CSS Syllabus with cheat sheet

1) Implementation

Inline css	we use style attribute
internal css	we use style tag in head
external css	we use separate document and then link

2) Basic CSS properties

color: aliceblue;

background-color: aliceblue;

background-image: url(images/flower.jpg);

background-repeat: no-repeat;

background-size: cover;

background-position: center;

background: linear-gradient(blue, red);

font-size: large;

font-family: cursive;

font-weight: bolder;

text-align: center;

text-decoration: underline;

text-shadow: 3px 3px 3px white;

text-indent: 100px;

word-spacing: 10px;

letter-spacing: 5px;

height: 100vh;

width: 100vw;

3) Selectors

Universal (*)

Type selector

Class Selector (.className)

id selector (#idName)

4) Combinators

descendant ( )

child (>)

5) Box Model

background-color: white;

height: 300px;

width: 320px;

border-style: solid;

border-color: white;

border-width: 2px;

(border: solid white 2px;) short hand property (style,color,thickness)

border-radius: 13px;

padding: 15px; //and different types of padding direction

margin:15px; //and different types of margin direction

box-shadow: 3px 3px 30px black; // x-dist,y-dist,z-dist, shadow color

6) Pseudo Properties

1) pseudo classes

:hover

:focus

:active

:first-child

:last-child

2) pseudo elements

::before

::after

::first-letter

::first-line

::selection

::placeholder

7) Display Properties

display : inline (for elements to be displayed in same line)

block (for elements to be displayed in different line)

inline-block (for elements to be displayed in same line and to be varied in dimension and movements)

flex (apply flex for parent and make all child display in same line and to align the items horizontally by justify-content and vertically by align-items prop)

8) Flexing

display:flex (for parent and make all child display in same line)

justify-content (align the items horizontally)

align-items (align the items vertically)

9) Position Properties

position : static (we cannot move items by using top and left)

relative (we can move items by using top and left from parent container)

absolute (we can move items by using top and left from body container)

fixed (we can fix items to one position)

sticky (we can stick items to one position , it will be scrolled and it will be sticked to position when it reaches to specified top value)

10) Transform Properties

Transform : Scale(value) (to vary the dimension of the element)

ScaleX(value) (to vary the dimension of the element in X direction)

ScaleY(value) (to vary the dimension of the element in Y direction)

Rotate(value in deg) (to Rotate the element in z axis)

RotateX(value) (to Rotate the element in x axis)

RotateY(value) (to Rotate the element in y axis)

Skew(value in deg) (to skew the element in x axis)

RotateX(value) (to skew the element in x axis)

RotateY(value) (to skew the element in x axis)

translate(value in px or %) (to move the elements either in x or y direction)

translateX(value in px or %) (to move the elements in x direction)

translateY(value in px or %) (to move the elements in y direction)

11) transition properties :- to apply the changes slowly with some time

transition: value in seconds

transition-delay: value in seconds;   // to delay the changes

transition-timing-function: ;   // to apply different behaviours (speed) of changes

12) Animation Properties :-

animation-name: anime;   // to specify animation name

animation-duration: 10s;   // time of animation

animation-iteration-count: infinite;   // to apply loop

animation-delay: 2s;   // to delay the animation

animation-direction: alternate-reverse;   // to change direction

CREATION OF ANIMATION

```
@keyframes anime
{
    0% {
       types of changes from 0% of given time to
    }
    100% {
       types of changes upto 100% of given time.
    }
}
```