UI BASICS

INTERACTION – TRANSFORM

CSS - TRANSFORM

CSS **transform** allows you to move, rotate, scale, and skew elements.

With the CSS **transform** property you can use the following transformation methods:

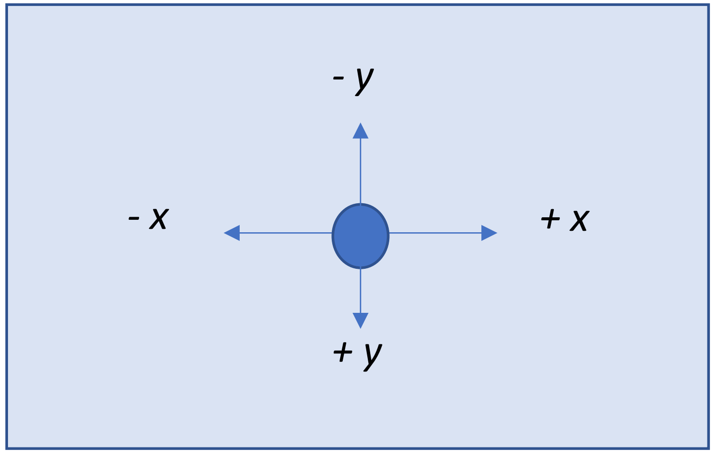
- translate()
- rotate()
- scale()
- skew()
- matrix()

CSS - TRANSFORM

translate()

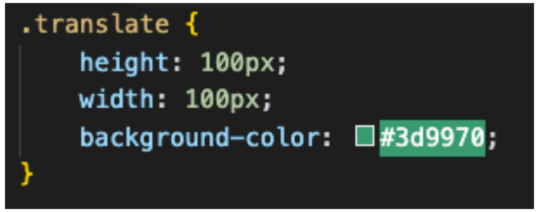
- The **translate()** method moves an element from its current position (according to the parameters given for the X-axis and the Y-axis).
- It moves an element sideways or up and down.

```
transform: translateX(x value);

transform: translateY(y value);

transform: translate(x value, y value);
```

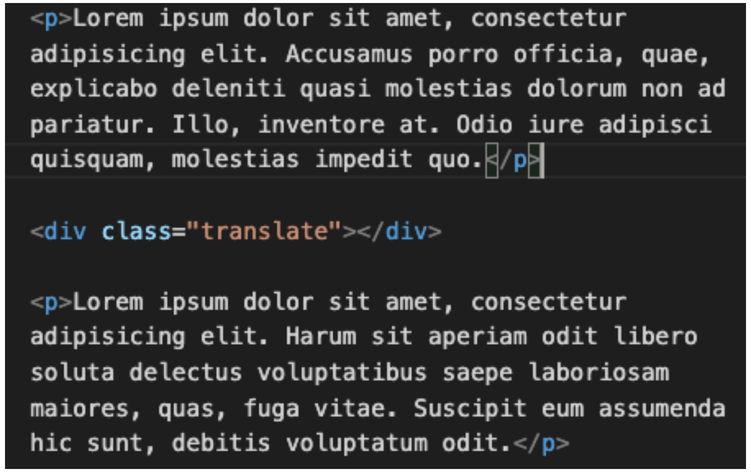


CSS - TRANSFORM

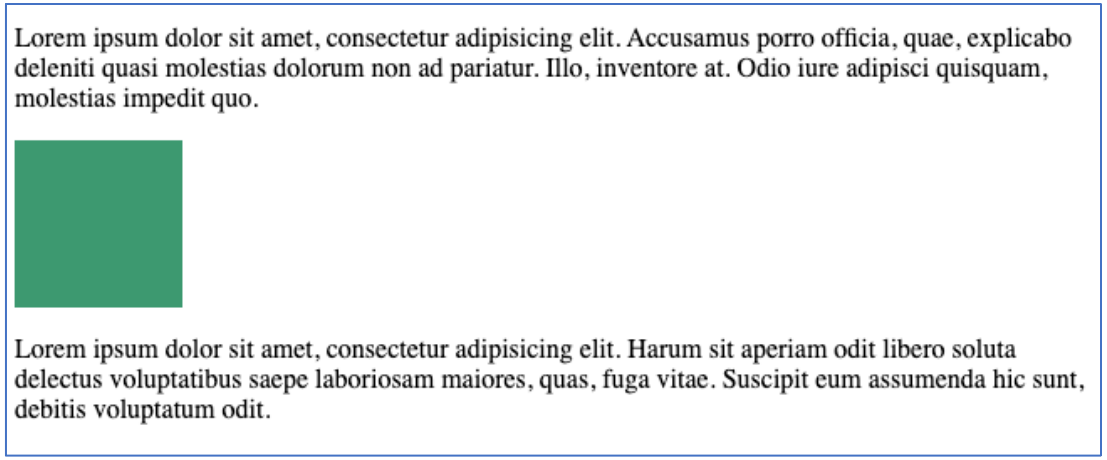
translate()



```css
.translate {
    height: 100px;
    width: 100px;
    background-color: #3d9970;
}
```

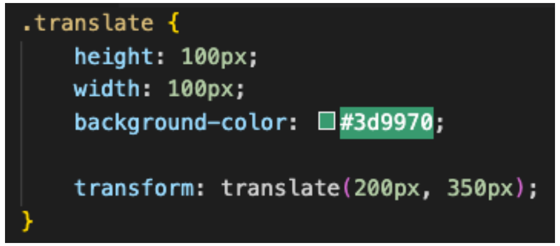


```html
<p>Lorem ipsum dolor sit amet, consectetur
adipisicing elit. Accusamus porro officia, quae,
explicabo deleniti quasi molestias dolorum non ad
pariatur. Illo, inventore at. Odio iure adipisci
quisquam, molestias impedit quo.</p>

<div class="translate"></div>

<p>Lorem ipsum dolor sit amet, consectetur
adipisicing elit. Harum sit aperiam odit libero
soluta delectus voluptatibus saepe laboriosam
maiores, quas, fuga vitae. Suscipit eum assumenda
hic sunt, debitis voluptatum odit.</p>
```

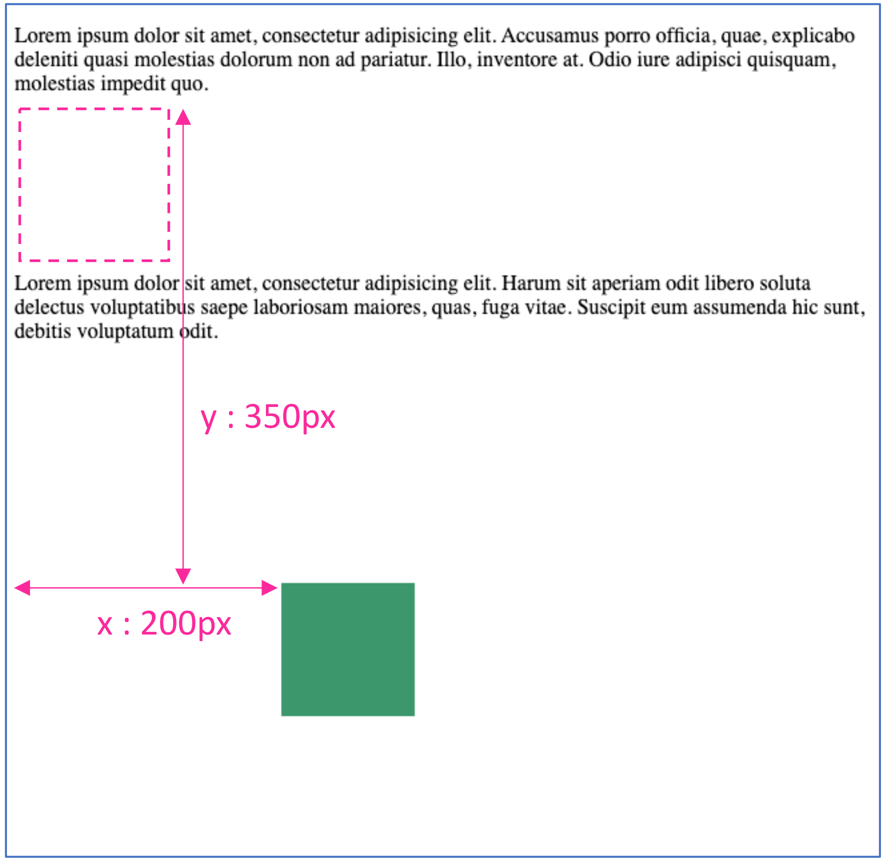


Lorem ipsum dolor sit amet, consectetur adipisicing elit. Accusamus porro officia, quae, explicabo deleniti quasi molestias dolorum non ad pariatur. Illo, inventore at. Odio iure adipisci quisquam, molestias impedit quo.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Harum sit aperiam odit libero soluta delectus voluptatibus saepe laboriosam maiores, quas, fuga vitae. Suscipit eum assumenda hic sunt, debitis voluptatum odit.



```css
.translate {
    height: 100px;
    width: 100px;
    background-color: #3d9970;

    transform: translate(200px, 350px);
}
```

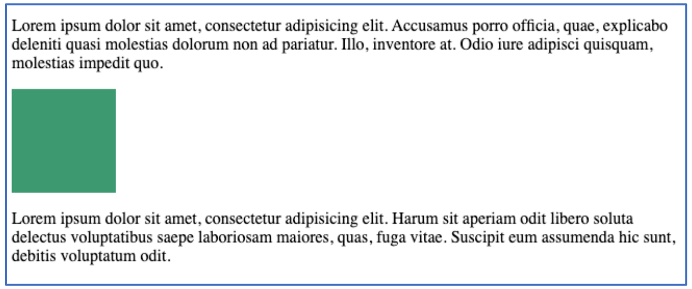


Lorem ipsum dolor sit amet, consectetur adipisicing elit. Accusamus porro officia, quae, explicabo deleniti quasi molestias dolorum non ad pariatur. Illo, inventore at. Odio iure adipisci quisquam, molestias impedit quo.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Harum sit aperiam odit libero soluta delectus voluptatibus saepe laboriosam maiores, quas, fuga vitae. Suscipit eum assumenda hic sunt, debitis voluptatum odit.

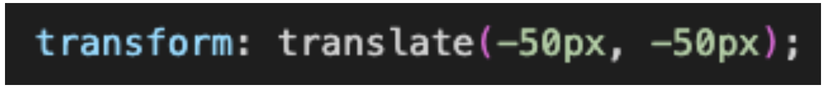
CSS - TRANSFORM

translate()

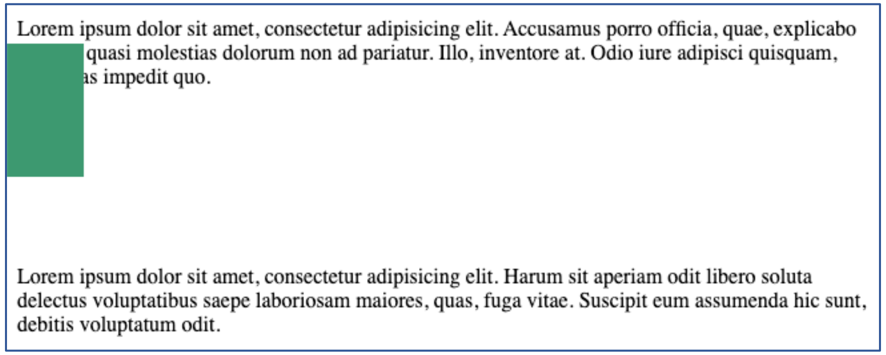


Lorem ipsum dolor sit amet, consectetur adipisicing elit. Accusamus porro officia, quae, explicabo deleniti quasi molestias dolorum non ad pariatur. Illo, inventore at. Odio iure adipisci quisquam, molestias impedit quo.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Harum sit aperiam odit libero soluta delectus voluptatibus saepe laboriosam maiores, quas, fuga vitae. Suscipit eum assumenda hic sunt, debitis voluptatum odit.

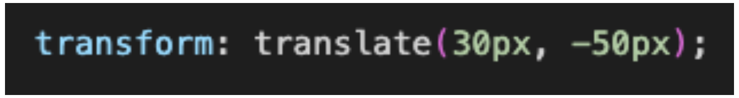


```css
transform: translate(-50px, -50px);
```

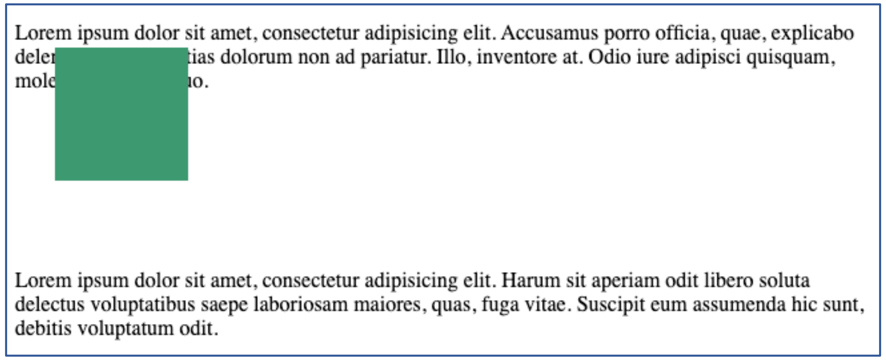


Lorem ipsum dolor sit amet, consectetur adipisicing elit. Accusamus porro officia, quae, explicabo
quasi molestias dolorum non ad pariatur. Illo, inventore at. Odio iure adipisci quisquam,
as impedit quo.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Harum sit aperiam odit libero soluta delectus voluptatibus saepe laboriosam maiores, quas, fuga vitae. Suscipit eum assumenda hic sunt, debitis voluptatum odit.



```css
transform: translate(30px, -50px);
```



Lorem ipsum dolor sit amet, consectetur adipisicing elit. Accusamus porro officia, quae, explicabo
deler tias dolorum non ad pariatur. Illo, inventore at. Odio iure adipisci quisquam,
mole uo.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Harum sit aperiam odit libero soluta delectus voluptatibus saepe laboriosam maiores, quas, fuga vitae. Suscipit eum assumenda hic sunt, debitis voluptatum odit.

CSS - TRANSFORM

rotate()

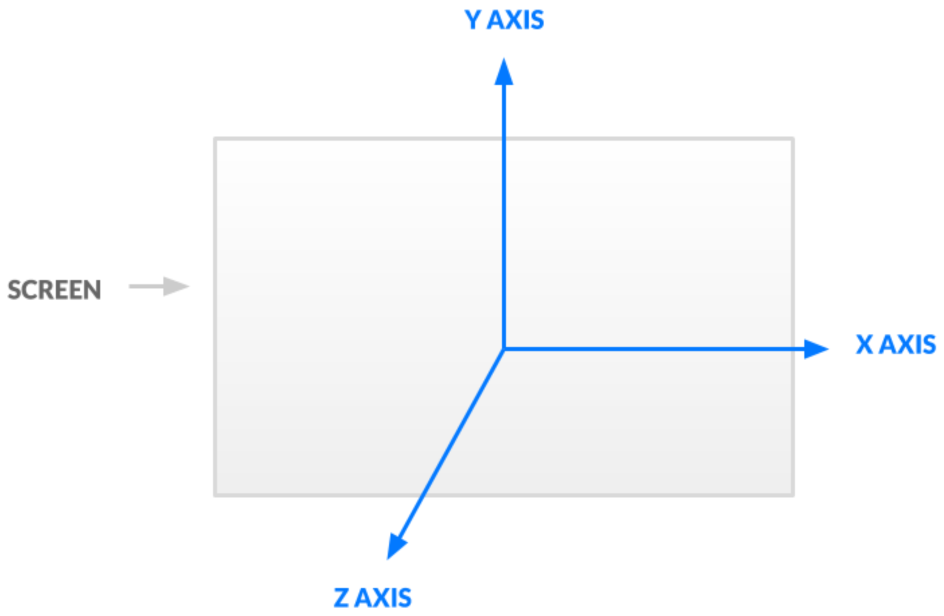
The **rotate()** method rotates an element clockwise or counter-clockwise according to a given degree.

transform: rotateX(deg);

transform: rotateY(deg);

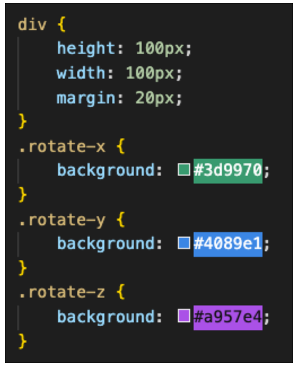
transform: rotateZ(deg);

transform: rotate(deg); → Rotates the element clockwise from its current position.

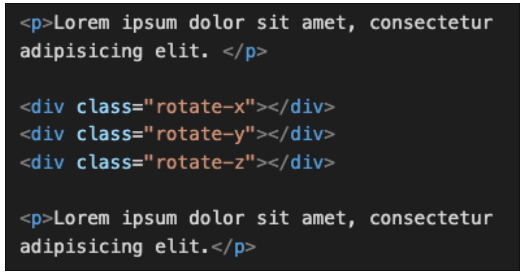


CSS - TRANSFORM

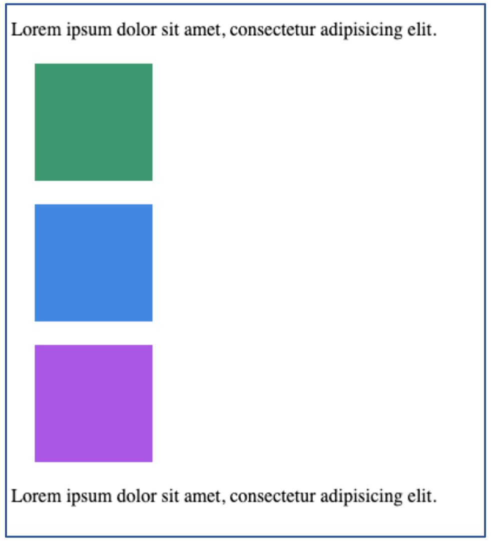
rotate()



```
div {
    height: 100px;
    width: 100px;
    margin: 20px;
}
.rotate-x {
    background: 🟩#3d9970;
}
.rotate-y {
    background: 🟦#4089e1;
}
.rotate-z {
    background: 🟪#a957e4;
}
```

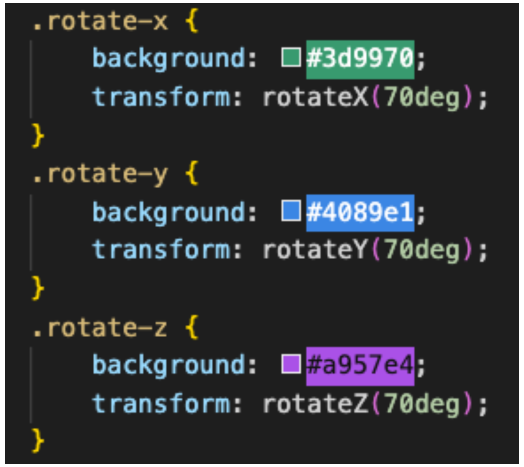


```
<p>Lorem ipsum dolor sit amet, consectetur
adipisicing elit. </p>

<div class="rotate-x"></div>
<div class="rotate-y"></div>
<div class="rotate-z"></div>

<p>Lorem ipsum dolor sit amet, consectetur
adipisicing elit.</p>
```

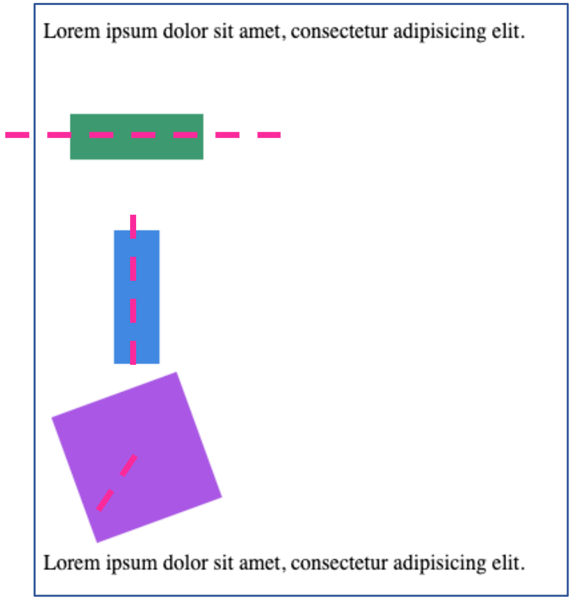


Lorem ipsum dolor sit amet, consectetur adipisicing elit.

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

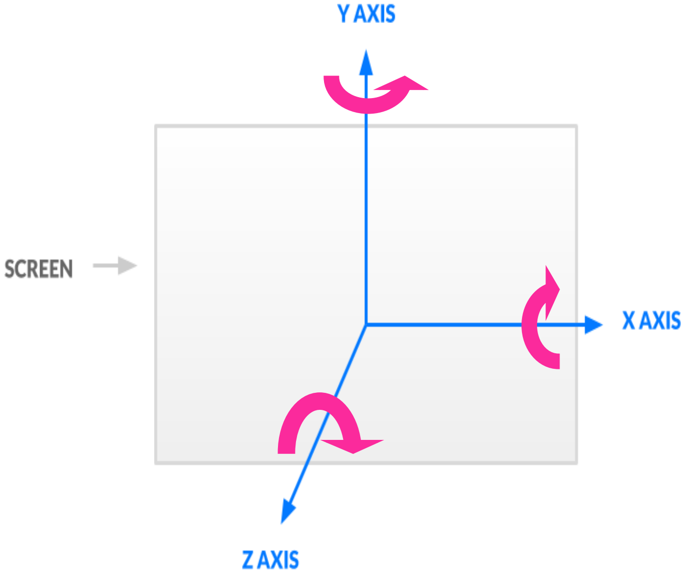


```
.rotate-x {
    background: 🟩#3d9970;
    transform: rotateX(70deg);
}
.rotate-y {
    background: 🟦#4089e1;
    transform: rotateY(70deg);
}
.rotate-z {
    background: 🟪#a957e4;
    transform: rotateZ(70deg);
}
```



Lorem ipsum dolor sit amet, consectetur adipisicing elit.

Lorem ipsum dolor sit amet, consectetur adipisicing elit.



CSS - TRANSFORM

scale()

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

```
transform: scaleX( width);
```

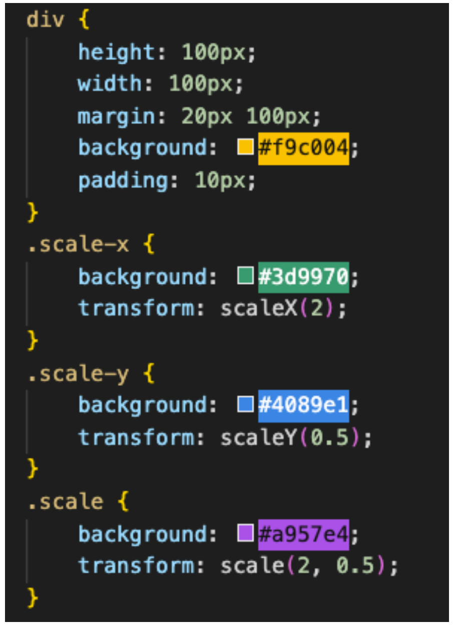
```
transform: scaleY( height);
```

```
transform: scale( width, height);
```

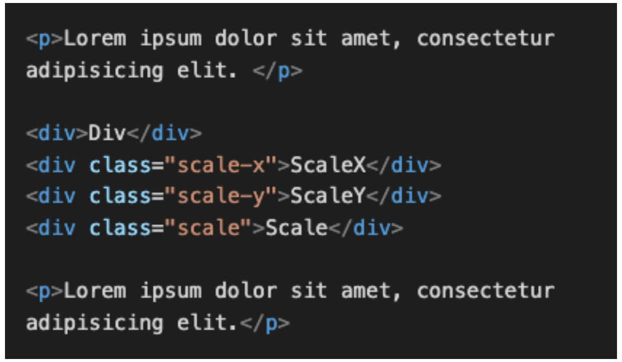
value > 1	: increment
0 < value < 1	: decrement

CSS - TRANSFORM

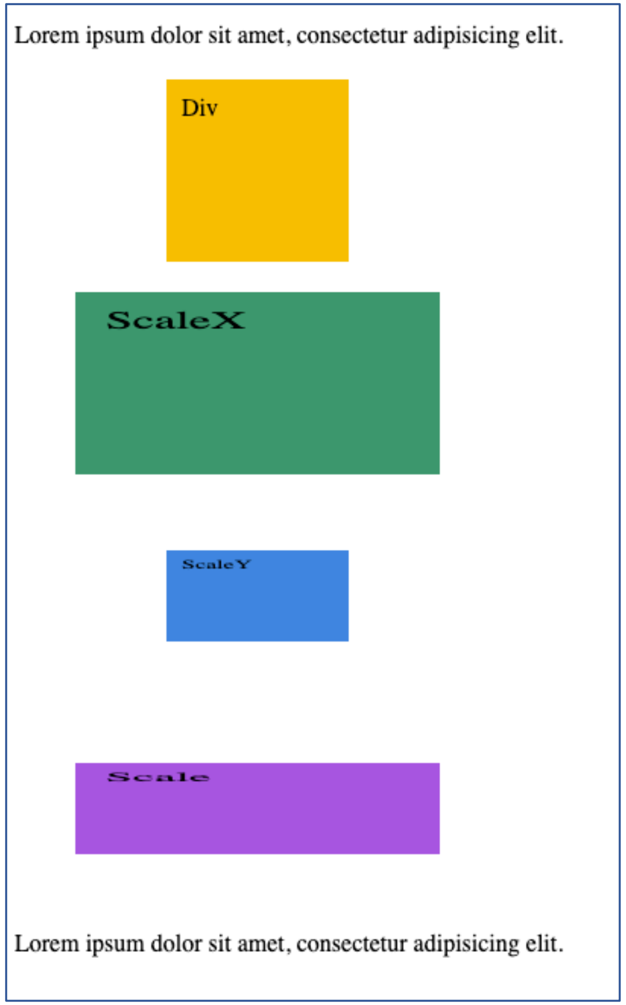
scale()



```css
div {
    height: 100px;
    width: 100px;
    margin: 20px 100px;
    background: #f9c004;
    padding: 10px;
}
.scale-x {
    background: #3d9970;
    transform: scaleX(2);
}
.scale-y {
    background: #4089e1;
    transform: scaleY(0.5);
}
.scale {
    background: #a957e4;
    transform: scale(2, 0.5);
}
```



```html
<p>Lorem ipsum dolor sit amet, consectetur
adipisicing elit. </p>

<div>Div</div>
<div class="scale-x">ScaleX</div>
<div class="scale-y">ScaleY</div>
<div class="scale">Scale</div>

<p>Lorem ipsum dolor sit amet, consectetur
adipisicing elit.</p>
```



Lorem ipsum dolor sit amet, consectetur adipisicing elit.

Div

ScaleX

ScaleY

Scale

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

NOTE : This also applies to the font-size, padding, height, and width of an element, too.

CSS - TRANSFORM

skew()

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

transform: skewX(deg);

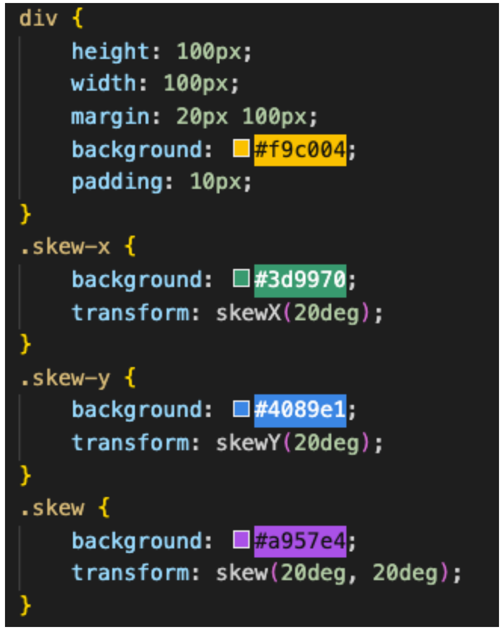
transform: skewY(deg);

transform: skew(x-deg, y-deg);

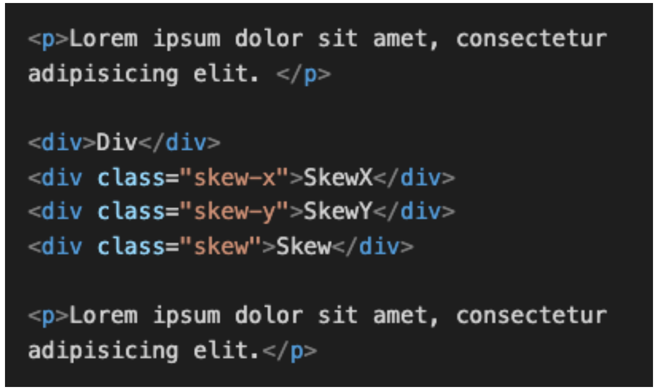
transform: skew(value-deg); → If the second parameter is not specified, it has a **zero value**. So, it will skew value-degree along the X-axis:

CSS - TRANSFORM

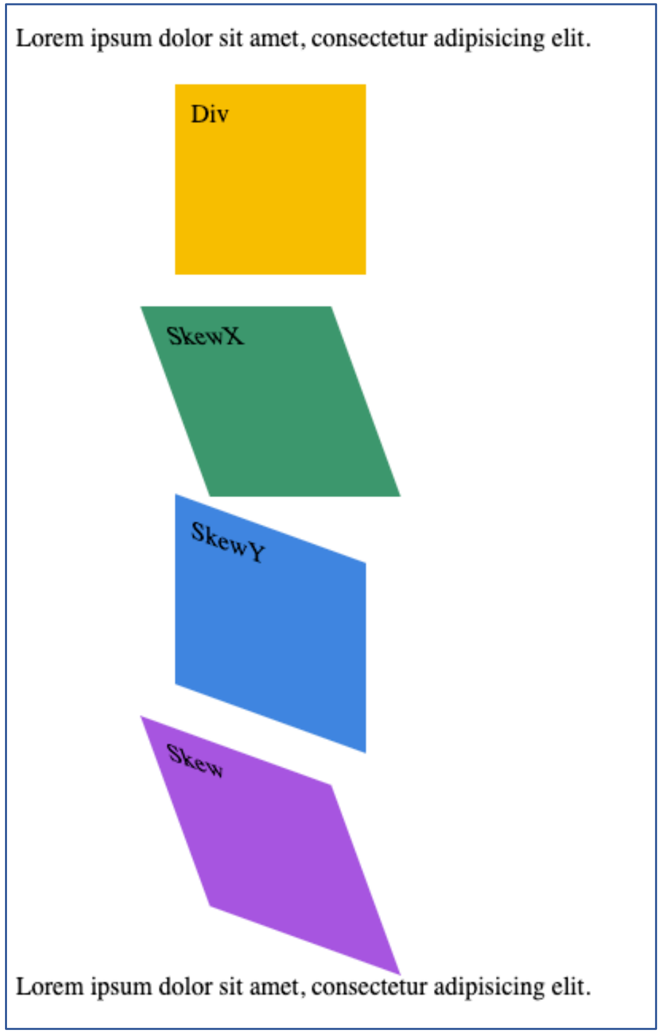
skew()



```css
div {
    height: 100px;
    width: 100px;
    margin: 20px 100px;
    background: #f9c004;
    padding: 10px;
}
.skew-x {
    background: #3d9970;
    transform: skewX(20deg);
}
.skew-y {
    background: #4089e1;
    transform: skewY(20deg);
}
.skew {
    background: #a957e4;
    transform: skew(20deg, 20deg);
}
```



```html
<p>Lorem ipsum dolor sit amet, consectetur
adipisicing elit. </p>

<div>Div</div>
<div class="skew-x">SkewX</div>
<div class="skew-y">SkewY</div>
<div class="skew">Skew</div>

<p>Lorem ipsum dolor sit amet, consectetur
adipisicing elit.</p>
```



Lorem ipsum dolor sit amet, consectetur adipisicing elit.

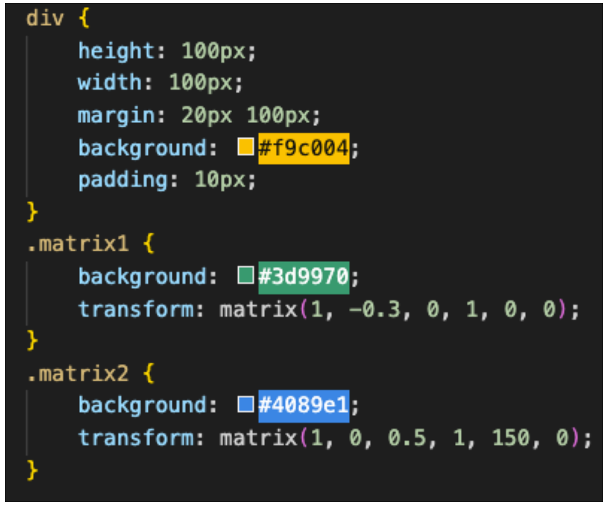
Lorem ipsum dolor sit amet, consectetur adipisicing elit.

CSS - TRANSFORM

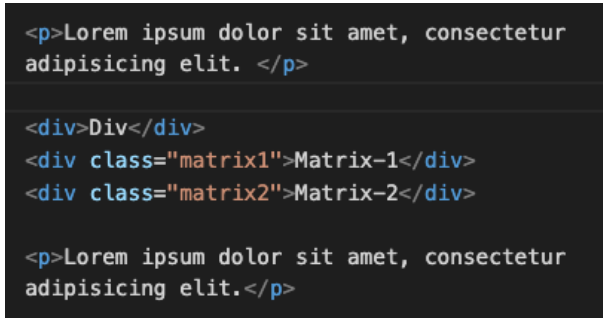
matrix()

- The **matrix()** method combines all the 2D transform methods into one.

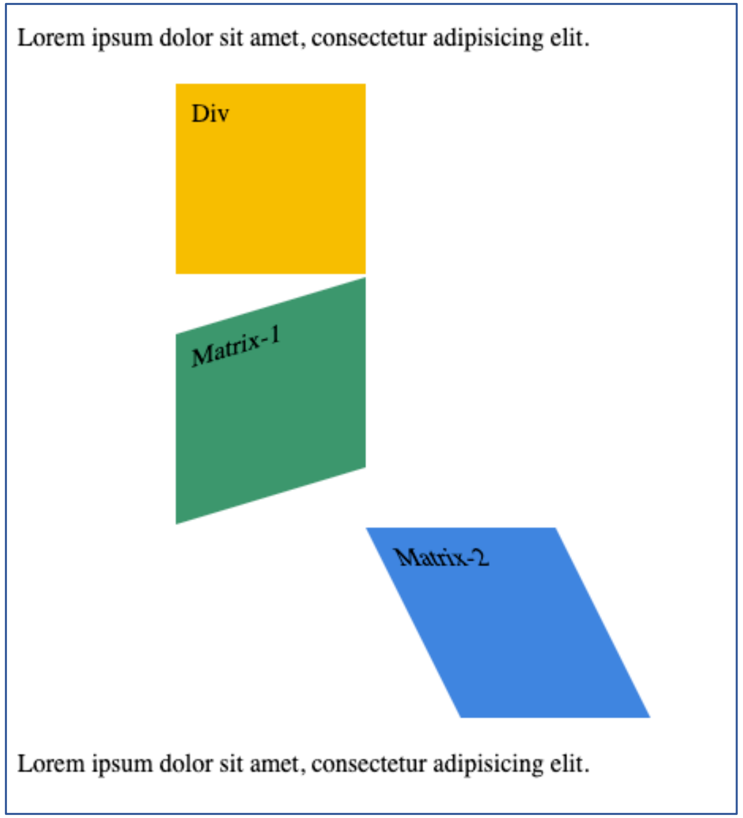
matrix (scaleX(), skewY(), skewX(), scaleY(), translateX(), translateY());



```
div {
    height: 100px;
    width: 100px;
    margin: 20px 100px;
    background: #f9c004;
    padding: 10px;
}
.matrix1 {
    background: #3d9970;
    transform: matrix(1, -0.3, 0, 1, 0, 0);
}
.matrix2 {
    background: #4089e1;
    transform: matrix(1, 0, 0.5, 1, 150, 0);
}
```



```
<p>Lorem ipsum dolor sit amet, consectetur
adipisicing elit. </p>

<div>Div</div>
<div class="matrix1">Matrix-1</div>
<div class="matrix2">Matrix-2</div>

<p>Lorem ipsum dolor sit amet, consectetur
adipisicing elit.</p>
```



Lorem ipsum dolor sit amet, consectetur adipisicing elit.

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

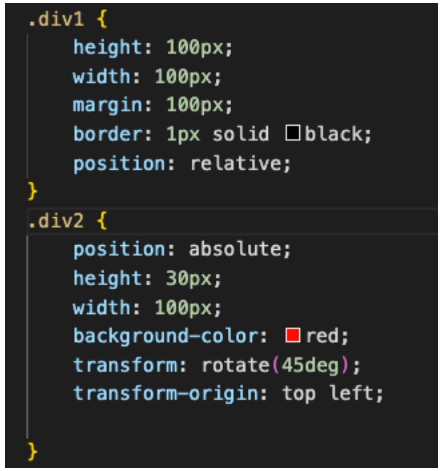
to convert a group of transforms into a single matrix declaration.

https://meyerweb.com/eric/tools/matrix/

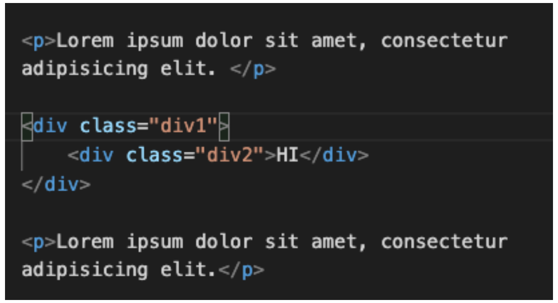
CSS – TRANSFORM-ORIGIN

- The **transform-origin** property is used in conjunction with CSS transforms, letting you change the point of origin of a transform.

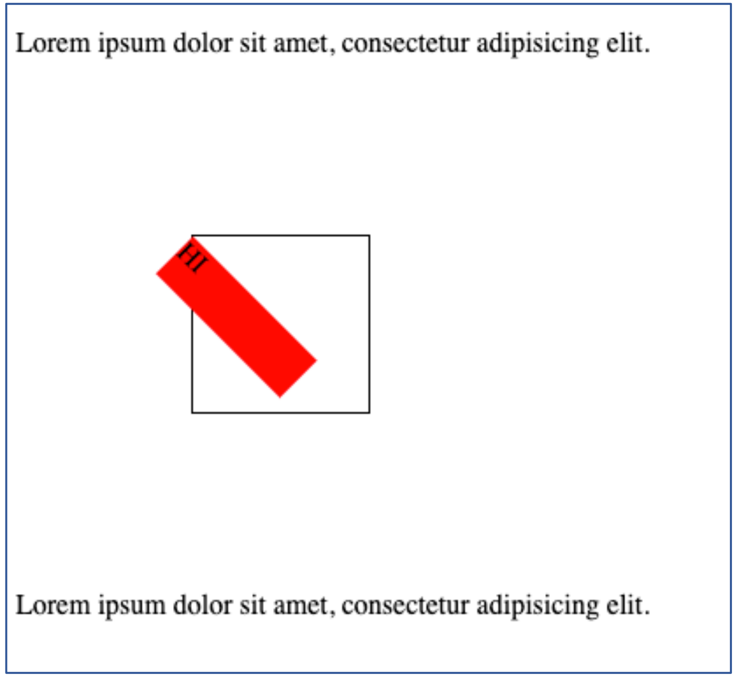
- Values can be lengths, percentages or the keywords top, left, right, bottom, and center.



```css
.div1 {
    height: 100px;
    width: 100px;
    margin: 100px;
    border: 1px solid □black;
    position: relative;
}
.div2 {
    position: absolute;
    height: 30px;
    width: 100px;
    background-color: 🟥red;
    transform: rotate(45deg);
    transform-origin: top left;

}
```



```html
<p>Lorem ipsum dolor sit amet, consectetur
adipisicing elit. </p>

<div class="div1">
    <div class="div2">HI</div>
</div>

<p>Lorem ipsum dolor sit amet, consectetur
adipisicing elit.</p>
```



Lorem ipsum dolor sit amet, consectetur adipisicing elit.

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

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