

Professional Skills
Agile Fundamentals
Jira
Git
DevOps
Cloud Fundamentals
Databases Introduction
Java Beginner
Maven
Testing (Foundation)
Java Intermediate
HTML
CSS <ul style="list-style-type: none"><li>Introduction to CSS3</li><li>Fonts</li><li>Transform</li><li>Selectors</li><li>Selectors - Pseudo-Class / Pseudo-Elements</li><li>Spatial Attributes</li><li>Alignment</li><li>Overflow</li><li>Display and Positioning</li><li>Background and Borders</li><li>Bootstrap Introduction</li><li>Bootstrap Nav</li><li>Bootstrap Collapse</li><li>Bootstrap Grid</li><li>Bootstrap Forms &amp; Inputs</li><li>Bootstrap Modal</li><li>Bootstrap Cards</li></ul>
Javascript
Spring Boot
Selenium

# Transform

## Contents

- Overview
  - What is the transform property?
    - transform-origin Property
  - 2D vs 3D Transformation
  - Rotating
  - Translate
  - Scale
  - Skew
- Tutorial
- Exercises

## Overview

In this module, we will be exploring the transform property.

### What is the transform property?

The transform property applies a 2D or 3D transformation to an element. This property allows you to rotate, scale, move, skew, etc., elements. It modifies the coordinate space of the CSS visual formatting model.

#### transform-origin Property

The transform-origin property allows you to change the position of transformed elements.

Note: This property must be used together with the transform property.

### 2D vs 3D Transformation

2D transformations can change the x- and y-axis of an element by using the following methods:

- transformMethodX() - This will transform along the X-axis
- transformMethodY() - This will transform along the Y-axis

3D transformations can also change the z-axis of an element, by using the following method:

- transformMethodZ() - This will transform along the Z-axis

### Rotating

The rotate() method rotates an element clockwise or counter-clockwise according to a given degree.

If the degree specified is positive the element will rotate clockwise; else, if negative the element will rotate counter-clockwise.

We can also rotate along a specified axis by using one of the following methods:

- rotateX()
- rotateY()
- rotateZ()

### Translate

The translate() method moves an element from its current position (according to the parameters given for the X-axis and the Y-axis)

Sonarqube
Advanced Testing (Theory)
Cucumber
MongoDB
Express
NodeJS
React
Express-Testing
Pre-Project Assessment
Markdown
IDE Cheatsheet

We can also translate along a specified axis by using one of the following methods:

- `translateX()`
- `translateY()`
- `translateZ()`

### Scale

The `scale()` method increases or decreases the size of an element (according to the parameters given for the width and height).

We can also scale along a specified axis by using one of the following methods:

- `scaleX()`
- `scaleY()`
- `scaleZ()`

### Skew

The `skew()` method skews an element along the X and Y-axis by the given angles.

If the second parameter is not specified, it has a zero value.

We can also skew along a specified axis by using one of the following methods:

- `skewX()`
- `skewY()`
- `skewZ()`

## Tutorial

In this tutorial, we are going to apply the transformations listed above to a HTML element and observe the behaviour.

1. Create the following HTML Document:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>CSS: Transform</title>
  <style>
    div {
      border: solid red;
      width: 140px;
      height: 60px;
    }
  </style>
</head>
<body>
  <div>Watch the magic happen...</div>
</body>
</html>
```

The output should look something like this:



2. Rotate the div 20 degrees clockwise - to apply a rotation to the div, we use the `translate` property along with the `rotate()` attribute and specify the degree as a parameter.

```
div {
  ...
  transform: rotate(20deg);
}
```

Change the angle to `-20deg` and observe the output.

3. Change the position of the transformed element to `left` - Set the `transform-origin` property to `left`

```
div {  
  ...  
  transform-origin: left;  
}
```

4. Now, rotate the same div 90 degrees around it's z-axis - To do this we use the `rotateZ()` method and pass in the degree as parameter.

```
div {  
  ...  
  transform: rotate(20deg) rotateZ(90deg);  
}
```

5. Move the element 200 pixels to the right and 100 pixels down from its current position - to achieve this, we use the `translate()` method.

```
div {  
  ...  
  transform: rotate(20deg) translate(200px,100px);  
}
```

6. Increase the size of the div 2 times of its original width, and three times of its original height - We need to use the `scale(w,h)` method and supply the width and height change as parameters.

```
div {  
  ...  
  transform: rotate(20deg) translate(200px,100px) scale(2,3);  
} ...
```


7. Skew the `<div>` element along the X-Axis, and 10 degrees along the Y-axis.

```
div {  
  ...  
  transform: rotate(20deg) translate(200px,100px) scale(2,3)  
  skew(20deg,10deg);  
}
```

## Exercises

1. Compare the two images below and apply the relevant transformations to achieve the output below.

Original:

Original Image

Output:

Output

▼ Solution

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Document</title>
  <style>
    #transformMe {
      transform: translate(100px, 250px) rotateY(180deg)
skew(-25deg, 20deg);
    }
  </style>
</head>
<body>
  
  
</body>
</html>
```