A previous challenge discussed the ease-outkeyword that describes an animation change that speeds up first and then slows down at the end of the animation. On the right, the difference between the ease-outkeyword (for the blue element) and linearkeyword (for the red element) is demonstrated. Similar animation progressions to the ease-outkeyword can be achieved by using a custom cubic Bezier curve function.

In general, changing the p1and p2anchor points drives the creation of different Bezier curves, which controls how the animation progresses through time. Here's an example of a Bezier curve using values to mimic the ease-out style:

animation-timing-function: cubic-bezier(0, 0, 0.58, 1);

Remember that all cubic-bezierfunctions start with p0at (0, 0) and end with p3at (1, 1). In this example, the curve moves faster through the Y-axis (starts at 0, goes to p1y value of 0, then goes to p2y value of 1) than it moves through the X-axis (0 to start, then 0 for p1, up to 0.58 for p2). As a result, the change in the animated element progresses faster than the time of the animation for that segment. Towards the end of the curve, the relationship between the change in x and y values reverses - the y value moves from 1 to 1 (no change), and the x values move from 0.58 to 1, making the animation changes progress slower compared to the animation duration.

To see the effect of this Bezier curve in action, change the animation-timing-functionof the element with id of redto a cubic-bezierfunction with x1, y1, x2, y2 values set respectively to 0, 0, 0.58, 1. This will make both elements progress through the animation similarly.