CT053-3-1
Fundamentals of Web Design & Development

CSS3 Transforms & Transitions

CSS3 Transforms

- CSS3 transforms allow you to translate, rotate, scale, and skew elements.
- A transformation is an effect that lets an element change shape, size and position.
- CSS3 supports 2D and 3D transformations.
- 2D vs. 3D Transformation sample

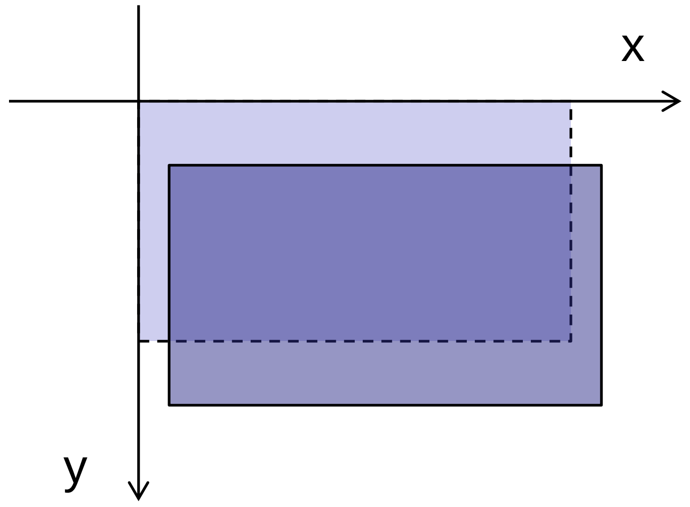
2D transformation methods

- translate()
- rotate()
- scale()
- skewX()
- skewY()
- matrix()

translate()

- The translate() method moves an element from its current position (according to the parameters given for the X-axis and the Y-axis).
- The following example moves the <div> element 50 pixels to the right, and 100 pixels down from its current position.

```
div {
    width: 300px;
    height: 100px;
    background-color: yellow;
    border: 1px solid black;
    -ms-transform: translate(50px,100px); /* IE 9 */
    -webkit-transform: translate(50px,100px); /* Safari */
    transform: translate(50px,100px); /* Standard syntax */
}
```

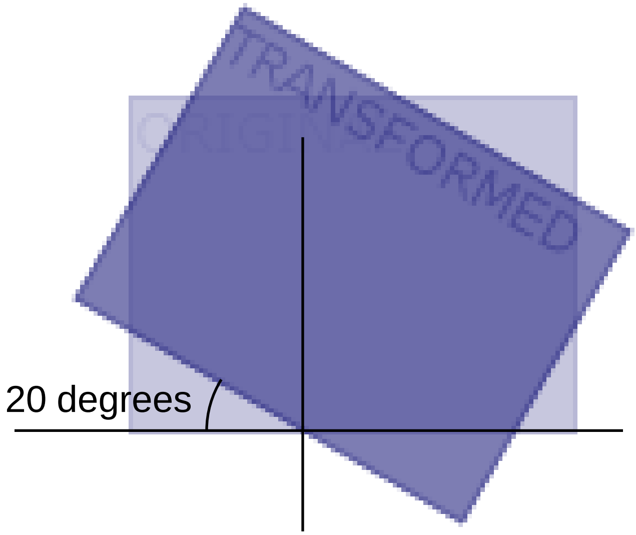


Some of the new CSS3 properties haven't yet been finalized and many of the browsers implemented early versions of these new ideas, and use the -name- prefix (like -webkit-, -moz-, -o-, and -ms-). So, to get your CSS to work in multiple browsers, you have to supply all the browser specific versions, plus the version that we think will eventually be the one that gets finalized in the standard.

rotate()

- The rotate() method rotates an element clockwise or counter-clockwise according to a given degree.
- The following example rotates the <div> element clockwise with 20 degrees

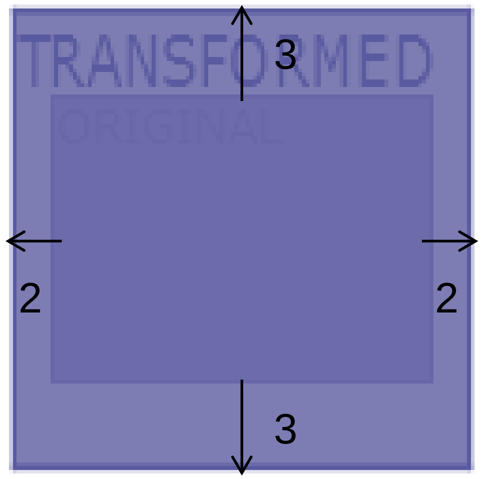
```
div {
    -ms-transform: rotate(20deg); /* IE 9 */
    -webkit-transform: rotate(20deg); /* Safari */
    transform: rotate(20deg);
}
```



scale()

- The scale() method increases or decreases the size of an element (according to the parameters given for the width and height).
- The following example increases the <div> element to be two times of its original width, and three times of its original height:

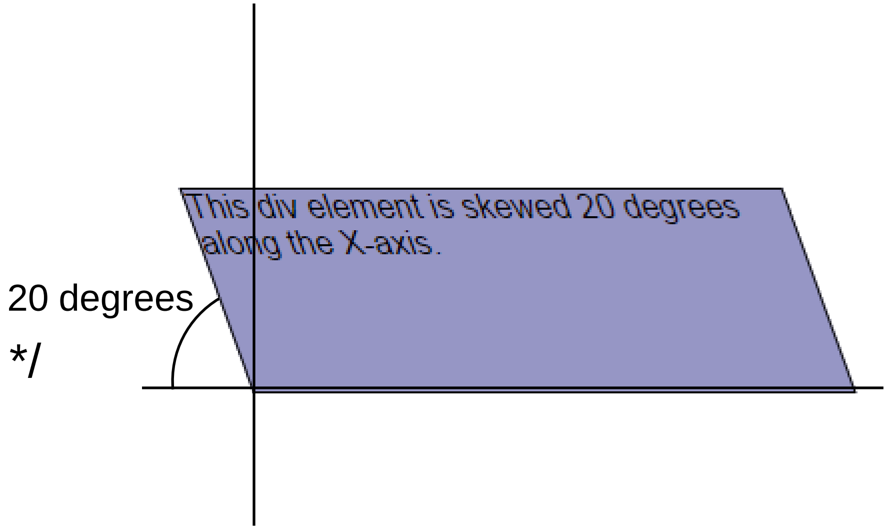
```
div {
    -ms-transform: scale(2, 3); /* IE 9 */
    -webkit-transform: scale(2, 3); /* Safari */
    transform: scale(2, 3);
}
```



skewX()

- The skewX() method skews an element along the X-axis by the given angle.
- The following example skews the <div> element 20 degrees along the X-axis:

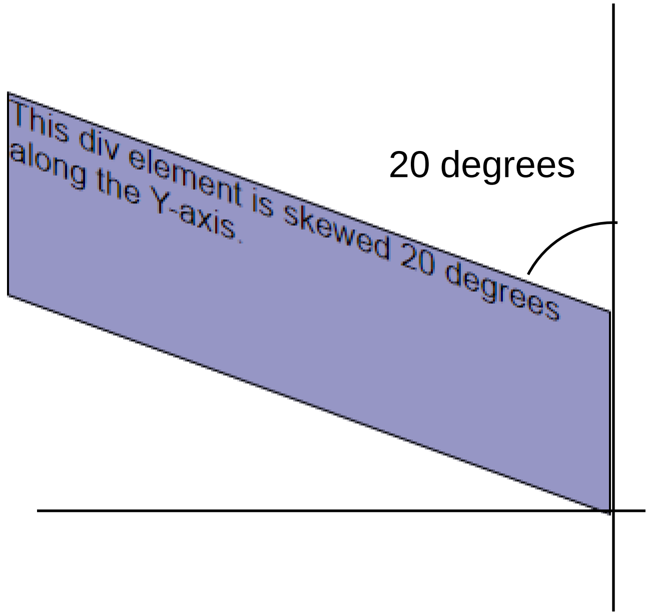
```
div {
    -ms-transform: skewX(20deg); /* IE 9 */
    -webkit-transform: skewX(20deg); /* Safari */
    transform: skewX(20deg);
}
```



skewY()

- The skewY() method skews an element along the Y-axis by the given angle.
- The following example skews the <div> element 20 degrees along the Y-axis:

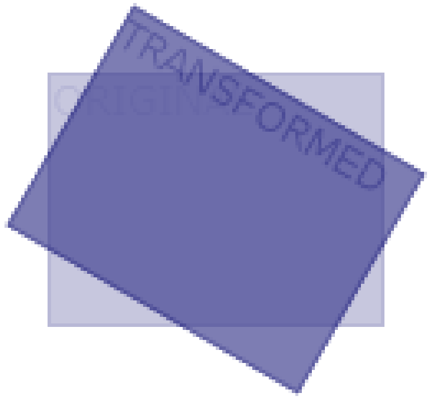
```
div {
    -ms-transform: skewY(20deg); /* IE 9 */
    -webkit-transform: skewY(20deg); /* Safari */
    transform: skewY(20deg);
}
```



matrix()

- The matrix() method combines all the 2D transform methods into one.
- The matrix() method take 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 {
    -ms-transform: matrix(1, -0.3, 0, 1, 0, 0); /* IE 9 */
    -webkit-transform: matrix(1, -0.3, 0, 1, 0, 0); /* Safari */
    transform: matrix(1, -0.3, 0, 1, 0, 0);
}
```



2D Transform Methods - Summary

Function	Description
matrix(*n,n,n,n,n,n*)	Defines a 2D transformation, using a matrix of six values
translate(*x,y*)	Defines a 2D translation, moving the element along the X- and the Y-axis
translateX(*n*)	Defines a 2D translation, moving the element along the X-axis
translateY(*n*)	Defines a 2D translation, moving the element along the Y-axis
scale(*x,y*)	Defines a 2D scale transformation, changing the elements width and height
scaleX(*n*)	Defines a 2D scale transformation, changing the element's width
scaleY(*n*)	Defines a 2D scale transformation, changing the element's height
rotate(*angle*)	Defines a 2D rotation, the angle is specified in the parameter
skew(*x-angle,y-angle*)	Defines a 2D skew transformation along the X- and the Y-axis
skewX(*angle*)	Defines a 2D skew transformation along the X-axis
skewY(*angle*)	Defines a 2D skew transformation along the Y-axis

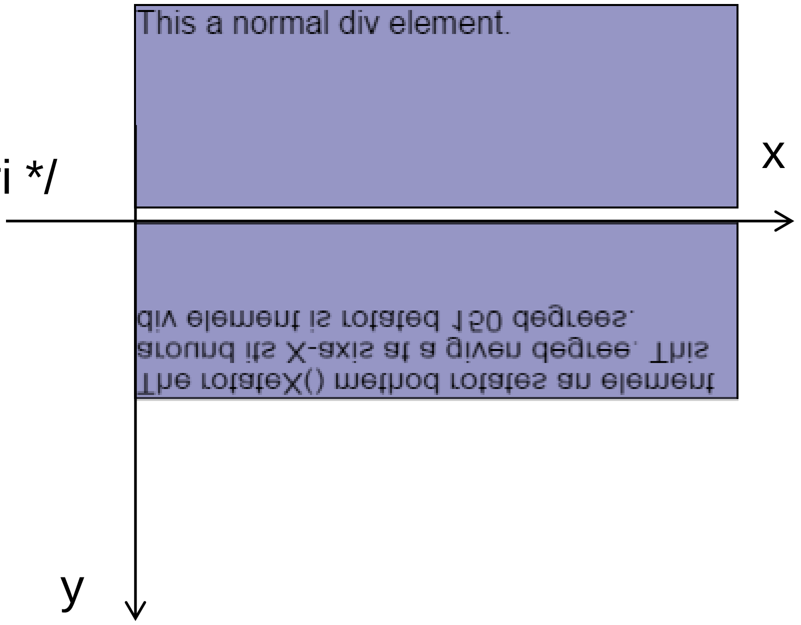
3D transformation methods

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

rotateX()

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

```
div {
    -webkit-transform: rotateX(150deg); /* Safari */
    transform: rotateX(150deg);
}
```

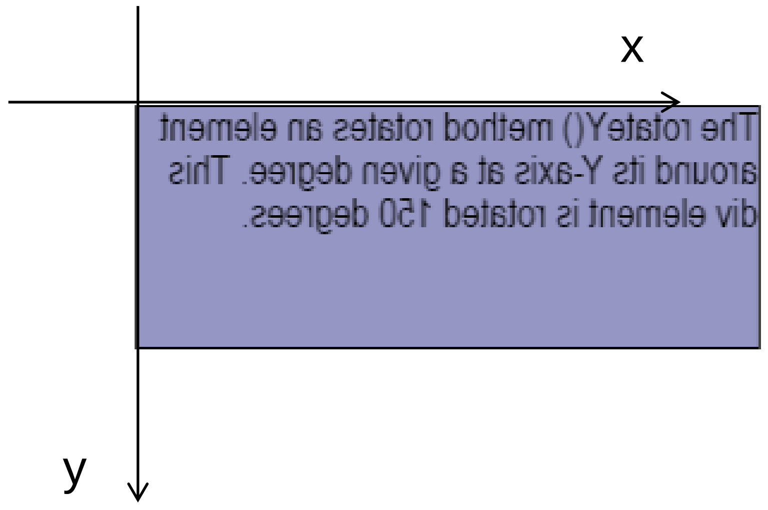


The rotateX() method rotates an element around its X-axis at a given degree. This div element is rotated 150 degrees.

rotateY()

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

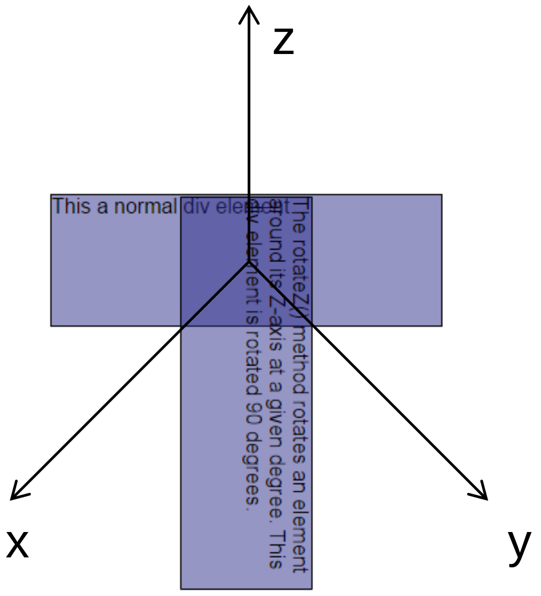
```
div {
    -webkit-transform: rotateY(130deg); /* Safari */
    transform: rotateY(130deg);
}
```


x
The rotateY() method rotates an element
around its Y-axis at a given degree. This
div element is rotated 150 degrees.
y

rotateZ()

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

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


z
x
y
This a normal div element.
The rotateZ() method rotates an element around its Z-axis at a given degree. This div element is rotated 90 degrees.

3D Transform Methods - Summary

Function	Description
matrix3d (*n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n*)	Defines a 3D transformation, using a 4x4 matrix of 16 values
translate3d(*x,y,z*)	Defines a 3D translation
translateX(*x*)	Defines a 3D translation, using only the value for the X-axis
translateY(*y*)	Defines a 3D translation, using only the value for the Y-axis
translateZ(*z*)	Defines a 3D translation, using only the value for the Z-axis
scale3d(*x,y,z*)	Defines a 3D scale transformation
scaleX(*x*)	Defines a 3D scale transformation by giving a value for the X-axis
scaleY(*y*)	Defines a 3D scale transformation by giving a value for the Y-axis
scaleZ(*z*)	Defines a 3D scale transformation by giving a value for the Z-axis
rotate3d(*x,y,z,angle*)	Defines a 3D rotation
rotateX(*angle*)	Defines a 3D rotation along the X-axis
rotateY(*angle*)	Defines a 3D rotation along the Y-axis
rotateZ(*angle*)	Defines a 3D rotation along the Z-axis
perspective(*n*)	Defines a perspective view for a 3D transformed element

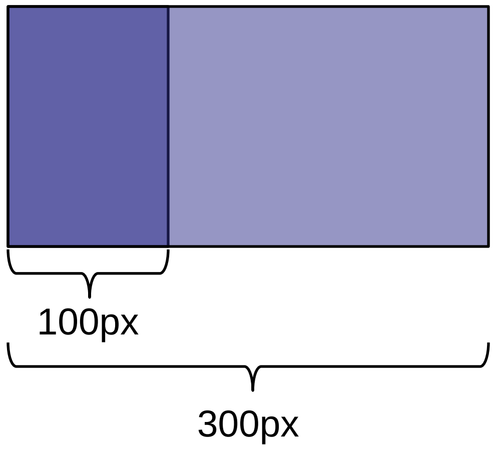
CSS3 Transitions

- CSS3 transitions allows you to change property values smoothly (from one value to another), over a given duration.
- To create a transition effect, you must specify two things:
  - the CSS property you want to add an effect to
  - the duration of the effect
- If the duration part is not specified, the transition will have no effect, because the default value is 0.

Transition Example

```
div {
    width: 100px;
    height: 100px;
    background: red;
    -webkit-transition: width 2s; /* For Safari 3.1 to 6.0 */
    transition: width 2s;
}

div:hover {
    width: 300px;
}
```

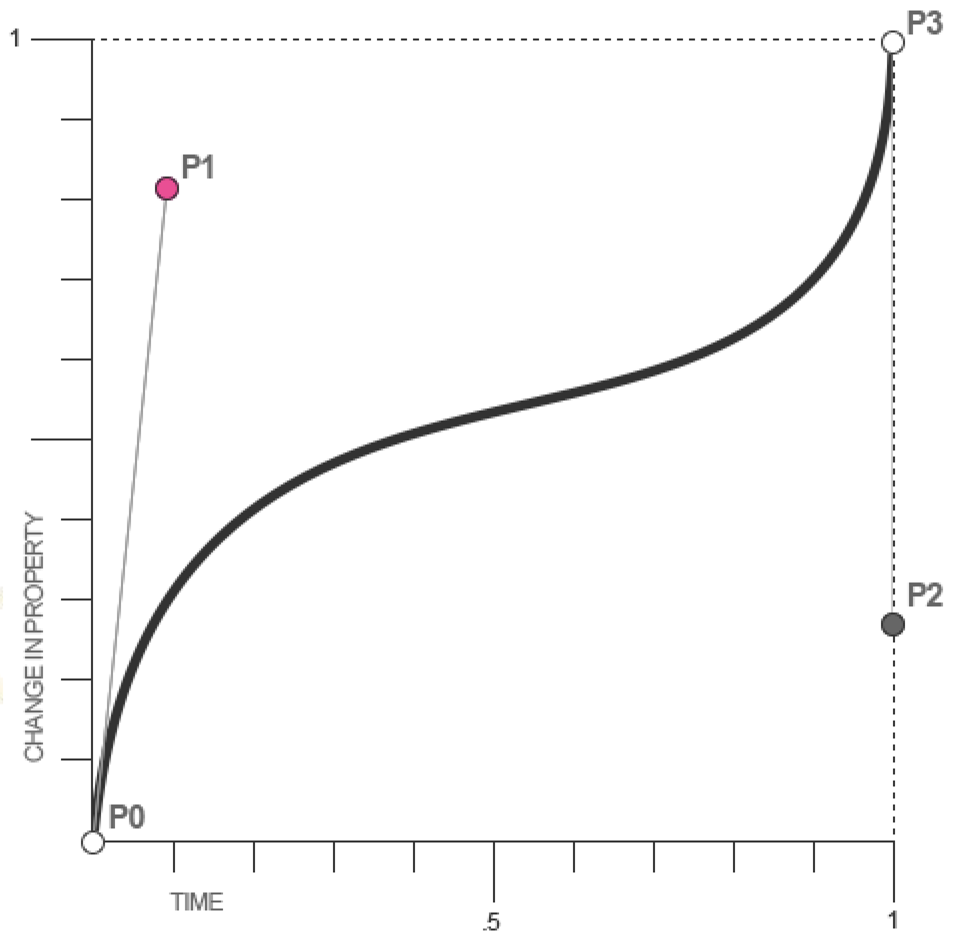


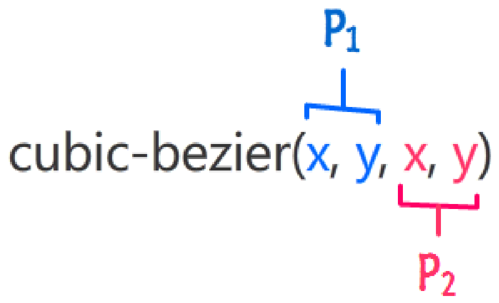
Specify the Speed Curve of the Transition

- The transition-timing-function property specifies the speed curve of the transition effect.
- The transition-timing-function property can have the following values:
  - **ease** - specifies a transition effect with a slow start, then fast, then end slowly (this is default)
  - **linear** - specifies a transition effect with the same speed from start to end
  - **ease-in** - specifies a transition effect with a slow start
  - **ease-out** - specifies a transition effect with a slow end
  - **ease-in-out** - specifies a transition effect with a slow start and end
  - **cubic-bezier(n,n,n,n**) - lets you define your own values in a cubic-bezier function

sample

cubic-bezier(n,n,n,n)





- For CSS Bézier Curves, P0 and P3 are always in the same spot.
- P0 is at (0,0) and P3 is at (1,1).
- An important thing to note is that the points that get passed in the cubic-bezier function can only be between 0 and 1.
- The x-axis is the time the animation takes, and the y-axis is the property being changed.

- Based on the graph on the left, you can visualize it easing quickly in the beginning, slowing down in the middle, and picking up speed at the end.

Delay the Transition Effect

- The transition-delay property specifies a delay (in seconds) for the transition effect.
- The following example has a 1 second delay before starting:

```
div {
    -webkit-transition-delay: 1s; /* Safari */
    transition-delay: 1s;
}
```

sample

Transition + Transformation

- The following example also adds a transformation to the transition effect:

```
div {
    width: 100px;
    height: 100px;
    background: red;
    transition: width 2s, height 2s, transform 2s;
}

div:hover {
    width: 300px;
    height: 300px;
    transform: rotate(180deg);
}
```

Can you describe the effect?

Q & A