In this lesson, we're going to take a look at the `transition` property, and work through some examples of how **transitions** can be applied to our designs.

So what are transitions? In CSS terms, a `transition` is a property that tells the browser how to change an object from one state to another. This means changing from Style A to Style B. The transition isn't just one directional though, it can apply whether we're going from A to B, or back again. Situations where you'll find yourself using transitions tend to be hover effects, as well as any where there's a change in state, such as when a form field becomes active or focused.

Transitions also work well when an element style changes. This might, for example, occur when you change an element's class using JavaScript. In this case, the way we define the transition property tells the browser how to transition between the two states.

The `transition` property is made up of four parts.

```
transition: transform 2s ease-out 1s;
```

What this transition means is "when any animatable property changes, wait for one second, then transition to the new style over a duration of 2 seconds". We'll go into timing functions in greater depth later.

The `ease-out` part of the definition, in this case, means to start fast and then gradually decelerate toward the end of the 2 seconds.

I've written this example using shorthand which is a popular way to write these kinds of properties. But you can set out each of the properties separately if you wish. Let's step through each:

- First, we have the `transition-property`. This defaults to "all", but you can use this property to apply the transition only to specific properties. If, say, you wanted to transition an element's opacity, but not height or background color, you could specify "opacity" as the transition property. This can be useful when using multiple transitions, but we'll get to that later.
- Next we have the `transition-duration`. This is the amount of time the transition takes. By default this is zero. But in this case 2 seconds, as shown by the "s". This can also be milliseconds, which will be written as "ms".
- Next we have the `transition-delay`. This is how long a browser should wait before it applies the transition. Like the duration, we can specify this in seconds or milliseconds.
- Last we have the `transition-timing` function. In the timing section we'll go through each of the available options here. But this is where we describe the shape of the animation. By shape, I mean whether the animation proceeds at a steady pace from start to finish, or starts fast and slows down toward the end, or vice versa. In this case, we're using "ease-out", which starts fast and then slows toward the end of the duration.

That's the theory behind the transition property. Let's take a look at an example.