

v) Sticky

(18)

Mix of relative & fixed
Acts like relative until you scroll to certain point, then it becomes fixed.

Z-index property

- Controls the stacking order of elements, determining which elements appear in front of or behind others useful when elements overlap.
- Higher z-index values appear in front of lower ones
- only works on elements with a positioning context.

Overflow Property

Specifies what should happen if content overflows an element's box

This works only with block elements with specified height.

overflow: visible | hidden | clip | scroll | auto;

Opacity

Specifies the transparency/opacity of an element
the lower the value, the more transparent.

Range: 0.0 to 1.0

CSS Math Functions

(19)

Allow mathematical expression to be used as property values

calc()

max()

min()

CSS Gradients

CSS gradients let you display smooth transition b/w two or more specified colors.

* CSS defines three types of gradients:

- i, linear gradients (goes up / down / left / right / diagonally)
- ii, Radial gradients (defined by their center)
- iii, Conic gradients (rotated around a center point)

CSS Linear gradients

To create linear gradient you must define at least 2 color stops. Color stops are the colors you want to render smooth transitions among. You can also set a starting point & a direction (or an angle) along with gradient effect.

Syntax: background-image: linear-gradient (direction, color-stop1, color-stop2, ...);

Directions: ① (Top to Bottom) Default ③ Diagonal

② Left to Right

Using Angles

You can also define angles, instead of predefined direction

Syntax: background-image: linear-gradient (angle, color1, color2, ...);

Radial Gradient

Defined by its center

Also needs to define atleast 2 color stops

Syntax

background-image: radial-gradient (shape size at position, color1, color2, ...);

By default shape is ellipse.

Conic Gradient

Gradient with color transition rotated around a center point.

→ Needs atleast 2 color

→ By default, angle is 0deg & position is center

→ If no deg. specified, colors will spread equally around center point

CSS Shadow Effects

(21)

With CSS you can add shadow to text & to elements :- * text-shadow
* box-shadow

Text shadow

The CSS text-shadow property applies shadow to text.

Syntax: text-shadow: value1 value2 value3 color;

↓ ↓ ↓
horizontal vertical Blur
shift shift effect

└──────────┘ └────────┘
 ↓
 optional

 ↓
 ~~mandatory~~
 mandatory

Ex p {
 text-shadow: 2px 2px ;
 }

By default shadow color is same as text color.

Box shadow

CSS Box-shadow property is used to apply one more shadows to an element.

Syntax: box-shadow: value1 value2 value3 value4 color;

└──────────┘ ↓ ↓
horizontal vertical Blur spread
 effect radius

Q How to add border using shadows?

Ans By putting horizontal & vertical shift value as 1px.

- => Multiple shadows can be added using comma
- => Color can be changed
- => Spread radius can be changed

Dimensions in CSS

height
width
Max-height
Min height
Max-width
Min-width

2D Transform

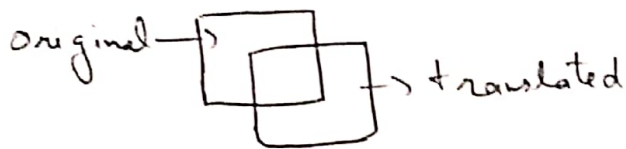
CSS transform allow you to move, rotate, scale & skew elements

With transform property you can use the following methods

translate()
rotate()
scale()
scaleX()
scaleY()
skew()
matrix()

translate() method

(23)



Moves an element from its current position according to its given parameters

Syntax:

transform: translate (value 1, value 2);

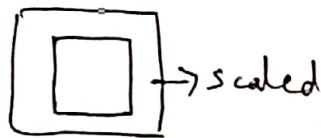
rotate() method



rotates an element clockwise or anti-clockwise

Ex div {
 transform: rotate (20 deg);
}

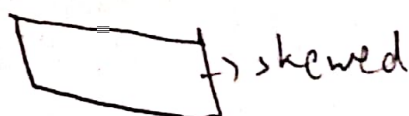
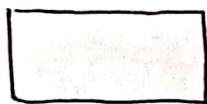
scale() method



Increases or decreases the size of an element

Ex div {
 transform: scale (2, 3);
 ↓ ↓
 x y
}

skew() method



(skew: Directed towards specific place or directed) (24)

skews an element along with X & Y axis by given angles

Ex div p

transform: skew(20deg, 10deg);

matrix() method

combines all 2D transform methods into one.

The parameters are as: matrix(scaleX(), skewX(), skewY(), scaleY(), translateX(), translateY());

3D Transform

Works on z-axis

CSS Transitions

allows you to change property values smoothly, over a given duration.

Properties:

transition

transition-delay

transition-duration

transition-property

transition-timing-function

How to use CSS Transitions?

(25)

To create transition effect, you must specify 2 things:

- > the CSS property you want to add an effect to
- > the duration of the effect.

Note: By default duration is 0

transition-timing-function

specifies the speed curve of the transition effect.

It can have values:-

ease -> specifies transition effect with slow start, then fast, then end slowly (default)

linear -> same speed from start to end.

ease-in -> slow start

ease-out -> slow end

ease-in-out -> slow start & end

cubic-bezier(n,n,n,n)

transition-delay

specifies a delay (in sec) for transition effect.

transition-duration

It specifies how many sec or milliseconds a transition takes to complete.

transition-property

(26)

It specifies the name of the CSS property the transition effect is for.

transition: shorthand property.

```
Ex    div {  
        transition: width 2s linear 1s;  
    }  
                ↑  
            timing function  
                ↓  
            property  ↓ duration  ↓ delay
```

CSS Variables

The `var()` function is used to insert the values of a CSS variable.

CSS variables have access to the DOM, which means that you can create variables with local or global scope, change the variables with JS & change the variables based on media queries.

Syntax:

The `var()` function is used to insert the values of a CSS variable. `var(--name, value)`

```
Ex:  
:root {  
    --blue: #1e90ff;  
    --white: #ffffff;  
}
```

The CSS math functions allow mathematical expressions to be used as property values.

Here we will explain the `calc()`, `max()` & `min()`

The calc() Function

The `calc()` function performs a calculation to be used as the property value

Syntax: `calc(expression)`

The max() function

The `max()` func. uses the largest value, from a comma-separated list of values, as the property value

Syntax: `max(value1, value2, ...)`

The min() function

The `min()` func. uses the smallest value, from a comma-separated list of values, as the property values.

Syntax: `min(value1, value2, ...)`

Media Queries

(28)

* Viewport

The area of the window web content can be seen.
We use the dimensions of the view-port as the basis of our media queries

Media queries are used to set different style rules for different devices or sized screens.
We use breakpoints to set the condition of a media query

The logic is @media (feature: value)

CSS Image Filter Effects

(42)

The CSS filter property is used to add visual effects to elements

Following functions used:

- * blur()
- * brightness()
- * contrast()
- * drop-shadow()
- * grayscale()
- * hue-rotate()
- * invert()
- * opacity()
- * saturate()
- * sepia()

Object fit Property

The CSS object-fit property is used to specify how an `` or `<video>` should be resized to fit its container.

Values: - * fill (Default)

* contain (resized to fit)

* cover (clipped to fit)

* none

* scale-down

- CSS allows animation of HTML elements without using Javascript or Flash!

Properties

- * @keyframes
- * animation-name
- * animation-duration
- * animation-delay
- * animation-iteration-count
- * animation-direction
- * animation-timing-function
- * animation-fill-mode
- * animation

What are CSS Animations?

An animation lets an element gradually change from one style to another.

You can change as many CSS properties you want, as many times as you want.

To use CSS animations, you must first specify some keyframes for the animation.

The @keyframe Rule

When you specify CSS styles inside the @keyframe rule, the animation will gradually change from

the current style to the new style at certain times (44)

Ex:

@keyframe example {

0% | from { background-color: red; }

100% | to { background-color: yellow; }

div {

width: 100px;

height: 100px;

background-color: red;

animation-name: example;

animation-duration: 4s;

}

NOTE: The animation-duration property defines how long an animation should take to complete. If it is not specified, no animation will occur, because default value is 0s.

animation-name

It specifies the name of the @keyframe animation.

Ex @keyframe ^{name} example {

}

Delay on Animation

The animation-delay property specifies a delay for

the start of an animation

=> Negative values are also allowed. If using negative values, the animation will start as if it had already been playing for N seconds.

Ex

```
div {
  animation-delay: 4s;
}
```

Set number of times an Animation should run.

The animation-iteration-count property specifies the number of times an animation should run.

Ex

```
div {
  animation-iteration-count: 3;
}
```

Above example will run animation 3 times.

Animation Direction

The animation-direction property specifies whether an animation should be played forwards, backwards, etc.

Values:

* normal →

* reverse ←

* alternate → ←

* alternate-reverse ← →

Ex: `div {`

`animation-direction: normal;`
`}`

Speed Curve of the Animation

The animation-timing-function property specifies the speed curve of the animation

Values:

- * ease
- * linear
- * ease-in
- * ease-out
- * ease-in-out
- * cubic-bezier(n, n, n, n)

Ex `#div {`

`animation-timing-function: linear;`
`}`

Fill mode in Animation

CSS animations do not affect an element before the 1st keyframe is played or after the last keyframe is played. The animation-fill-mode property can override this behavior.

The values of this property are:

- * none
- * backwards
- * forwards
- * both

Ex

div {

- - -

animation-fill-mode: forwards;

}

Animation Shorthand Property

The animation property is a shorthand property that is used to put all the properties into a single property.

Ex:

div {

animation: example 5s linear 2s infinite alternate;

}