There are 3 types of css.

1. **Inline css..**

e.g <body style=”background-color:red;”>

**properties..**

**background-color**: <body style=”background-color: #ECF2FF;”>

1. **Internal css..**

Internal css can be written in <head> using <style>

Ex

<head>

<title>title here</title>

<style>

b

{

color:red;

}

</style>

</head>

1. **External css.**

Here you have to link .css file using <link>. You have to declare it in <head>

<link rel="stylesheet" href="css/style.css">

**NOTE:** If we use all 3 css types at same time, then 1st priority will apply to inline css then internal css and at the end external..

**CSS Syntax**

selector{ property:value; }

**where selector is tag..**

**EX**

h1{color:red;}

Always use css properties in alphabetical order of particular element.

**Comments in css..**

**/\*……..\*/**

**Selectros in css..**

There are 3 types of selectors. By using tag name,class,id

You can target particular tag by using tag name, while class can be targeted using .classname while id can be targeted using #id

If you want to apply same style to different or multiple same tags use class. And if you want to apply unique style to tag then use id.

You can add multiple class to particular one html tag.

Below image has 2 classes. One is bro and 2nd is circular. But it cannot be done same with id

  <img class="bro circular" src="https://em-content.zobj.net/thumbs/120/apple/325/broccoli\_1f966.png" alt="broccoli-img">

**Pseudo selector**: can be identified using : before property..

p:hover

{

    color: chartreuse;

}

**Group Selector:** if you want to same property and same value to different elements then you can give it by using **,**

Ex:

h1,h2,div{

color:red;

}

We can use class with element also.. following will apply to only h1 tag which has class red. Evenif there are too many h1 tag.

Ex:

h1.red{

color:white;

}

**Universal Selector:** \* can be used to define universal selector. Following will apply to all elements.

**\*{**

margin:0px;

padding:0px;

**}**

**FavIcon:** to put icon with website title..

**To create icon -> favicon.cc**

**To add icon in title ->**

**Use <link> with rel and href attribute.**

<link rel="shortcut icon" href="favicon.ico" type="image/x-icon">

**How to remove default space from all 4 side ?**

body

{

    /\* this will remove default space from top-bottom right-left \*/

    margin: 0px;

}

**And then the tag which you used set its top margin property to 0**

Margin-top:0px;

**Padding**: it gives space b/w content and all four edges of an element.

**Margin:** gives space between two elements

**Properties of css**

1. **background-color:** to change color of background..

values should be in color name or hexa values**..**

**website: colorhunt**

1. **border-style:** to change the style of border.

values should be recatngel,none,dotted,inset

you can use border-top property

when one value specified it applied to all four side, top bottom left and right. If two specified first one is for top-bottom and second one for right-left. If three specified the first one is for top ,second for left-right and third for bottom and if four applies then value is applied top right bottom and left means it works clock-wise..

1. **height:** to change height of element. Values should be in pixels,em,% How total height of content is count? Suppose you set height 200px. You also give 10px padding, so in this case top&bottom padding will be 20px and border you set is 10 px, so in this case top-bottom border will be total 20px. Now you set margin 5px. So margin top-bottom will be 10px. Now total height of content will be acutal height(200px)+padding(20px top+bottom)+border(20px top+bottom)+margin(10px top+bottom). So total height of content will be 250px
2. **width:** to change width of element. Values should be in pixels,em,%. Here total width will be same as above rule which we define in height, but only diffrence will be that, here left-right padding,border,margin will be counted. **Note:** when you set height and width using css, you just set height and width of content area. To calculate full area of an element need to add margin,padding,border
3. **border-width:** sets width of border..
4. **border-color:** sets color of border..
5. **color:** change color of text..

RGB,HFX CODE,HSLA(Hue,Saturation,Lightness,Alpha),RGBA(Red,Green,Blue,Alpha) whereas A defines opacity

1. **font-size:** increase-decrease size of font. Values in px,em and %. Also there are predefine values such as xx-small,x-samll,smaller,small,medium,and same for large which is similar values to small. Suppose if you define 18px font size in <body> and then you define 50% font-size in particular tag then size of font will be 9px..
2. **font-family**: e.g Times New Roman,verdana,arial..

if there font-family have multiple values then it will apply priority vise..

1. **.border-radius**: to make border in shape. Values in percentage or px. Sub properties are border-top-left-radius, border-top-right-radius, border-bottom-right-radius, border-bottom-left-radius

11**. display**: can be used to set position of an element..

values-block,inline,inline-block,none

Setting value inline of **inline** element that convert block element into inline element. But once it done you cant change height-width of that element evenif it was block element before change into inline, also top-bottom margin and padding will not apply. But left-right padding & margin will apply.

Setting display property’s value as **block**, you can convert inline element into block level element. Here you can change height-width of element.

To solution both of values use **inline-block** where you can change height-width of element.

To remove element set display property’s value as **none..** space will not reserved

If you use set **visiblity** property’s value as **hidden** it also remove element but it will keep space of that removed element in web page..space will reserved

12.**position:**sets position of an element..

Values: static,relative,absolute,fixed

static- default value.top/botttom/left/right/z-index will not work.

relative- when you give this position to an element and give margin then this element will move from its current position and also put gap from its current position to previous position. top/botttom/left/right/z-index will work.

absolute: it is reverse of relative. Once absolute property used on element, the next element will take it’s place.

If we apply right then it will moved from right side of browser’s edge. And if element is in another element then it will moved based on it’s parent element. When you want to move child element in parent, set parent property relative and set child property absolute. You can use left/right/top/bottom/z-index

Fixed: if we set value fixed of an element then it will fixed evenif we scroll webpage..

13. **line-height**: gives spaces between two lines..

14. **font-weight:** values: normal,bold-bolder,light-lighter,700px=bold,900=bolder

15. **font-style:** values: italic,oblique

16. **font-variant:** values: small caps. If all letters are in capital and you use small caps 1st letter’s size will increase and remaining will display small size. 1st letter must be capital and also after space.

17. **letter-spacing:** gives space b/w 2 letters.

18. **word-spacing:** gives space b/w 2 words.

19. **text-align**: values left/start, right/end, center, justify

20. **text-decoration-line:** values: underline, line-through, overline

21. **text-decoration-color:** gives color to text-decoration-line property’s value..

22. **text-decoration-style**: values: dotted,dashed,double,wavy

23. **text-decoration-thickness:** to increase or decrease text-decoration-line or text-decoration-style

values in px

24. **text-indent:** values in px. Gives margin to paragraph from left side..

25. **text-align-last:** values:center,justify,auto,left,right

26. **text-transform:** converts text into lowercase if uppercase, and converts into uppercase if lowercase. And capitalize will make each word’s 1st letter capital.

27. **text-shadow:** to give shadow from different angles.

It takes 2 values mandatory. 1st for left-right. If we give values in negative then it will work for left side and positive for right side. 2nd for top-bottom. If we give values in negative then it will work for top side and positive for bottom side. 3rd how much you want to blur which is optional and 4th is for color which is optional..

28. **outline:** to give line over border. Takes 3 values. 1st for width 2nd for style and 3rd for color..

29. **outline-offset:** gives space b/w border and outline.

30. **min-height:** you can set min-height of content, whenever content increase it automatically increase. Content will never overflow..

31. **max-height:** you can set max-height of content, once content will reach at defiend height, content will overflow..

32. **min-width:** same as min-height. Works in width

33. **max-width:** same as max-height.

34. **box-sizing:** suppose div has width of 200px and we also give 10px margin from left and right. So width of div became 220 px, we also give padding from left and right 10px respectively, so width of div will became 240px. To keep width of div 200px, we can use **box-sizing.** It has **value border-box.** When there is a child in parent element, only padding and border consider when box-sizing is used. Margin will not be counted..

**Default value** will be **content-box**

35. **background-image:** to set image image in background.

Ex:

body{

background-image:url(“abc.jpg”)

}

36. **background-repeat**: once you set image using background-image property, you can use background-repeat property.

It has following values..

none: display once

repeat-x: display images horizontally..

repeat-y: display images vertically..

space: gives space between multiple images..

37. **background-size:** sets how image will be displayed in background.

**Values:**

cover->display in full size

contain: image will be displayed in full quality. There is possiblity that image may not be full displayed in its parent element.

auto: display image in its orignal size..

**38. opacity/transperancy:**specify opacity of an element. It takes values from 0.0 to 1.0. if value is more lower than transperancy is more. Can be used with background-color,background-image property.

**39. background-position:** sets the position of background-image. Values: left ,top,center,right,bottom, can be multiple values also.

**40. background-attachment:**values-fixed,scroll

**41. background-origin:**

values: padding-box(default),border-box,content-box

**42. border-width:** values should be in px,cm,pt,em. Also pre-defined values like thin,medium,thick

**43.** **how to use google font?**

**Step1.** Go to fonts.google.com

**Step2.** Select any font

**Step3.** Select sub size of particular font

**Step4.** Go to section of <link> which is right side in google font

**Step5.** Copy all part of that and paste into <head> after <title>

**Step6.** Copy all part of CSS rules to specify families. And paste it wherever you want to apply it in particular element.

**What is web-safe fonts?**

These fonts are universally installed across browsers..

**Generic-family:** it is type of font. In which we can say sarif and sans-sarif.

**44.** **direction:** can be used to set direction of text. Default value will be ltr. Another value will be rtl.

**45.** **unicode-bidi:** see example D:\red & white\css\css properties\direction.html

**46.** **vertical-align:** sets verical alignment of element.

47. **z-index:** defines stack of element. Highest value will display 1st when there are too many elements are above eachother..

48. **float:** to set block element in one line, without using display property. It has 2 values right and left.

49. **clear:** can only used when we use float property..it has 3 values right,left and both. When we apply right in particular child, no element can be placed on right side. Same as for left and if we use both then both side also.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FLEX-BOX\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Can be used to better way align and distribute space among elements. You have to give display flex in parent.. it is used to create one dimensional layouts. That means at a time you can set child elements horizontally or vertically. In flex-box rows also known as main-axis while columns are known as cross-axis. Child-Elements are known as flex-items and parent element is known as flex-container.

**50. flex-direction:**always define in parent element

you can set child elements in row,row-reverse,column,column-reverse

**51. flex-wrap:** used in parent element. sometimes parent element’s height or width are limited and we define more child elements, in this case child elements are overflow. For that solution we use flex-wrap.

**Values:** no-wrap(default),wrap,wrap-reverse

**52.** **flex-flow:**it is shorthand of flex-direction and flex-wrap. Takes 1st value flex-direcction and 2nd is flex-wrap

**53. justify-content:** used in parent element. to set child elements horizontally. Values: flex-start, flex-end, center, space-around(1st and last will get same space from edges while remaining got space double of it), space-evenly(everyone got same space) ,space-between(1st and last will go to both edges and remaining will get same space)

**54.** **align-items:** used in parent element. To set child elements vertically. Can be used for 1st row only. Values: flex-start, flex-end, stretch, center, baseline

Stretch(default value)

55. **align-content:** used in parent element. If you use flex-wrap and content goes in 2nd row then you can use this property to set elements. For vertical alignment

Values: flex-start,flex-end,center,stretch,space-around,space-between,space-evenly

56. **align-self:** used in child element. Used vertically. We can set position of individual child element.

Values: center,flex-end,flex-start,stretch

57. **order:** used in child element. You can set position of child element whichever you wants.

Takes values in numbers. Which may be either positive or negative. Lowest value element will start from left and highest value element will take place on right side.

**58.flex-grow:**used in child or parent element. Takes numerical value. If we set in parent by giving value as 1 then all child

element width will be same. If we give 2 in particular child element then width of that child element will be 2x then others.

**59. flex-basis:** used in child element. It works same as max-width. In responsive website, when resolution decreased width also decreased when resolution decreased from specified flex-basis.

**60.** **flex-shrink:** to reduce element size..

**61. margin-auto:**defines in child elements

We can also use properties like margin-top,margin-bottom,margin-left and margin-right. We can define values in pixels also.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CSS GRID\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Grid can be intialize using display:grid in parent element.

**61. grid-row-gap:** defines in parent element. can be used to give space b/w multiple rows. Values in numbers

**62.** **grid-template-column:** defines in parent element. Can be used to create columns or define width. You can give values in pixels. If you give 2 values then it will create 2 columns in single row.. if you give value auto then it will take remaining space in single row for creating column.

**63. grid-column-gap:** defines in parent element. Can be used to give space b/w multiple cols. Values in numbers

**64.** **grid-gap:** shorthand property of grid-row-gap and grid-column-gap. 1st value will take for grid-row-gap and 2nd for grid-column-gap. If we give only one value then it will appply for row and column.

**65.** **grid-template-row:** can be define in parent. Can be used to create row or define height. Same as grid-template-column.

**66.justify-content:**define in parent. Can be used to set child element horizontally.

**67. align-content:**defines in parent element. Can be used to define child element in verical position.

**68.** **grid-column:** defines in child element. Works same as colspan

**69.** **grid-row:** defines in child element. Works same as rowspan

**73. overflow:** to stop content out of element.

Values: visible(default),hidden,auto,overflow-x(horizontal),overflow-y(vertical)

**74.box-shadow:**to give shadow to an element. We can give multiple shadows to a single element.

it takes 5 values.

1st h-offset 2nd for v-offset 3rd for blur 4th for spread 5th for color

In 1st value if you give positive value then it will work left to right and give minus then it will work right to left. 2nd value work for top to bottom if value in positive and if in negative then will work for bottom to top.

If you want to give shadow into inside element then give **inset** value..

**75. white-space:** can be used to remove space.

**Values:**

wrap:default value

nowrap: content will display in single line. Content will go out side of element if element size is lesser than content. If you don’t want content to display outside element then use overflow:hidden

pre: this will work same as <pre>. If there is space, line break then ouput will also be same.

Pre-line: it works same as pre but in pre-line it takes only space evevnif there are too many space given between words.

pre-wrap: same as pre but nowrap will not work

**76. writing mode:** to change direction of text.

**values:**vertical-rl,verical-lr

**77.column-count:** you can divide text. Takes values in numbers.

**78.column-gap:** can be used to gap b/w column. Takes values in pixels.

**79. column-rule:** can be used to create line b/w column

**80.column-rule-width:**can be used to set thickness of column. Will put line b/w column.

**81. column-rule-style:**sets line style b/w column. It may be solid,dashed e.t.c

**82. column-rule-color:**sets color of line**.**

**83. gradient:** to give multiple colors in background. It has 4 values

**Linear gradient:**works top-bottom(default),bottom-top,right-left,left-right

**Radial gradient:** works on center(default),right bottom,left bottom,left top

**84. background-blend-mode:** when you give image a linear gradient or radial gradient. And you want image a different shades or effects then you can use this property.

**85. media queries:** can be used to create responsive webpage.

**Syntax:**

@media screen and (max-width:900px){

}

Where 900px is a breakpoint

0-480 smaller smartphones

481-768 tablets(horizontal mode) and larger smartphones

769-1279 laptops,larger tablets(landscape mode), small deskstops

1280+ larger desktops and monitors

**86. table properties:**

**border-spacing:** GIVES SPACE B/W td

**87.** resize: can be used to resize an element. But you must have to use overflow:auto with it. Otherwise will not work.

Values: horizontally,verically,both(hor&ver)

88. cursor: change design of cursor.

89. units:

**Units**

Size of an element can be defined using px,rem,vh,vw,percentage e.t.c

Following are Relative Lengths..

1. Em: if you specify font-size 3em in child element and parent has already font-size 10px. Then font size of child element will be 30px. 1em = 16px
2. rem: it specify size of an element according <html>. By default it has 16px and if you specify 3rem then size of define element will be 48px. you can change default size of an <html> by defining size in <html>
3. vw: viewportwidth. Displays data according to viewport width.
4. vh:viewportheight.
5. percentage:you can specify in percentage also..

**90. var():** it is variable. Sometimes you have to use same property in many elements and you have to define particulary in each elements. For the soltuion of that we can use var(). We can create var in :root{}. You can create variabe using --. It is global scope. And if you create in any particular eleemnt and define there then it is called local scope.

**syntax:**

**:root{**

**--main-color:red;**

**}**

**whereas main-color is name of var.**

**how to use declared var in element/class/id**

**.a{**

**color:var(--main-color);**

**}**

**so text color will be red in class a**

**91. clip-path:** can be used to create a shape of an image.

Values: circle,ellipse,inset,polygon

Circle: takes 3 values. 1st for radius 2nd for x-axis and 3rd for y-axis

Elipse: takes 4 values. 1st for horizontal 2nd for verical 3rd x-axis and 4th for y-axis

Inset: makes rectangle part of an image. Takes 4 values. 1st top 2nd right 3rd bottom and 4th left

Polygon: you can make whatever shape you want of an image. Each point takes 2 values. 1st for x-axis and 2nd for y-axis

**92. shape-outside:** you can give shape current element, as per its before element. Clip-path property must be applied to it.

**93. shape-margin:** gives space b/w 2 shaped elements.

**94. filter:** can be used to give image different shades.

\*\*\*\*\*\*Transitions\*\*\*\*\*\*\*

Can be used to give elements smoothness effects

95. **transition-property():** you have to specify value as property name. can be defined multiple values also. **define in parent.**

**96. transition-duration():define in parent.** can be used to set time for transition effect. Can specify different times for different transition-property()

**97. transition-timing-function: define in parent.**

Values: linear(same speed for start to end), ease-in(slow start then increase untill complete), ease-out(quickly start slow end), ease(slow start fast in middle then end slowely), ease-in-out(slow start and slow end), steps() you have to define values in numbers for steps() to perform operation, cubic-bezier() takes 4 values and that is in second.

**98. transition-delay()**: defines when you wants to start transition.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*transform 2d\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**99. transform:rotate():** can be used to rotate the element. takes values in degree. Degree values either in positive or negative.

**100. transform:skew():** takes values in deg. 1st value for horizntal and 2nd for vertical. You can use either skewX() for horizontal or skewY() for verical. Which takes only one value

**101. transform-origin:** you can set position from where transform will happen. 1st value for x-axis and 2nd for y-axis. Transform property must be used with it.

**102.** **transform:translate():** can be used to move element from its current position. Values in px. 1st for horizontal and 2nd for vertical

you can use either translateX() or translateY(). Which takes only one value.

103. **tansform:scale():** will increase element’s height and width from current. You have to give values in numbers.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*transform 3D\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**103. Perspective()**: must be define in transform property to see 3d effect. Values in px. less value, size will bigger

**104**. **perspective-origin():** can be used to define view of element.

**105. transform:rotateX():** can be used to rotate element. Perspective() can be define befrore rotateX()

**Same for rotateY()**

**106. transform-style():** can be used in parent. It takes only one value which is preserve-3d

**107.** **backface-visiblity():**can be used to see back part of an element.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Animation\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Animation can be applied using @keyframes and then you have to use name of animation.

**108. animation-name:** can be used to give name of an animation.

**109.** **animation-duration:** you have to define time. How much time you continue animation.

**110. animation-iteration-count:** how many time animation will happen. If you give 2 then it will perform twice.

**111.** **animation-direction:** can be used to set direction of an animation. Values: alternate(fromt-to to-from),alternate-reverse(starting will happen from reverse and then from-to to-from) ,reverse(to-from)

**112. animation:** it is shorthand property of above 4 properties.

It takes 4 values. Sequence is as per above.

**113. animation-delay:** after which time animation will start. Takes values in time

**114. animation-timing-function:** Values: linear(same speed for start to end), ease-in(slow start then increase untill complete), ease-out(quickly start slow end), ease(slow start fast in middle then end slowely), ease-in-out(slow start and slow end), steps() you have to define values in numbers for steps() to perform operation, cubic-bezier() takes 4 values and that is in second.

**115. animation-fill-mode:**values: both,back,forward. See example to see in details.

**116. animation-play-state:** to stop or run animation. Values: paused, running

**117. object-fit:** see example

**118.** **user-select: to provide security to an element.**

If we set **none** then we can not select element or select text from there.

If we set **text** then we can select text by double click

If we set **all** then we can select text by single click

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*border-image\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**119. border-image-source:** you can use it by using url()

**120. bordr-image-slice:** can be used to set image all around to content by giving values in numbers

**121.** **border-image-repeat:**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**122. caret-color: can be used to change color of cursor.**

**123. icon-fonts:**

Go to css folder where you can find fontawesome-free-6.4.0-web folder, then go to css folder of it. Where you can add css whatever you want.

**\*\*\*\*\*\*\*\*\*\*\*\*Advance Selectors\*\*\*\*\*\*\*\*\*\*\*\*\***

1. .box>p: means css will apply to those p who are direct child of .box
2. div+p: it will apply to only p which is after div. means once div tag is over, and if any p available then it will apply to it.
3. div~p: it will apply all p which are after div..
4. p::first-letter: it will apply to first letter of <p>
5. p::first-line: will apply to first line of <p>
6. p::selection: when we select text of <p> in webpage, effect will apply..
7. ::placeholder: will work in form. To give css to placeholder.
8. ::before:-while using this property, you must have to use content property..

Ex: p::before

means element before <p>’s content not element.

1. ::after