Module 3 – Databinding and Animations

Demo 3: CSS Transformation and Animations

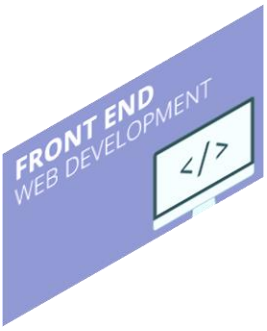
edureka!

CSS – 2D Transformation

2D transformations are a part of computer graphics. You can rotate, scale, position, shape and change the view of the element using the CSS 2D transform properties

Rotate

Scale



Ske

With 2D transformation HTML elements can be stretched, spinned, moved, scaled etc.

Some of the methods of 2D transformation are:

scale (): Increases or decreases the size of the elements

translate (): Moves the current element to another position

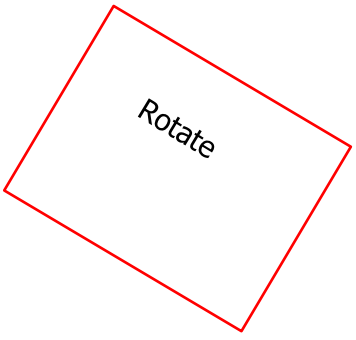
rotate (): Rotates the element by taking a degree

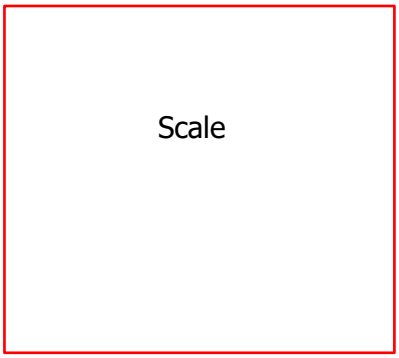
skewX (): Skews an element along the X-axis by the given angle

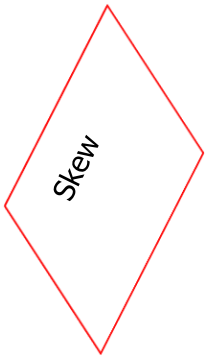
skewY (): Skews an element along the Y-axis by the given angle

skew (): Skews an element along the X and Y-axis by the given angles

matrix (): Can combine all the 2D methods







CSS – 2D Transformation – Translate

→ To display an element from one position to another, translate () method is used

→ translate (x,y) – Translates the element to the specified position. Here the element is text

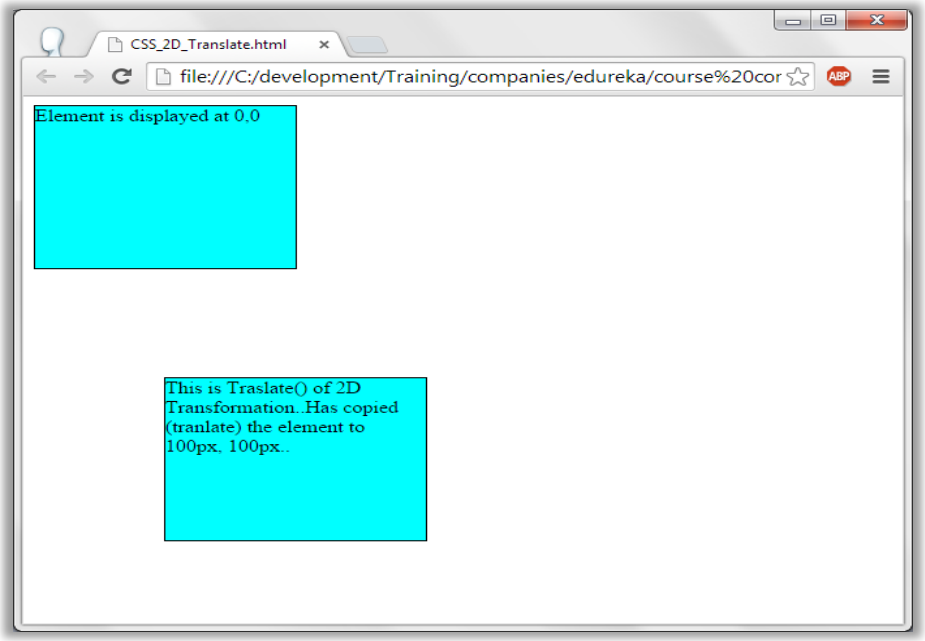
→ In our example, using the translate property of 2D transformation, the element from position (0,0) is translated to position (100,100)

Example

```
<style>
div {
    width: 200px;
    height: 150px;
    background-color: #00FFFF;
    border: 1px solid black;
}

div#div2 {
    -webkit-transform: translate(100px,100px);
}
</style>
```

At

Output



Different browsers support different transform properties. So, make sure you add transformation properties for all the browsers e.g. Chrome, IE, Mozilla, Safari, Opera. Otherwise transformations will not work across different browsers

CSS – 2D Transformation – Rotate

→ To rotate an element by a certain degree, **rotate ()** method is used

→ **rotate (xdeg)** - Rotates the element to x degree. Here the element is image

→ In our example, using the rotate property of 2D transformation, the image is rotated to 30deg

Example

```
img#id1 {
        position:absolute;
        left: 600px;
         top: 100px;
    -webkit-transform: rotate(30deg);
}
img#id2 {
         position:absolute;
        left: 300px;
         top: 100px;
    -webkit-transform: rotate(30deg);
}
```

The image is rotated by 30 degrees

Output

CSS – 2D Transformation – Scale

→ To enlarge the size of the element, scale() method is used

→ scale() - Scales the element up/down depending on the axis mentioned in the code

→ Syntax: -webkit-transform: scale(x,y);

Scales the element to the x (horizontally) and y (vertically) position appropriately

If scale (2,2) is used, the width and height is scaled twice as the original element (e.g. image)

Example

```
img#id1{
        position: absolute;
        left: 400px;
        top: 100px;
       -webkit-transform: scale(1,1);
}

img#id2 {
       position: absolute;
        left: 400px;
        top: 300px;
    -webkit-transform: scale(2,2);
}
```

Output

CSS – 2D Transformation – Skew

→ When you want the element to be displayed in an oblique angle, skew() method is used

→ skew() - Rotates the element in horizontal and vertical axis

→ Syntax: webkit-transform: skew(20deg,30deg);

This will skew the element by 20 degrees horizontally(x-axis) and 30 degrees vertically(y-axis)

Example

```
img#id1 {
         position:absolute;
         left: 650px;
          top: 200px;
       -webkit-transform: skew(20deg,30deg);
}
img#id2 {
          position:absolute;
         left: 800px;
          top: 100px;
     -webkit-transform: skew(10deg,50deg);
}
```

Output

CSS – 3D Transformation

→ 3D transforms operate horizontally(x-axis) and vertically(y-axis)

→ They make the elements look interactive so that the user is always associated with the webpage

→ It gives you a look of a virtual reality

In the above image, you can see the transformation of number 3 to number 5, using the 3D transform

Now let us learn how to write code to perform such transformations

→ With CSS elements can also be rotated in 3D direction using following methods :

→ rotateX() : rotates an element around its X-axis at a given degree

→ rotateY() : rotates an element around its Y-axis at a given degree

→ rotateZ() : rotates an element around its Z-axis at a given degree

```
div {
    -webkit-transform: rotateX(30deg); /* Safari */
    transform: rotateX(30deg);
  }
```

```
div {
    -webkit-transform: rotateY(90deg); /* Safari */
    transform: rotateY(90deg);
}
```

```
div {
    -webkit-transform: rotateZ(60deg); /* Safari */
    transform: rotateZ(60deg);
}
```

CSS – 3D Transformation – X-axis

→ Now let us see the code to create a 3D transform, horizontally

CSS – 3D Transformation – Y-axis

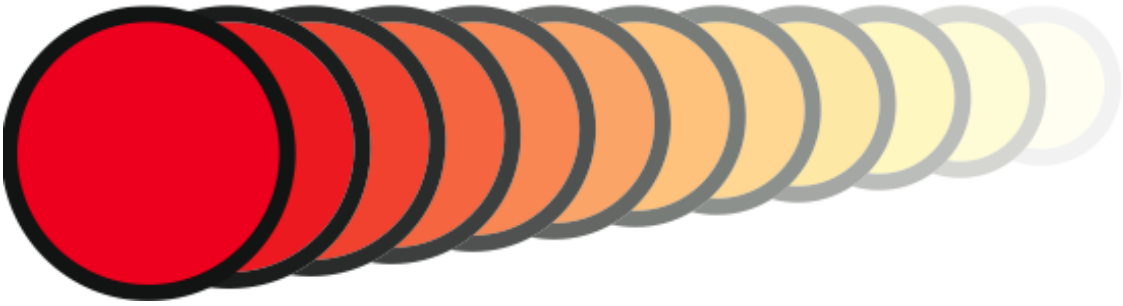
→ Now let us see the code to create a 3D transform, vertically

This type of image rotation can be seen in the e-commerce websites

CSS – Transition

→ When you want to change the state of an element, you can use the CSS3 transition property

CSS3 transitions allows you to change an element's properties smoothly over a given duration



Here you see a transition of a circle from red to yellow color

→ Transitions are the effects which changes from one state to another. To generate changes like this transition can be used. We use transitions in all the places where the object is required to change its state in a graphical way

→ It can be done with styles in CSS3 without using JavaScript or flash

→ -webkit-transition: width 2s, height 2s, -webkit-transform 2s;

In transitioning, for width it takes 2 seconds, for height it takes 2 seconds and to show the effect of transformation will take 2 seconds

→ -webkit-transform: rotate(360deg);

Rotates the elements by 360 degree

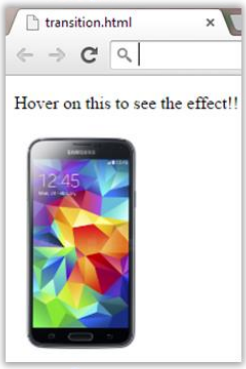
If the degree is 720 then it takes 360degress x 2. Complete rotation will take place twice

If the given degree is 3600, the element will be rotated 10 times

Example

```
<style>
div {

    width: 100px;
    height: 100px;
    background: DeepPink;
    -webkit-transition: width 2s, height 2s, -webkit-transform 5s;

}

div:hover {
    width: 400px;
    height: 400px;
    -webkit-transform: rotate(3600deg); /* Chrome, Safari, Opera */

}
</style>
```

Output



Before Hovering

After Hovering

CSS – Animation

→ Animation is a technique where movements are created using a sequence of images

» In animation, several images are taken together and are displayed one after the other

» To use CSS3 animation, you must first specify some keyframes for the animation

» Keyframes hold what styles the element will have at certain times

→ With CSS3 animation properties, it is possible to replace flash and JavaScript animations

→ **Syntax:**

-webkit-animation: name 5s; ⟶ Animation is defined under name and it displays in 5 seconds

@-webkit-keyframes name { ⟶ This is the definition of how colors have to change

0% {background: red;} ⟶ Initially red color is displayed

25% {background: green;} ⟶ After 25% of the animation, color changes to green

50% {background: blue;} ⟶ After 50% of animation, color changes to blue

100% {background: yellow;} ⟶ After 100% of animation, color changes to yellow

}

→ In animation, several images are taken together and are displayed one after the other

→ Animations are used to display the images to generate very good visual effects

→ In our example, we are 1st displaying the block in red color, then color changes to green and then to red and finally to yellow

The block here does not move. It just changes the color, and the script is executed in 5 seconds

Example

```
<style>
div {
width: 100px;
height: 100px;
background: red;
-webkit-animation: myfirst 5s;
}
@-webkit-keyframes myfirst {
0%   {background: red;}
25%  {background: green;}
50%  {background: blue;}
100% {background: yellow;}
}
</style>
```

Output

→ Here the block moves from one position to another, forming a pattern. The block is of height and width 100px

Example

```
<style>
div {
width: 100px;
height: 100px;
background: red;
position:relative;
-webkit-animation: myfirst 5s;
}
@-webkit-keyframes myfirst {
0%   {background: red;left:0px; top:0px;}
25%  {background: green;left:500px; top:0px;}
50%  {background: blue;left:500px; top:500px;}
75%  {background: brown;left:0px; top:500px;}
100% {background: yellow;left:0px; top:0px;}
}
</style>
```

Displays at 0,0

Displays at 500,0. So the box moves from 0,0 to 500,0 as animation and so on

Output