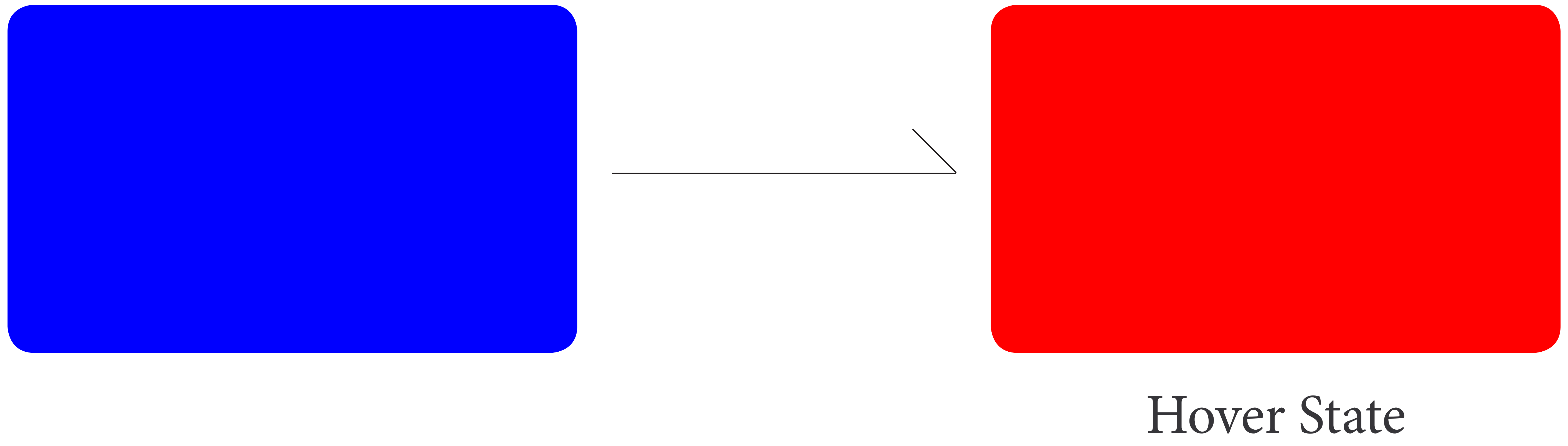
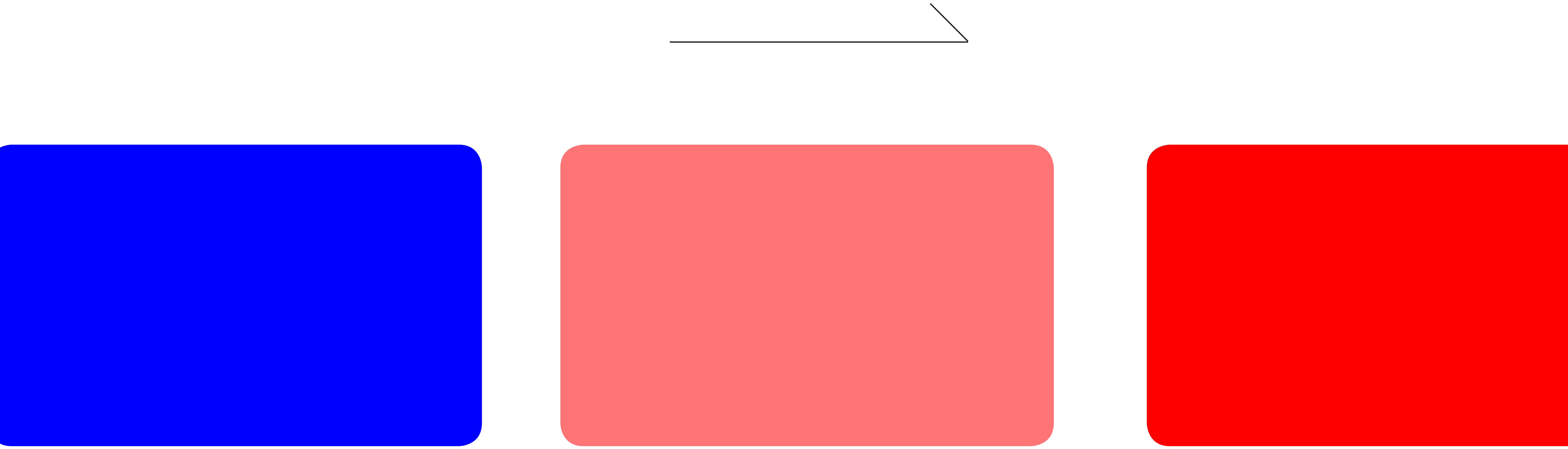
Web Animations with CSS

What is a transition?

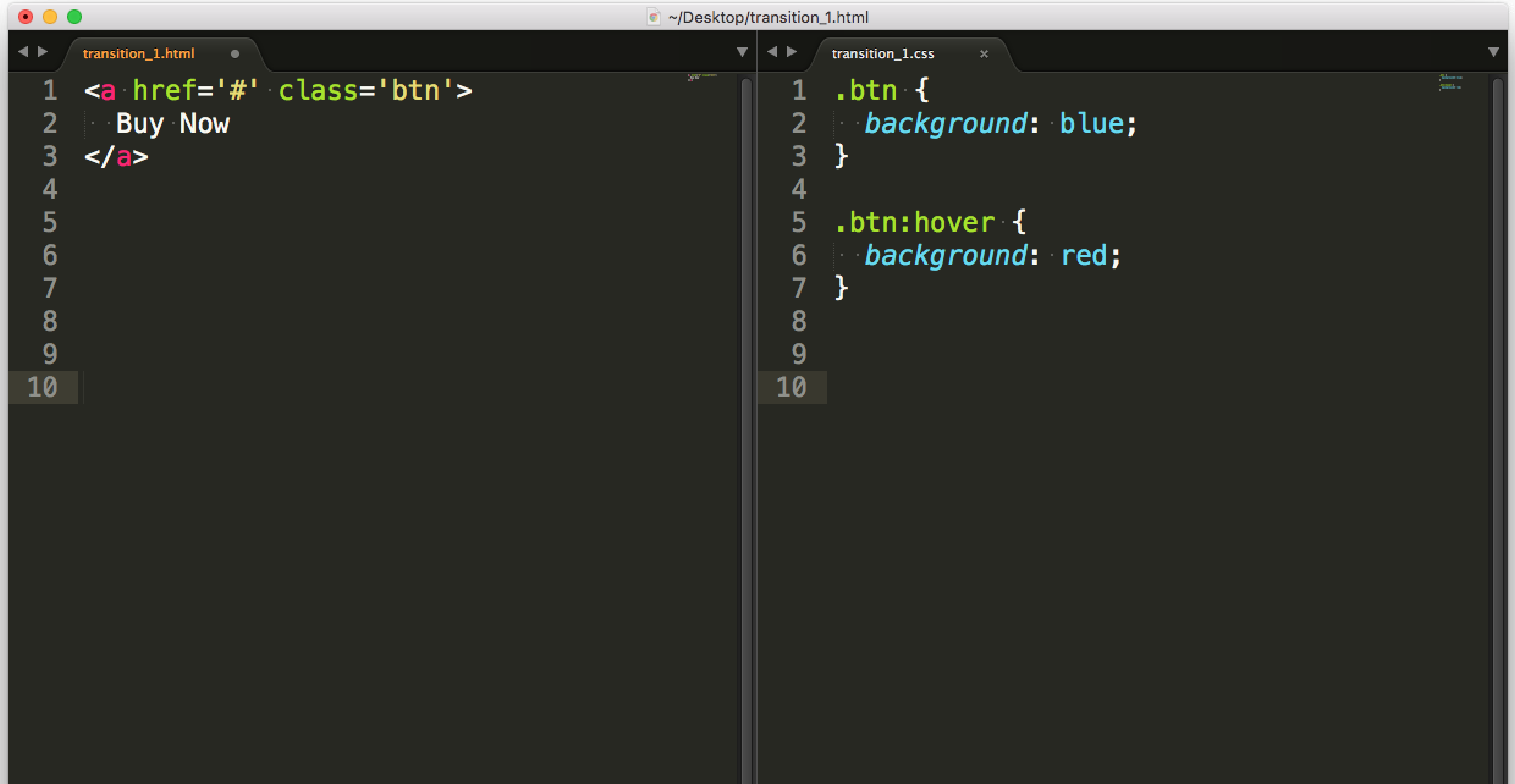
- Transitions cause changes to property and take place over a period of time.



Transitions allow the change to happen over a “duration” of time



First style start and end state



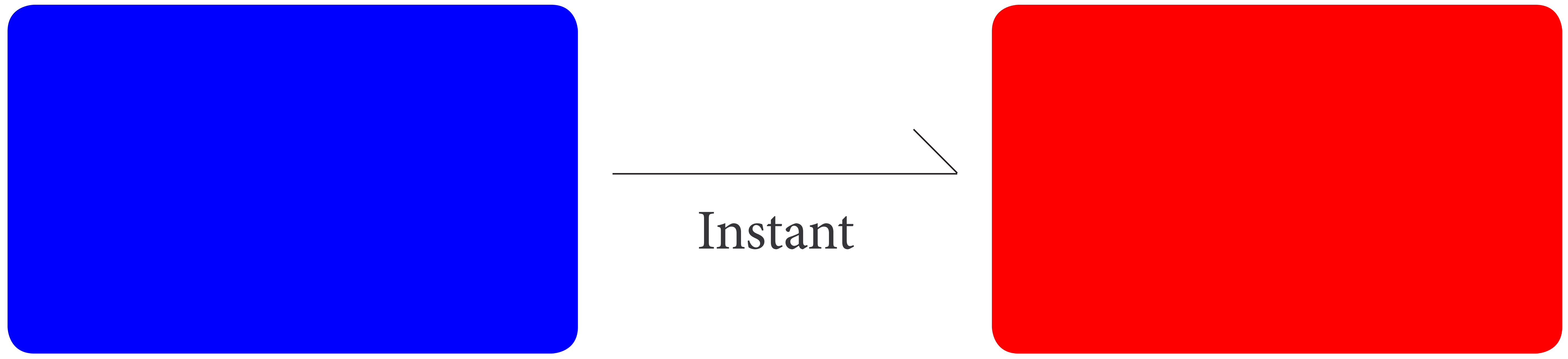
~/Desktop/transition_1.html

transition_1.html

```html
<a href='#' class='btn'>
  Buy Now
</a>
```

transition_1.css

```css
.btn {
  background: blue;
}

.btn:hover {
  background: red;
}
```

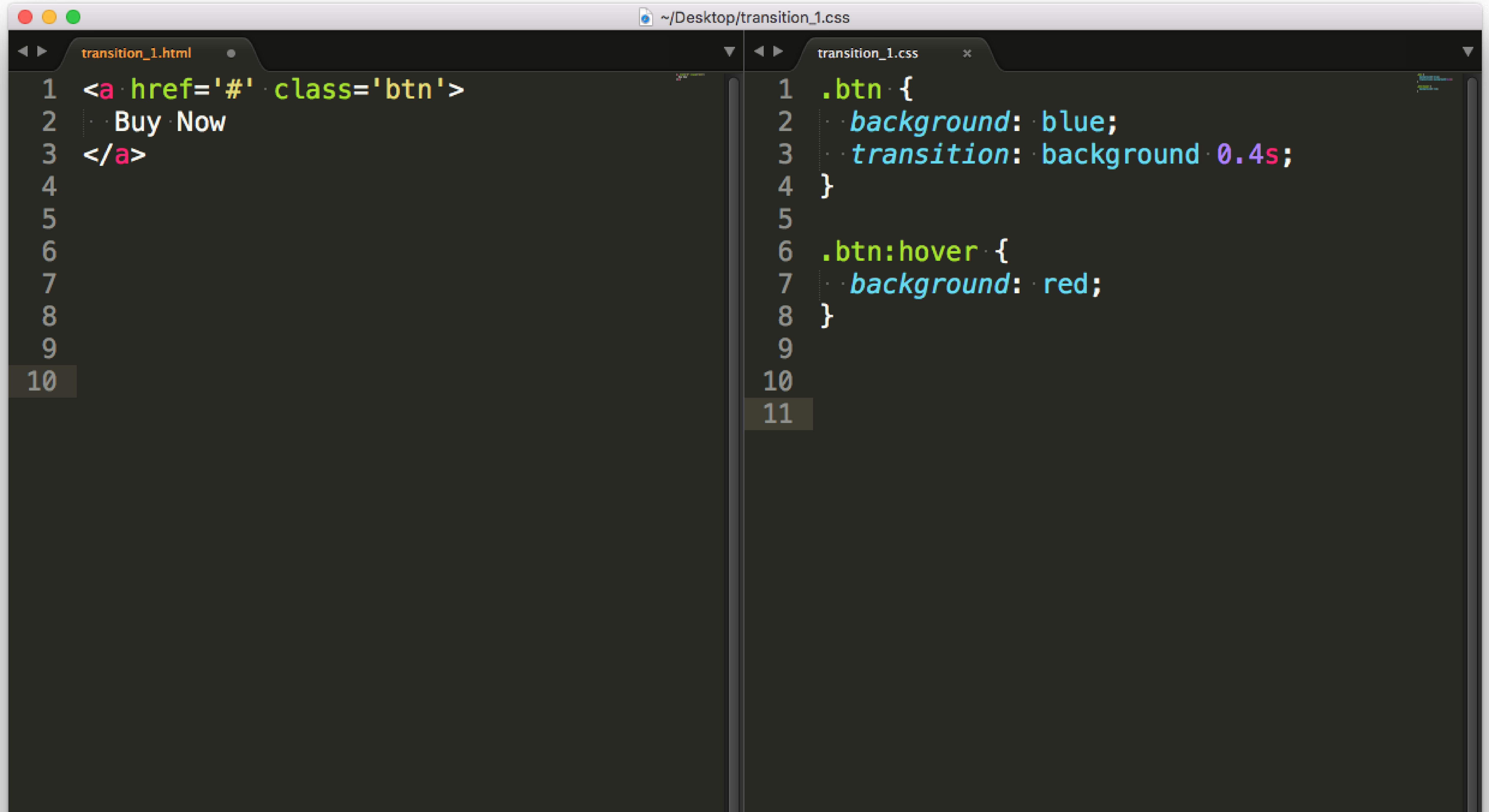

Instant

Transition Recipe

It works like this:

```
transition: <property> <duration>;
```

Transition Recipe



```html
<a href='#' class='btn'>
  Buy Now
</a>
```

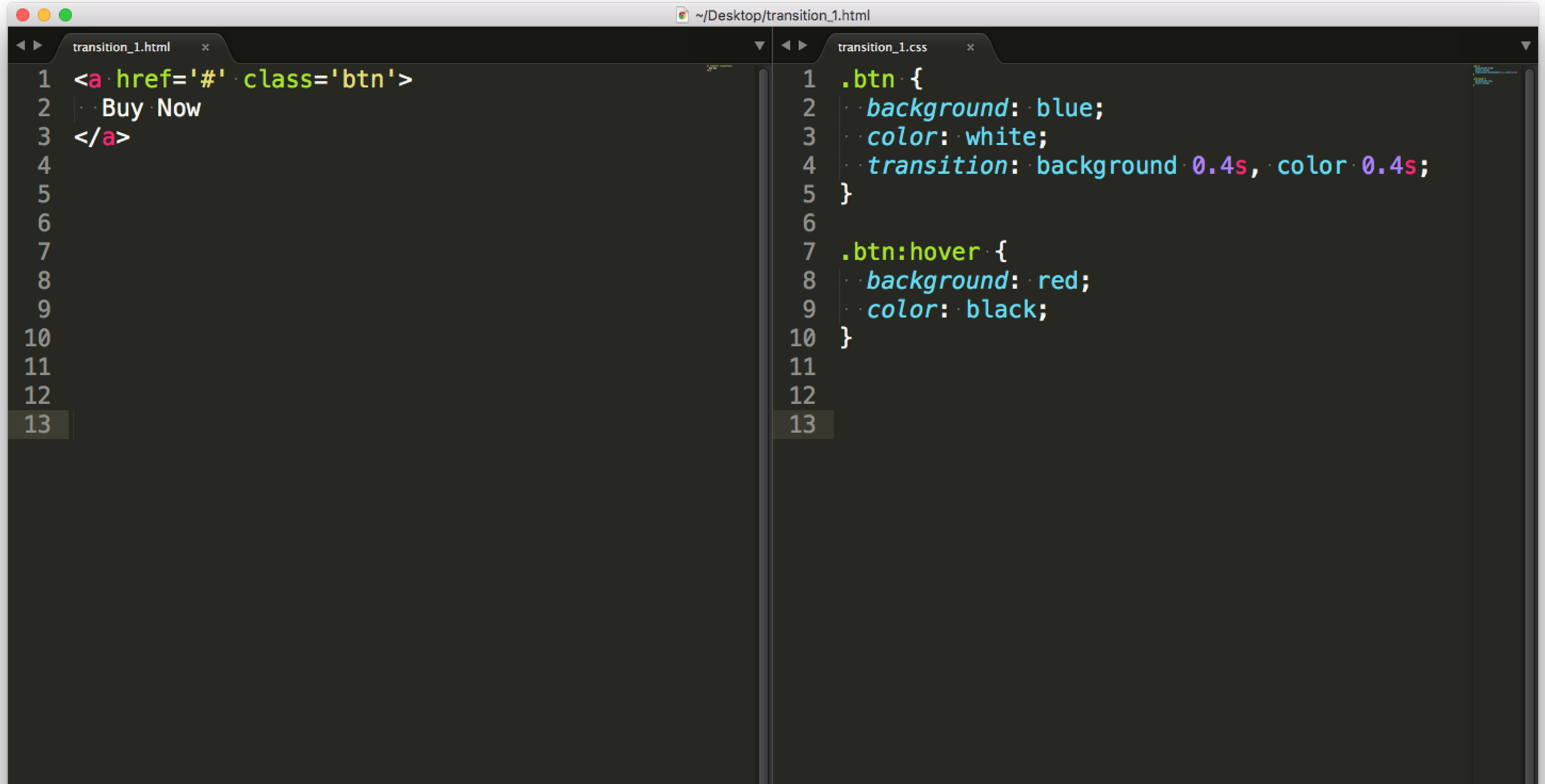
```css
.btn {
  background: blue;
  transition: background 0.4s;
}

.btn:hover {
  background: red;
}
```

Transition Multiple Properties

- You can transition multiple comma-separated properties

```
transition: <property> <duration>, <property> <duration>;
```

Transition Multiple Properties



~/Desktop/transition_1.html

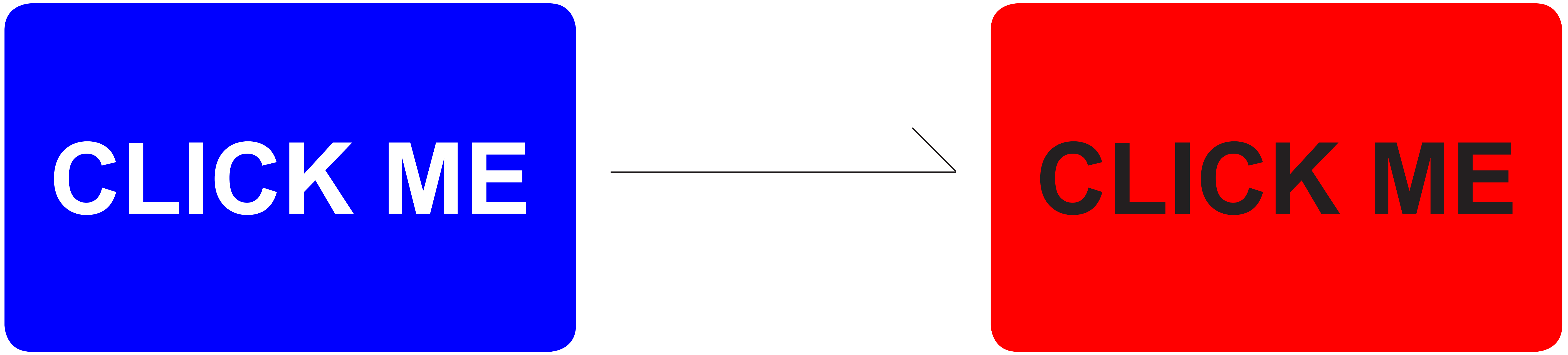
transition_1.html

```html
<a href='#' class='btn'>
  Buy Now
</a>
```

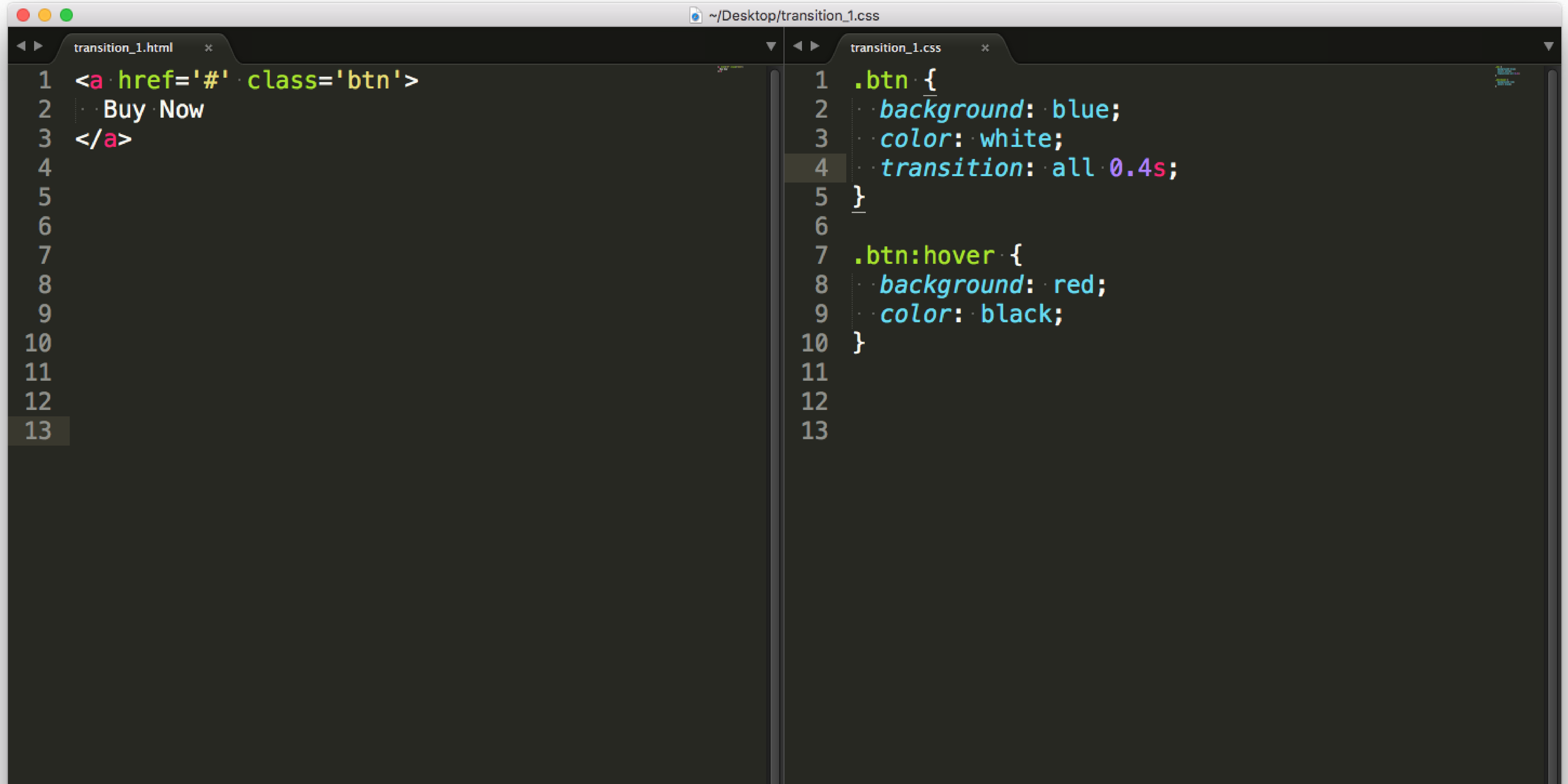
transition_1.css

```css
.btn {
  background: blue;
  color: white;
  transition: background 0.4s, color 0.4s;
}

.btn:hover {
  background: red;
  color: black;
}
```

Transition Multiple Properties



A Keyword to Transition All Properties



~/Desktop/transition_1.css

transition_1.html

```html
<a href='#' class='btn'>
  Buy Now
</a>
```

transition_1.css

```css
.btn {
  background: blue;
  color: white;
  transition: all 0.4s;
}

.btn:hover {
  background: red;
  color: black;
}
```

Full Transition Recipe

- Order is irrelevant as long as you have your duration number specified before the delay number.

```
.btn {
  transition: <property> <duration> <timing-function> <delay>;
}
```

Defaults

- these properties are given defaults if you don’t specify something for each of them.

```
.btn {

  transition: <property> <duration> <timing-function> <delay>;

}
```

Default: ‘all’

Default: ‘ease’

Default: 0

When to use vendor prefixes

- All of these examples have been without vendor prefixes, but you might need to include them. They look like this:

```
.btn {
  -webkit-transition: <property> <duration>;
  -moz-transition: <property> <duration>;
  –ms-transition: <property> <duration>;
  transition: <property> <duration>;
}
```

When to use vendor prefixes

Use a site like caniuse.com to check

What can you use transition between?

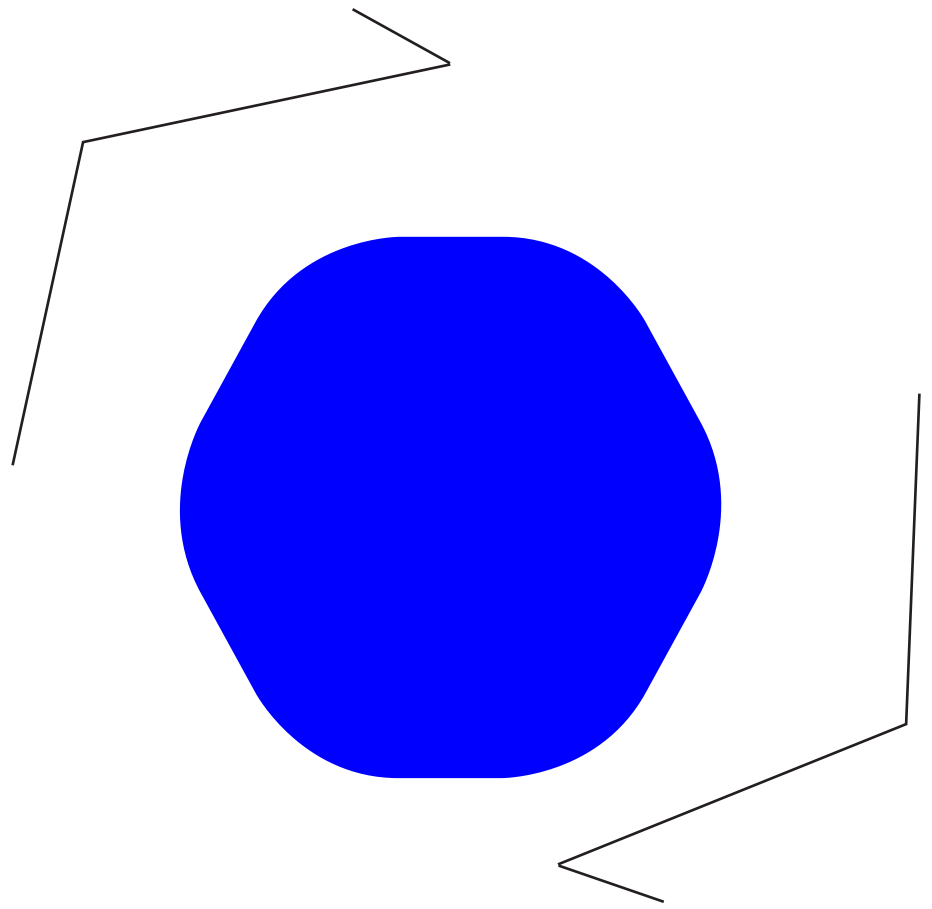
Basically any css property that has a middle state.

Easier to think about what you CAN’T transition between.

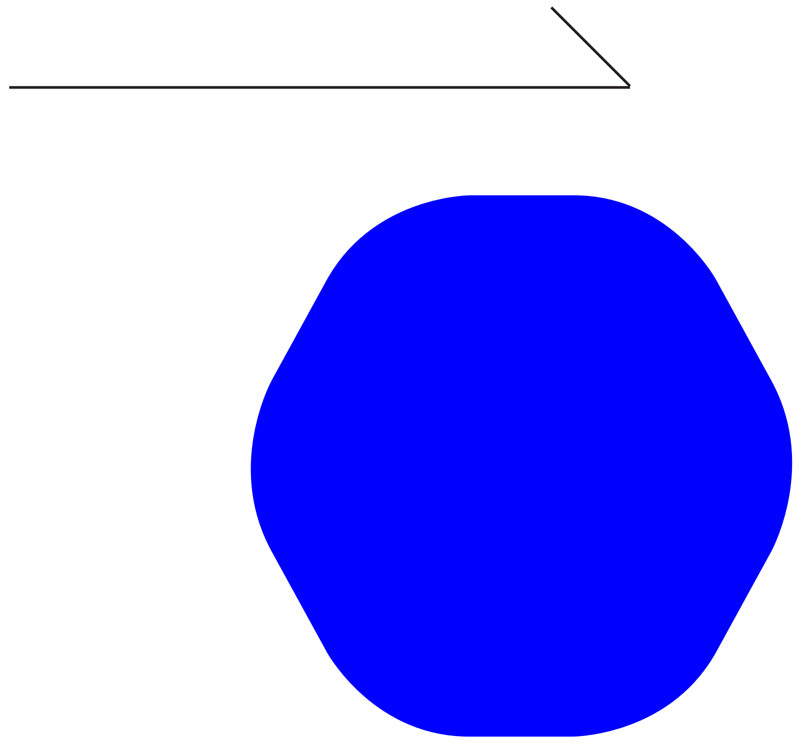
What about the ‘display’ property?

What is a transform?

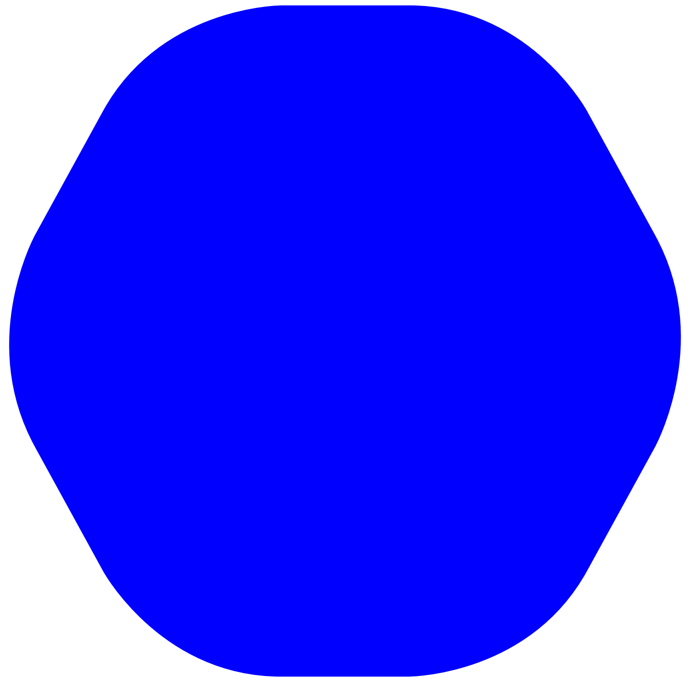
- CSS tranfromation let you modify elements in their coordinate space. They can be rotated, translated, scaled, and skewed



Rotate



Translate



scale

Rotate recipe

- Rotate takes any value with a degree or turn suffix.

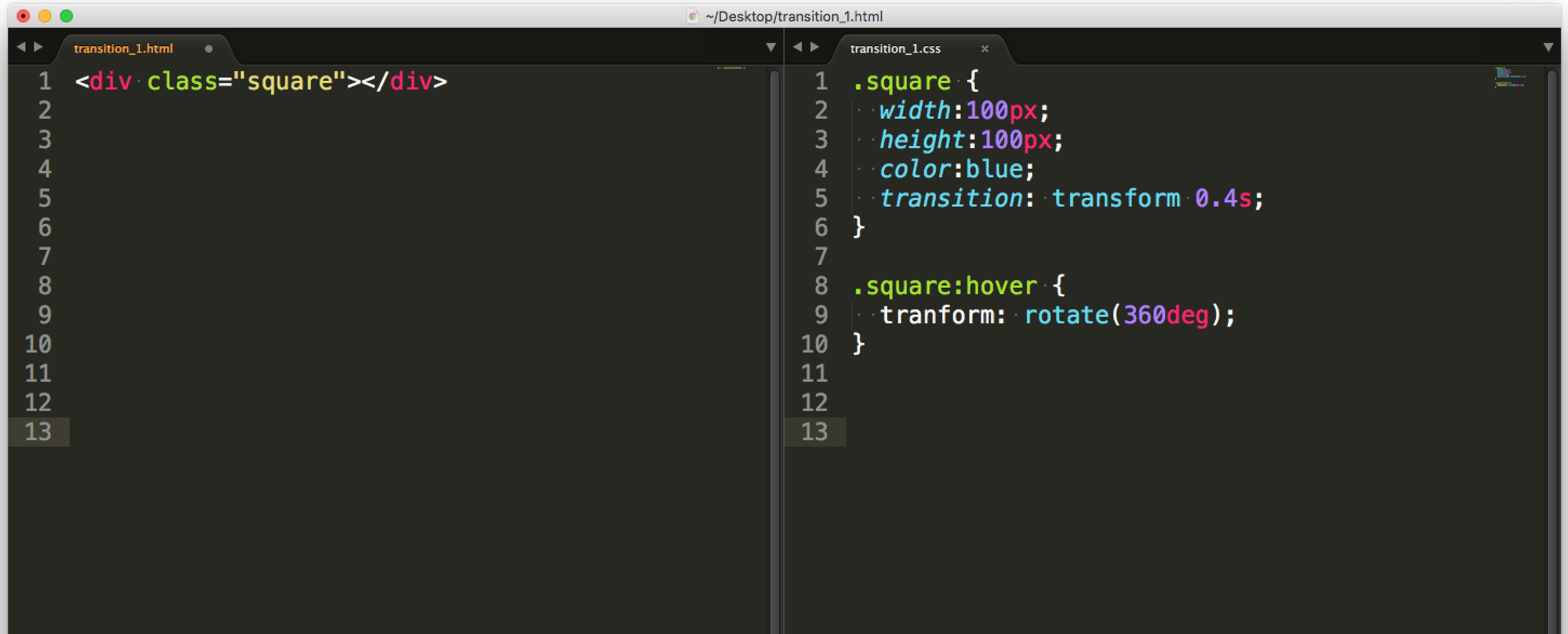
```
tranform: rotate(360deg);
```

or

```
tranform: rotate(1turn);
```

Rotate + Transition

- Rotate still needs to happen over a duration! So you need to combine it with a transition to SEE it happen.


~/Desktop/transition_1.html

transition_1.html

```html
<div class="square"></div>
```

transition_1.css

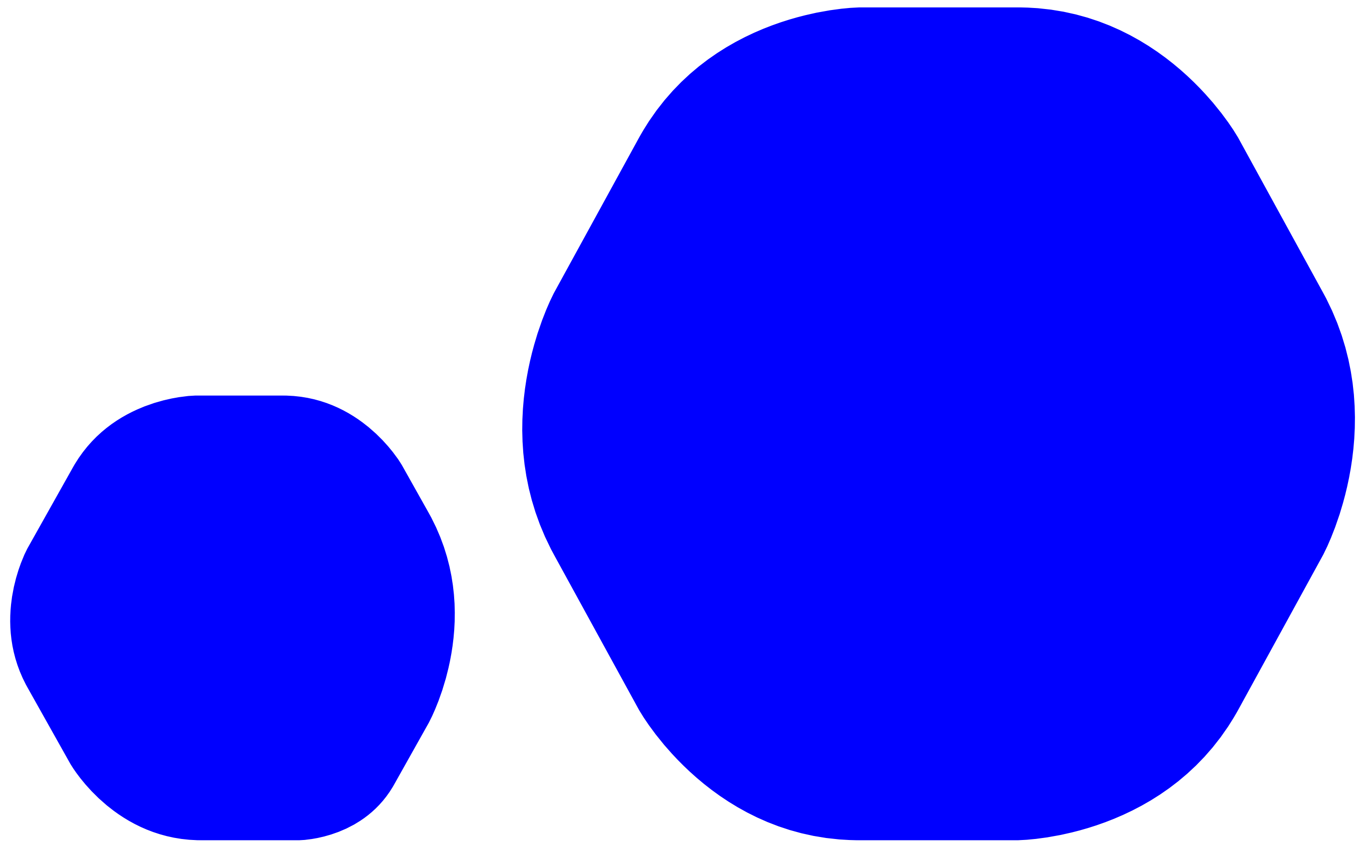
```css
.square {
  width:100px;
  height:100px;
  color:blue;
  transition: transform 0.4s;
}

.square:hover {
  tranform: rotate(360deg);
}
```

Tranforming Scale

- Scale is used to stretch an element based on a value multiplier.

If only 1 value is provided, it will scale the element in both directions by that value:

```
tranform: scale(2);
```



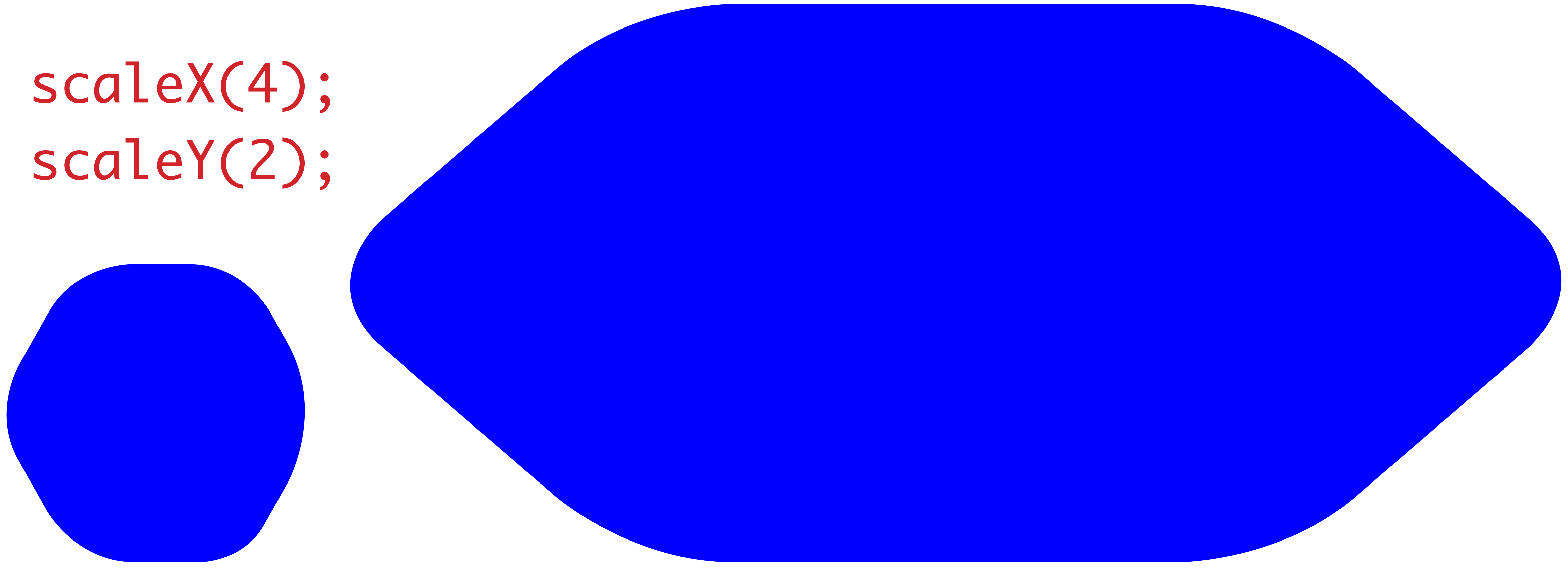
Tranforming Scale

If only 2 value are provided, it will scale the element in the horizontal direction first, then the vertical direction.

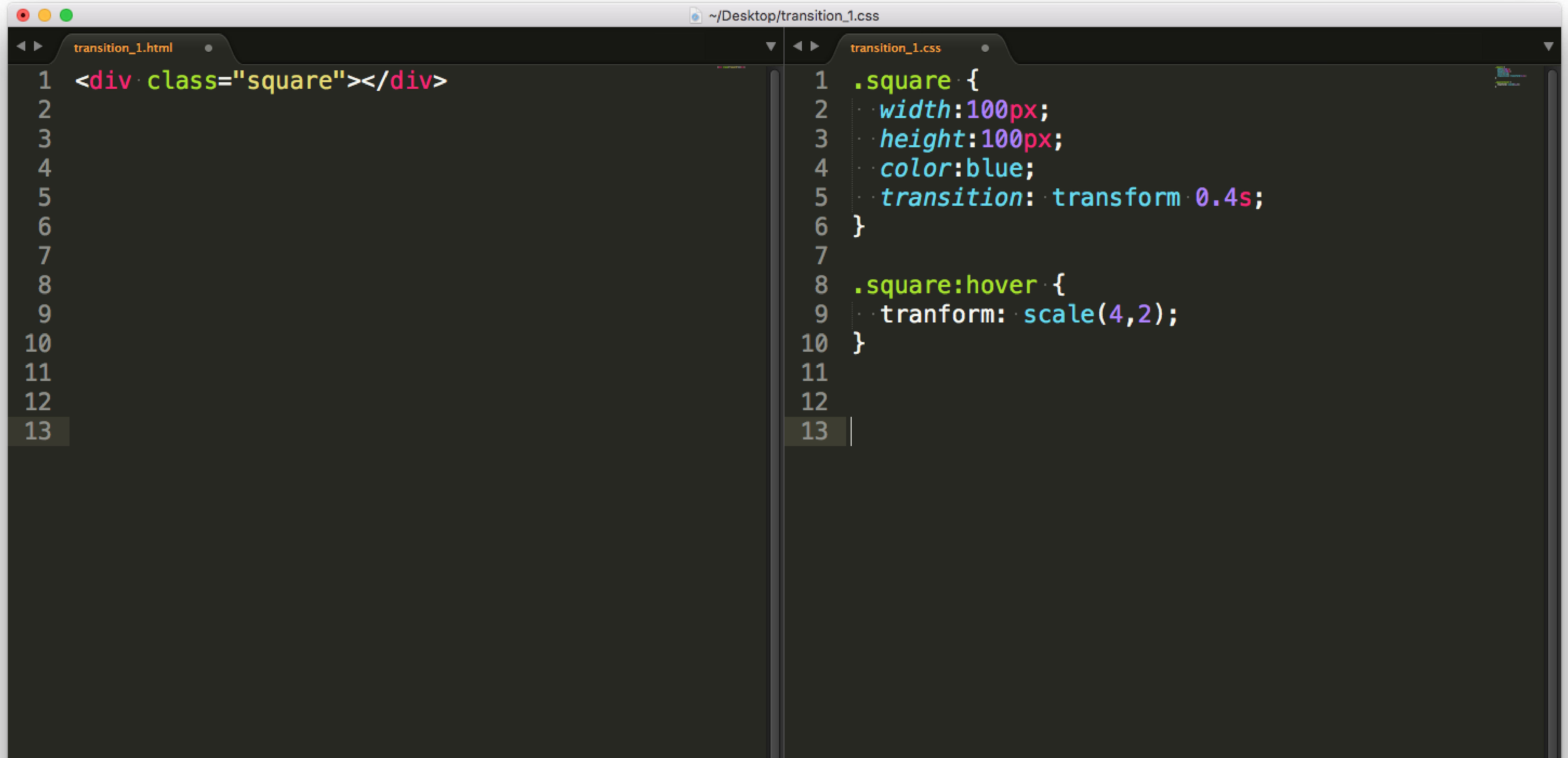
```
tranform: scale(4,2);
```

or

```
tranform: scaleX(4);
tranform: scaleY(2);
```



Tranforming Scale



~/Desktop/transition_1.css

transition_1.html

```html
<div class="square"></div>
```

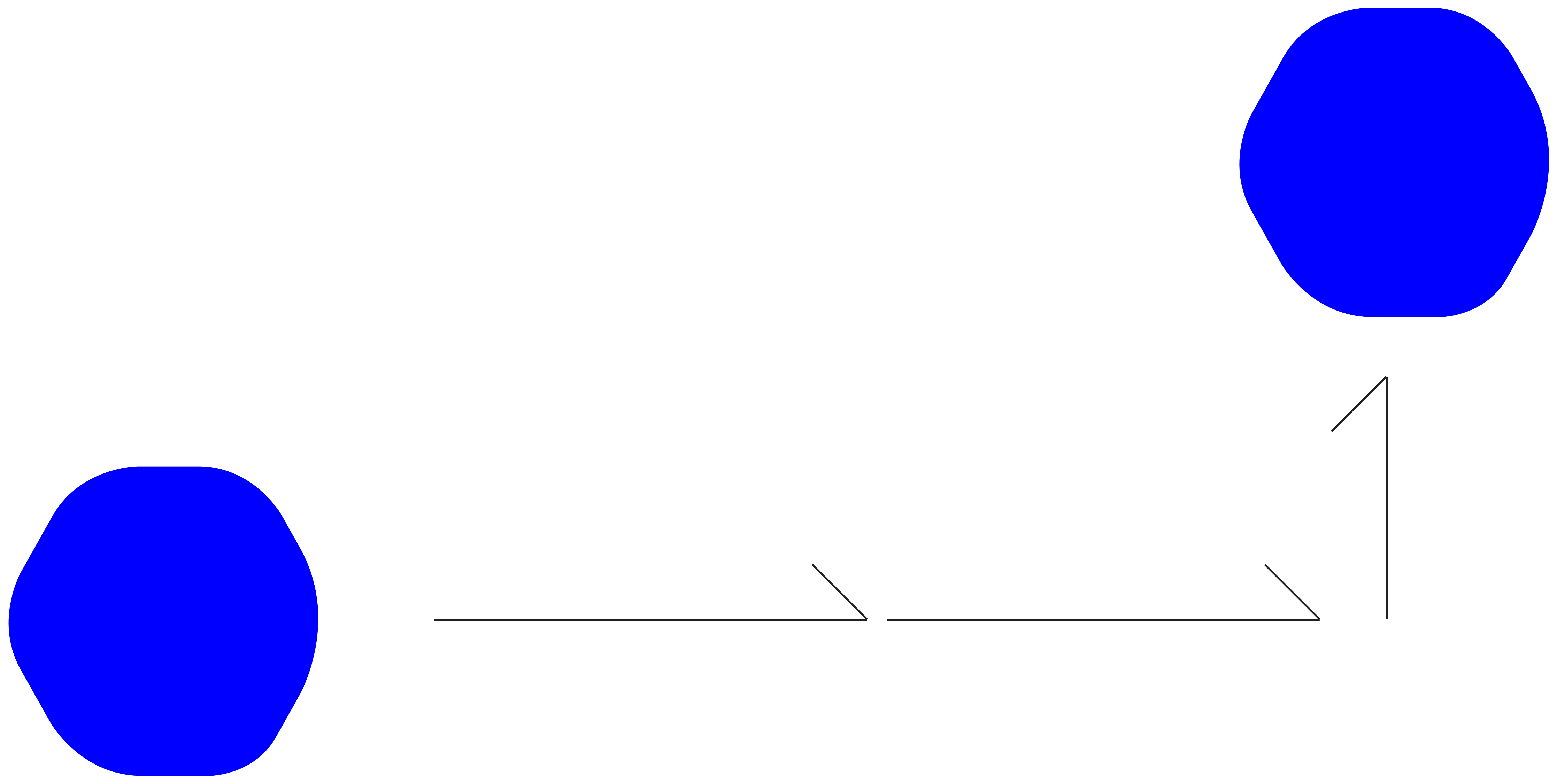
transition_1.css

```css
.square {
  width:100px;
  height:100px;
  color:blue;
  transition: transform 0.4s;
}

.square:hover {
  tranform: scale(4,2);
}
```

Tranforming Translate

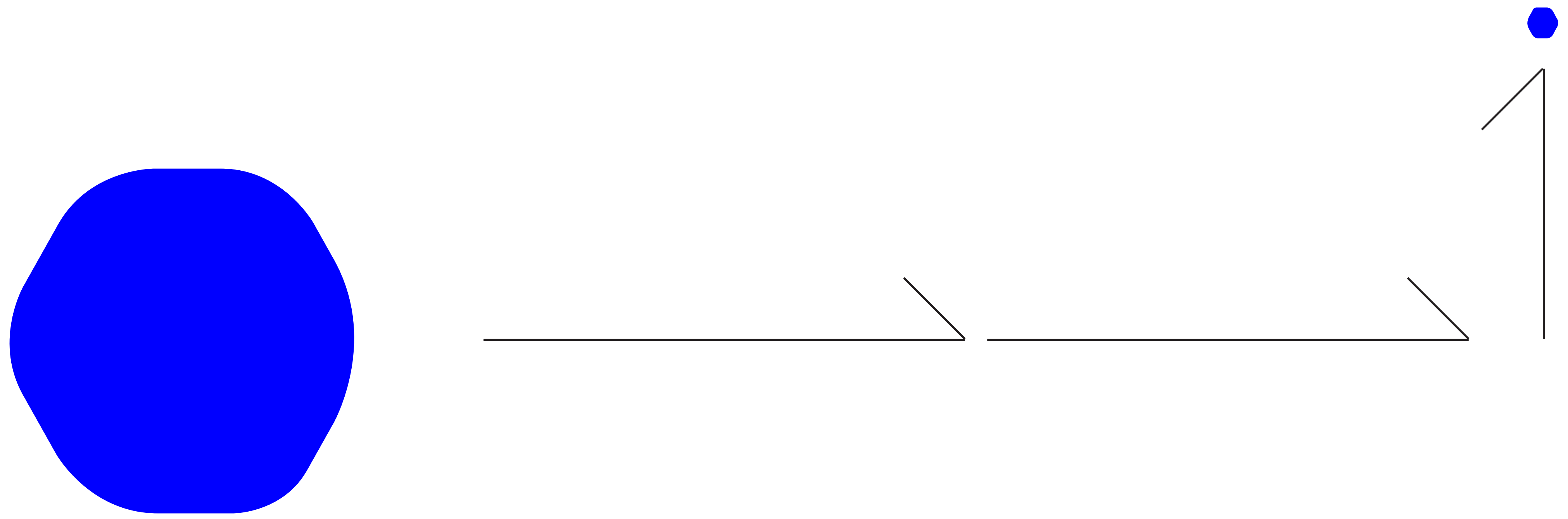
Translate simply means to move something.

```
tranform: translateX(500px);
tranform: translateY(30px);
```



Chaining

You can of course do many of these changes at once. For instance, you could scale down and translate to a new position.



Next time

Keyframe animations
SVG animations