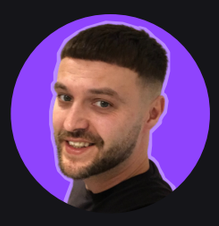
Learn CSS By Use Cases

ebook by Joe Harrison

Advanced transform



Joe Harrison
@frontendjoe

transform

Advanced Properties

Intro

The transform property allows us to apply both 2D and 3D transformations to an element. I'll only show 2D use cases in this ebook.

Syntax

As we've already learnt, some properties accept a variety of functions. Transform is solely made up of function value type arguments that do a range of different things.

```
.active { transform: scaleX(1.25) scaleY(1.25); }
```

transform

Advanced Properties

Special Power

In CSS, the power of transform really comes into play when creating animations. They’re simple and smooth with the least amount of jank possible. Translate is useful when working with position absolute to center an element on itself.

Tips

Scaling or zooming animations have been known to trigger migraines. For websites that have a lot of traffic you should consider providing a control to disable these animations. In my opinion the transform property is a “nice-to-have” in CSS – many non-animated things it does, can be achieved with other properties.

scale

Alter size of elements

.active

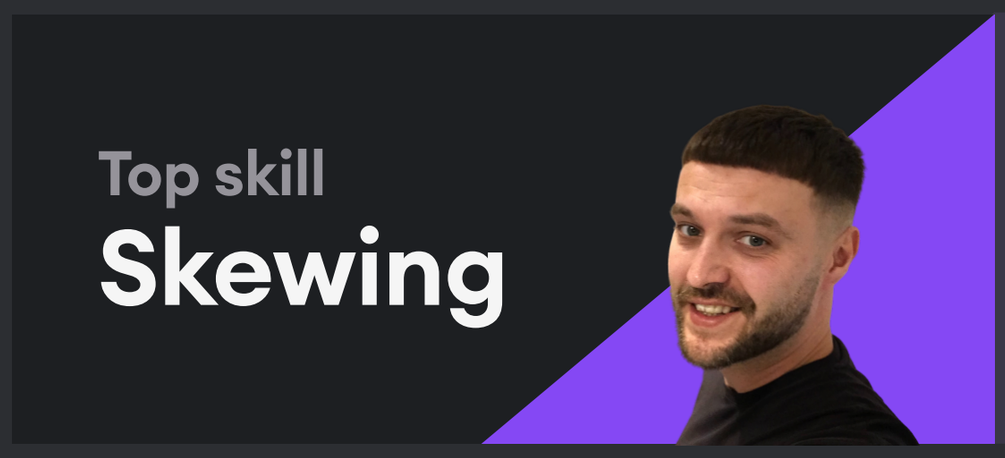
```
/* longhand - one function (X, Y) */
.page.active {
  transform: scale(1.25, 1.25);
}

/* shorthand - one function (X & Y) */
.page.active {
  transform: scale(1.25);
}
```

Function implementation same as **scale**

skew

Abstract shapes

Parallelogram
(skewed rectangle)



Triangle with
overflow: hidden

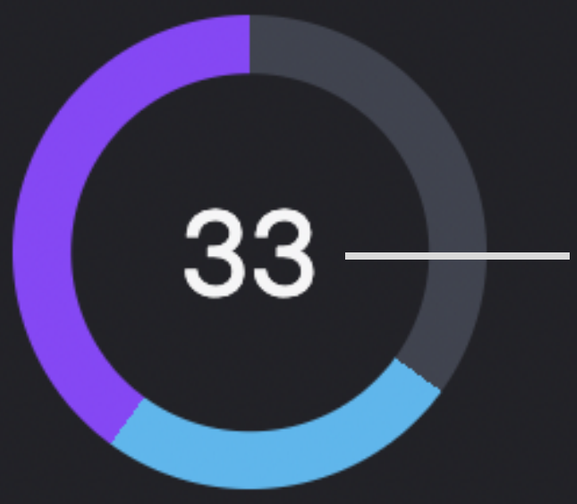
```
.background {
  width: 16rem;
  height: 5rem;
  background: #8548F4;
  transform: skew(45deg);
}
```

One param sets both X and Y angles the same

Function implementation same as **scale**

translate

Intelligent centering



.chart-label will be placed perfectly in the center

```
.chart { position: relative; }

.chart-label {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
}
```

offset properties will not work how we want without transform

x y

% inside translate is of itself (very useful)

Function implementation **NOT** same as **scale**

rotate (2D)

Rotated 2D elements

Diamond shapes by rotating a square

```
.diamond {
  width: 4rem;
  height: 4rem;
  background: #8F44FD;
  transform: rotate(45deg);
}
```

transform

Advanced Properties

Knowledge Gained

- 🏆 Transform accepts only function arguments (scale, skew, rotate, translate)
- 🏆 Translate is useful when we need to center an element on itself
- 🏆 2D rotating can be useful in static website design – where as, 3D rotating is more pertinent in animation
- 🏆 Transform should not be overused in static (non-animated) websites

Joe Harrison
@frontendjoe