

# CSS “2D” Transforms

\*CSS transforms allow you to move, rotate, scale, and skew elements.

\*Mouse over the element below to see a 2D transformation:



```
...
  .filterByOrg {
    status = filterByStatus ? study.status === filterByStatus : true
    matchStatus) {
  }

  function filterStudies({ studies, filterByOrg }) {
    return studies.filter(study => {
      return study.lead_organization === filterByOrg
    })
  }
}
```

# CSS 2D Transforms Methods

With the `Css transform` property you can use the following 2D transformation methods:

- 1- `translate`
- 2- `Rotate`
- 3- `scale x()`
- 4- `scale y()`
- 5- `scale()`
- 6- `skew x()`
- 7- `skew y()`
- 8- `skew()`
- 9- `matrix()`



# The translate( ) Method

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



```
<div>
```

```
    This div element is moved 50 pixels to the  
    right, and 100 pixels down from its current position.
```

```
</div>
```

```
div{
```

```
    width: 300px;  
    height: 100px;  
    background-color: yellow;  
    border: 1px solid black;  
    transform: translate(100px,100px);
```

```
}
```

This div element is moved 50 pixels to the right, and 100 pixels down from its current position.



# The rotate( ) Method

- The rotate ( ) method rotates an element clockwise or counter-clockwise according to a given degree.
- Deg( ) , return ( ) , rad ( ) , grad ( )

```
..<div>
  ..This a normal div element.
  ..</div>
  ..
  ..<div id="myDiv">
  ..This div element is rotated clockwise 20 degrees.
  ..</div>
```

```
div {
  width: 300px;
  height: 100px;
  background-color: ■ yellow;
  border: 1px solid □ black;
}
```

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

This a normal div element.

This div element is rotated clockwise 20 degrees.

# The scale( ) Method

- The scale ( ) method increases or decreases the size of an element (according to the parameters given for the width and height)
- scale( x, y) , scalex( ) , scaley( )



```
.scale{
margin: 150px;
width: 200px;
height: 100px;
background-color: yellow;
border: 1px solid black;
transform: scale(2,3);
}
```

```
.scale-x{
margin: 150px;
width: 200px;
height: 100px;
background-color: yellow;
border: 1px solid black;
transform: scaleX(2);
}
```

```
.scale-y{
margin: 150px;
width: 200px;
height: 100px;
background-color: yellow;
border: 1px solid black;
transform: scaleY(3);
}
```

This div element is two times of its original width, and three times of its original height.

This div element is two times of its original width.

This div element is three times of its original height.

# The skew( ) Method

- The skew( ) method skews an element along the X-axis by the given angle.
- skew( x, y ) , skewx( ) , skewy( )

```
div{  
  width: 300px;  
  height: 100px;  
  background-color: ■ yellow;  
  border: 1px solid ■ black;  
  margin: 20px;  
  margin-bottom: 50px;  
}
```

```
.skew-x{  
  transform: skewX(20deg);  
}
```

```
.skew-y{  
  transform: skewY(20deg);  
}
```

```
.skew{  
  transform: skew(20deg, 10deg);  
}
```

This div element is two times of its original width.

This div element is three times of its original height.

This div element is two times of its original width, and three times of its original height.



# The matrix( ) Method

- The matrix( ) method combines all the 2D transform methods into one.
- The matrix() method takes six parameters, containing mathematic functions, which allows you to rotate, scale, move (translate), and skew elements.
- The parameters are as follow: matrix(scaleX(), skewY(), skewX(), scaleY(), translateX(), translateY())

```
div {  
  width: 300px;  
  height: 100px;  
  background-color: yellow;  
  border: 1px solid black;  
}
```

```
div.myDiv1 {  
  transform: matrix(1, -0.3, 0, 1, 0, 0);  
}
```

```
div.myDiv2 {  
  transform: matrix(1, 0, 0.5, 1, 150, 0);  
}
```

This a normal div element.

Using the `matrix()` method.

Another use of the `matrix()` method.

# CSS«3D »Transforms

*\*In this chapter you will learn about the following CSS property:*



```
...org === filterByOrg ? study.lead_organization === filterByOrg : false  
...Status = filterByStatus ? study.status === filterByStatus : true  
...atchStatus) {
```

```
function filterStudies({ studies, filterByOrg, filterByStatus }) {  
  return studies.filter(study => {  
    return study.lead_organization === filterByOrg &&  
    study.status === filterByStatus
```



# CSS 3D Transforms Methods

With the `Css transform` property you can use the following 3D transformation methods:

**1-rotate( )**

**2-rotatey( )**

**3-rotatez( )**





# The rotateX( ) Method

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

```
div {  
  width: 300px;  
  height: 100px;  
  background-color: yellow;  
  border: 1px solid black;  
}  
  
.section-1 {  
  transform: rotateX(60deg);  
}  
  
.section-2 {  
  transform: rotateX(90deg);  
}  
  
.section-3 {  
  transform: rotateX(150deg);  
}  
  
.section-4 {  
  transform: rotateX(180deg);  
}
```

This div element is rotated 150 degrees.

This div element is rotated 120 degrees.

This div element is rotated 120 degrees.

# The rotatey( ) Method

- The rotatey( ) method rotates an element around its y-axis at a given degree:



```
div {  
  width: 300px;  
  height: 100px;  
  background-color: yellow;  
  border: 1px solid black;  
}  
  
.section-1 {  
  transform: rotatey(60deg);  
}  
  
.section-2 {  
  transform: rotatey(90deg);  
}  
  
.section-3 {  
  transform: rotatey(150deg);  
}  
  
.section-4 {  
  transform: rotatey(180deg);  
}
```



This div element is rotated 60 degrees.

This div element is rotated 120 degrees.

This div element is rotated 180 degrees.

# The rotatez( ) Method

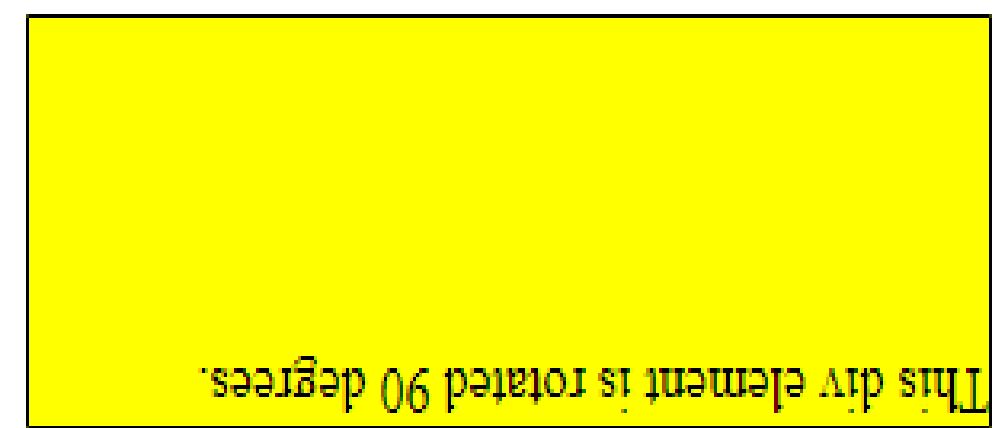
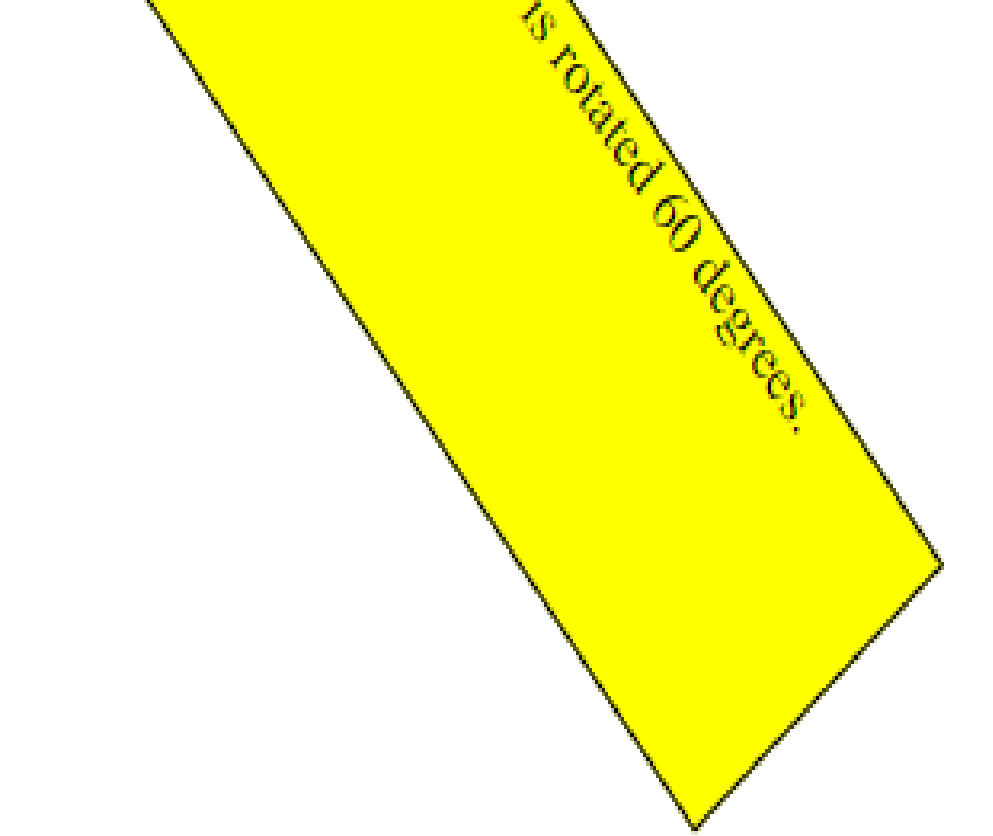
- The rotatez( ) method rotates an element around its z-axis at a given degree:



```
div {
  width: 300px;
  height: 100px;
  background-color: yellow;
  border: 1px solid black;
  margin-bottom: 200px;
}

.section-1 {
  transform: rotate(50deg);
}

.section-2 {
  transform: rotate(180deg);
}
```



# Some references

- MDN Web Docs
- W3Schools
- Css-Tricks
- Elzero-web-school
- Css community
- Git-Hub





THANKS FOR YOU

