CSS Basic -II

CSS Gradients :-

CSS gradient let you display smooth transition between two or more specified colors,

↳ There are three type of gradients:

1] Linear gradient
2] Radial gradient
3] Conic gradient

1] Linear gradient :-

To create a linear gradient you must define at least two color stops.
You can also set a starting point and a direction (or an angle) along with the gradient effect.

Syntax background-image: linear-gradient (direction, color1, colorn);

Direction-default : (red, blue)
Specific direction : (to right, red, blue)
Using angles :
Using transparency:

2] Radial gradient :-

A radial gradient is defined by its center. To create a radial gradient you must also define at least two color stops.

Syntax background-image: radial-gradient (shape size at position, start-color, . . . , last-color);

You can change spacing of ~~it~~ colors / set predefined shapes / repeating gradient.

3] Conic Gradient

A conic gradient is a gradient with color transitions rotated around a center points. To create a conic gradient you must define at least two colors.

Syntax background-image: conic-gradient([from angle] [at position,] color[degree], color[degree],...);

CSS Shadow effects:-

with CSS you can add shadow to text & to element.

1] Text-shadow

2] Box-shadow.

1] Text-shadow:-

↳ You can add horizontal/vertical shadow: textshadow: 3px 3px;

↳ Color can be added :- textshadow: 3px 3px red;

↳ Blur can be added :- text-shadow: 3px 3px 3px red;

↳ Mutiple shadow on 1 text can be added:

text-shadow: 3 px 3px 3px red, 5px 5px 5px green;

↳ How can we add Border using shadows?

We can add border using shadows by putting horizontal shift value and vertical shift value as 1px.

2] Box-shadow:-

The CSS box-shadow property is used to apply one or more shadow to an element.

↳ Defaut color of shadow is text colour.

↳ Color of shadow can be changed: box-shadow: 10px 10px red;

↳ Blur can be added: box-shadow: 10px 10px 3px red;

↳ Spread-radius can be changed:

↳ Multiple shadows can be added using coma:

box-shadow: 10 px, 10px red , 15 px 15px green;

CSS Dimension properties :-

Width
Min-heigh
Max-heigh

Height
Min-width ↗ if content width is more then container also get stretch.
Max-width ~> if content width is more then container will remain same.

Overflow property :-

The overflow css shorthand property ~~st~~ sets the ~~the~~ desired behaviour for an element's overflow.

overflow: visible -> content of out of the box will be visible.
overflow: hidden -> out ~~to~~ of box content will be hidden
overflow: scroll -> if content is more you can scroll it
overflow: auto -> if content less -> visible
if content more -> scroll

CSS position property :-

Specifies the type of position method used for an element.

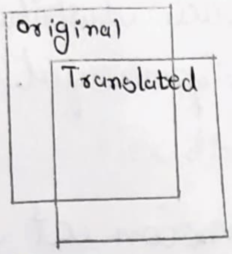
↳ Static - ~~by~~ default
↳ Relative - positioned relative to its normal position
↳ Fixed - it will stick on viewport
↳ Absolute - positioned relative to the nearest positioned ancestor
↳ Sticky - toggles between relative & fixed

CSS 2D transforms :-

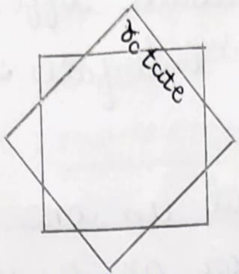
transform: translate ($\overset{x\ axis}{100}$ px, $\overset{y\ axis}{200}$ px);
transform: rotate(45 deg); + for clockwise
- for anticlockwise
transform: scale ($\overset{horizontal}{2}$, $\overset{vertical}{3}$); behaves like zooming
transform: skew(20 deg); it will bend your text
transform: matrix(1, 2, -1, 1, 80, 80); combination of all over

CSS 3D-Transforms :-

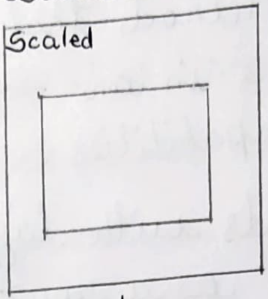
```
transfort : perspective(15 px) translatez(-10px);
transform : perspective(35 px) scalez(6);
    ↳ to view it set rotate(45 deg);
transform : rotatez(30 deg);
```



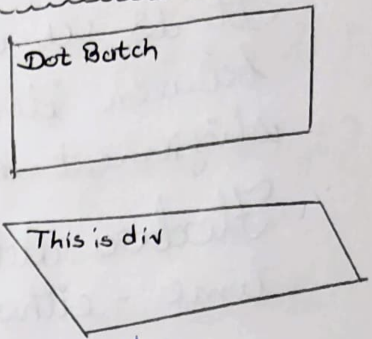
translate



rotate



scale



skew

The matrix() method :-

The matrix() method combine all the 2D transform method into one.

The parameter are as follows :

```
matrix(scalex(), skewY(), skewX(), scaleY(),
                    translateX(), translateY());

    transform : matrix(1, -0.3, 0, 1, 0, 0);
```