



They say **timing** is everything. When using animations, we frequently want to control the speed and the style of the animation. That is to say, the rate at which elements transition, or tween, between states. Timing functions control the way these tweens are calculated, and so give us the opportunity to finely control the way our animations look.

Browsers give us a bunch of preset timing functions we can use. Let's run through each of them, and see what they do before we look at some examples:

- First, we have "ease". Ease is the default, and it makes animations begin a little slowly, speed up, and then decelerate toward the end. It's a little like ease-in-out, except the ease-in part at the beginning of the movement is less pronounced.
- Next, we have "ease-in" and "ease-out". Ease-in is when an animation begins slowly and speeds up toward the end. Ease-out is the opposite; it starts fast and slows down toward the end. Combining these together, we have "ease-in-out". This both starts and ends slowly.
- Then we have "linear". This is the most computery of the animations. It moves from the beginning to the end in a robotic, unchanging pace.
- Then we have "steps". Steps is a function that breaks the animation into a series of discrete stages, with no tweening between each of the steps. Steps timing function lets you divide a transition into as many individual steps as needed. It's useful for stop-motion animations, like a cartoon, by being made up of individual frames put together to create the impression of movement. Twitter famously used this for the fave animation, in which they animated an image sprite made up of individual frames.
- Lastly, there's my favorite, cubic-bezier. Most of the timing-function presets are themselves cubic-bezier curves. A cubic-bezier curve is a curve made by defining two points. The curve represents the rate of change from the beginning to the end of the animation. We'll talk about this more, as there are tools we can use to create cubic-bezier curves to save doing the math by hand.

Lets see what these timing functions look like in code.