribute selector** – syntax -[attribute]{ property:value; }

Ex- [type="text"]{

background-color:yellow;

color:red;

}

All atrribute value selector- [class\*='test']{color:red;}

**Properties:**

**Text Properties:**

Color: red; | #77D1DF; |rgb(45,33,122);– color can be applied using name, color code or rgb value

text-align: justify | right |left | center

font-size:60px;

font-weight: bold;

font-style: italic | Normal

text-decoration: underline | overline | none| line-through

text-transform: lowercase | uppercase | capitalize

text-indent: 50px; ---for indent to paragraph’s first line

line-height: 25px; ---space between lines

word-spacing: 10px; --- space between words

letter-spacing: 5px; ---space between letters

text-shadow: 10px 10px 8px #ff3399; --x-offset y-offset blurrness color;

font-family: arial; ---any font

we can use google fonts -https://fonts.google.com/ select any font and click on right side - copy the third link - paste that link in head section and copy css style and paste it in external css file and use it for universal selector

**Static Layout**-

The layout in which elements width and height are mentioned in pixels is called as Static Layout. If width is given in pixels, then if we reduce the screen, then horizontal scroll bar appears, for user horizontal scrolling is tedious, it’s a bad user experience.

Ex. width:800px;

**Fluid Layout:**

The layout in which width and height are mentioned in percentage are called as Fluid layout.

Ex- width:50%;

**Disadvantage of Fluid Layout:**

If the width is given with small percentage, while decreasing the screen size, the content become so narrow, that it is difficult for the user to read the content.

**Border Properties:**

border-style: solid | inset | offset | dashed | dotted

border-color: rgb(255, 0, 145);

border-width:2px;

border-radius:10px;

border-bottom-style: solid; bottom | top | left | right

border-bottom-color: red;

border-bottom-width:4px;

border-bottom-right-radius:15px; bottom-right | bottom-left | top-right | top-left

border-bottom-left-radius:15px;

**Padding properties:**

padding:15px;

padding-top: 10px; -- padding-top | padding-right | padding-bottom |padding-left

padding: 5px 10px 5px 10px; --top right bottom left -- short hand property for padding

padding: 10px 20px; -- similar to 10px 20px 10px 20px

padding: 10px 20px 30px; -- top (left-right) bottom

**Margin Properties:**

margin is a gap between two elements. it is applied outside the box.

margin-top:100px; -- margin-left | margin-bottom | margin-right

margin:100px 300px 100px 300px; --top right bottom left

margin: 100px 400px; --similar to top right bottom left

margin: 200px auto; -- align it to center

**Box Property:**

box-sizing: border-box; --to adjust box size

**Display Property:**

Display: inline;

display: block;

display: inline-block;

disadvantage of inline - width cannot be applied with inline, so in that case use inline-block

**Background Property:**

background-color: red;

background-image: url (“relative path”)

background-repeat: no-repeat; -- repeat-x | repeat-y

background-size:100%; -- cover- but loss of image

background-attachment: fixed | scroll

**Box Shadow Property:**

box-shadow: 10px 10px 5px 10px #94b8b8; --x-offset y-offset blurrness spread color;

**Position property:**

With position property you can position element on the browser.

syntax:

selector{

position: static| fixed| relative |absolute;

top: npx;

right: npx;

bottom: npx;

left: npx;

}

**Float Property:**

float:right; -- left| right

**Whatsapp and tel symbol on webpage**:

Ex- <a href="https://api.whatsapp.com/send?phone=917774955365&text=hiiii"><img src="asset/photos/whatsapp.png" id="whatsapp" width="50px" height="50px" alt="" /></a>

<a href="tel:+918087440731"><img src="asset/photos/tel1.png" id="tel" width="50px" height="50px" alt="" /></a>

**Gradient Property:**

property is actually a **function** in which we provide values- direction, color1, color2, ....

**default direction** is top to bottom

background-image: linear-gradient (direction, col1, col2, ...);

**ex.** background-image: linear-gradient (red, yellow);

background-image: linear-gradient (to left, red, green, yellow);

background-image: linear-gradient (to top left, red, yellow);

**linear direction:**

-----------------

final direction--

to right: from left to right

to left: from right to left

to top: from bottom to top

to bottom: from bottom to top

**diagonal direction:**

-------------------

to bottom right: from top left to bottom right

to top right: from bottom left to top right

to bottom left: from top right to bottom left

to top left: from bottom right to top left

refer mdn documentation on google

radial gradient mdn documentation

https://developer.mozilla.org/en-US/docs/Web/CSS/gradient/radial-gradient

**radial-gradient**:

**Transform property** -means change

**Translate** -------

1) transform: translate (Xpx, Ypx);

2) transform: translateX (Xpx); [positive and negative values]

3) transform: translateY (Ypx); [positive and negative values]

This property helps to move element from one position to another postion making that element dynamic.

Ex. transform: translateX(-300px);

transform: translateY(300px);

transform: translate(300px, -300px)

**Scale ():** --------

this is used to scale the element [zoom in and zoom out]

1) transform: scaleX()

2) transform: scaleY()

3) transform: scale(n)

Ex. transform:scaleY(1.5); scaleX(2) \ scale(1.5)

**Rotate ():** -----------

transform: rotate(ndeg); positive for clockwise and negative for anticlockwise rotation.

transform: rotateX(ndeg);

transform: rotateY(ndeg);

Ex. transform: rotate(-45deg);

transform:rotateY(180deg);

transform:rotateX(180deg);

**transition**

--------------

with this transition property, transformation can be controlled

transition-property:property name;---------used with translate, scale and rotate----

transition-duration:ns;----this is the time in which the transformation need to be completed.

transition-delay: ns;

**ex.**

.trans{

width:300px;

height:300px;

border-style:solid;

background-color:red;

color:white;

text-align:center;

margin: 200px auto;

transition-property:transform;

transition-duration:2s;

transition-delay: 1s;

}

.trans:hover{

transform:translateX(-300px); ---------it will move that box after delay of 1s for 2s duration---

transform:scale(1.5); --------- it will scale that box after delay of 1s for 2s duration---

transform:rotateY(180deg); --------- it will rotate that box after delay of 1s for 2s duration---

}

**Types of Website or webpage Layout**

There are four types of Layouts :

1) static.

2) Fluid.

3) Adaptive.

4) Responsive.

**1)static layout:**

In this layout all width and height of the HTML elements are mentioned in pixels.

**Disadvantage:**

When we decrease the screen size,it generate horizontal scroll bar

and user need to scroll horizontally which gives bad

user experience.

**2)Fluid Layout:**

In this layout, dimensions are mentioned in percentage.

This layout removes disadvantage of Static layout. When

we reduce screen size the element decreases in the proportion and

content can be view in a single site by the user.

**disadvantage:**

If the width mentioned in the percentage is too small, on

decrease in the size of the screen content become narrow and

it is difficult for the user to read the content.

**3)Adaptive layout**

There is different layout for different breakpoint.

breakpoint: It is a width size in pixel at which the layout changes from one form to another form.

e.g breakpoint:768px

If screen width>768 px => Horizontal card layout

if screen width<768 px => Vertical card layout

**4)Responsive layout**

Responsive is a types of Webpage layout in which there is

a combination of Fluid and Adpative layout.

Responsive = Fluid + adaptive

**Adaptive Layout**

==============

In adaptive layout there is different layout for different screen size.

This can be achieved with the help of media query which apply code written into it when screen size falls within the range defined in the media query.

@media only screen and (max-width:768px){

css classes for different width.

Ths css code written inside this media query will be applicable when screen size is 0-768px and when it goes above 768 px this code is not applicable to the HTML elements.

}