CSS Transition, Animation & Transform

CSS Transform

The transform property allows elements to be rotated, scaled, skewed, and translated in both 2D and 3D space. It can take various functions as values to perform different types of transformations.

Syntax

```
selector {
    transform: function(value);
}
```

1. Translate : The translate function moves an element from its original position without affecting surrounding elements.

```
.box {
    transform: translate(50px, 100px); /* Moves element 50px right and 100px down */
}
```

2. Scale : The scale function resizes an element based on a scaling factor.

```
.box {
    transform: scale(1.5); /* Increases size by 1.5 times */
}
```

CSS Transform

3. Rotate : The rotate function rotates an element around a specified point.

```
.box {
    transform: rotate(45deg); /* Rotates the element by 45 degrees */
}
```

4. Skew : The skew function distorts an element by tilting it along the X or Y axis.

```
.box {
    transform: skew(20deg, 10deg); /* Skews the element along X and Y axes */
}
```

5. Combining Multiple Values :

```
.box {
    transform: rotate(30deg) scale(1.2) translate(50px, 20px);
}
```

CSS Transform

Transform-origin : The rotate function rotates an element around a specified point.

```
.box {
    transform-origin: top left;
    transform: rotate(45deg);
}
```

Values can be: center (default), top, right, bottom, left, or %, & px

CSS Animations

CSS animation allows elements on a web page to change their appearance over time without requiring JavaScript. This makes animations more efficient, smoother, and easier to implement.

CSS Transitions

CSS transitions allow you to change CSS properties smoothly over a given duration. It works by specifying which properties should be animated and over what duration.

Syntax

```
selector {
    transition: property duration timing-function delay;
}
```

Example

```
a {
    color: black;
    transition: color 0.5s ease-in-out;
}
a:hover {
    color: red;
}
```

CSS Animation @keyframes

For more complex animations, the @keyframes rule defines the sequence of styles at different points during the animation.

```
@keyframes moveBox {
    0% { transform: translateX(0); opacity: 1; }
    50% { transform: translateX(100px); opacity: 0.5; }
    100% { transform: translateX(0); opacity: 1; }
}
.box {
    width: 100px;
    height: 100px;
    background-color: green;
    animation: moveBox 2s ease-in-out infinite alternate; /* name duration timing-function delay  iteration-count direction*/
}
```

This animation moves the .box element 100px to the right and then back to its original position while also adjusting its opacity in a continuous loop.

CSS Animation @keyframes

Animation Properties

The animation property is a shorthand for seven individual properties:

animation-name: specifies the name of the @keyframes rule that defines the animation.

animation-duration: Makes the animation last for specified time

animation-delay: Adds a delay before the animation starts.

animation-iteration-count: Specifies how many times animation will loop eg. infinite, once, number

animation-direction: Specifies the direction of animation eg. normal/forward, reverse, alternate, alternate-reverse

animation-timing-function: Determines the acceleration pattern of the animation. ease, linear, ease-in, ease-out, ease-in-out etc.

animation-fill-mode: Specifies a style for the target element when the animation is not playing

THANK YOU

PHONE NUMBER

(+91) 778 899 2897

WEBSITE

www.indixpert.com