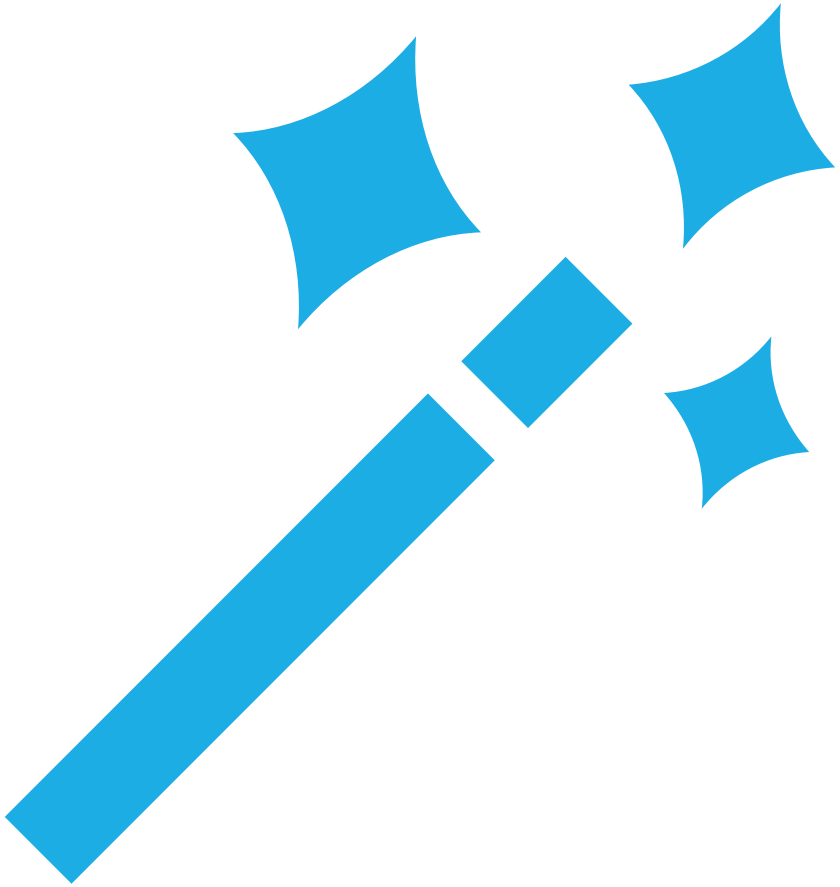
CSS ANIMATION



ANIMATION

Before we move on to the specifics of CSS animation, we need to understand the basics of animation in general.

To create an animation we need **at least** 3 things:

1. A **duration** in time the animation should last. This could be 1 second, 1 minute, 250 milliseconds and so on.
2. A **property** and value we want to change on an item on our page. This is usually the size, shape, or color of something on the page.
3. A **curve** that describes how the change should occur over our duration.

1 & 2 are pretty straightforward. You need a length of time and something you want to change. 3 is a bit more interesting a deserves some more explanation.

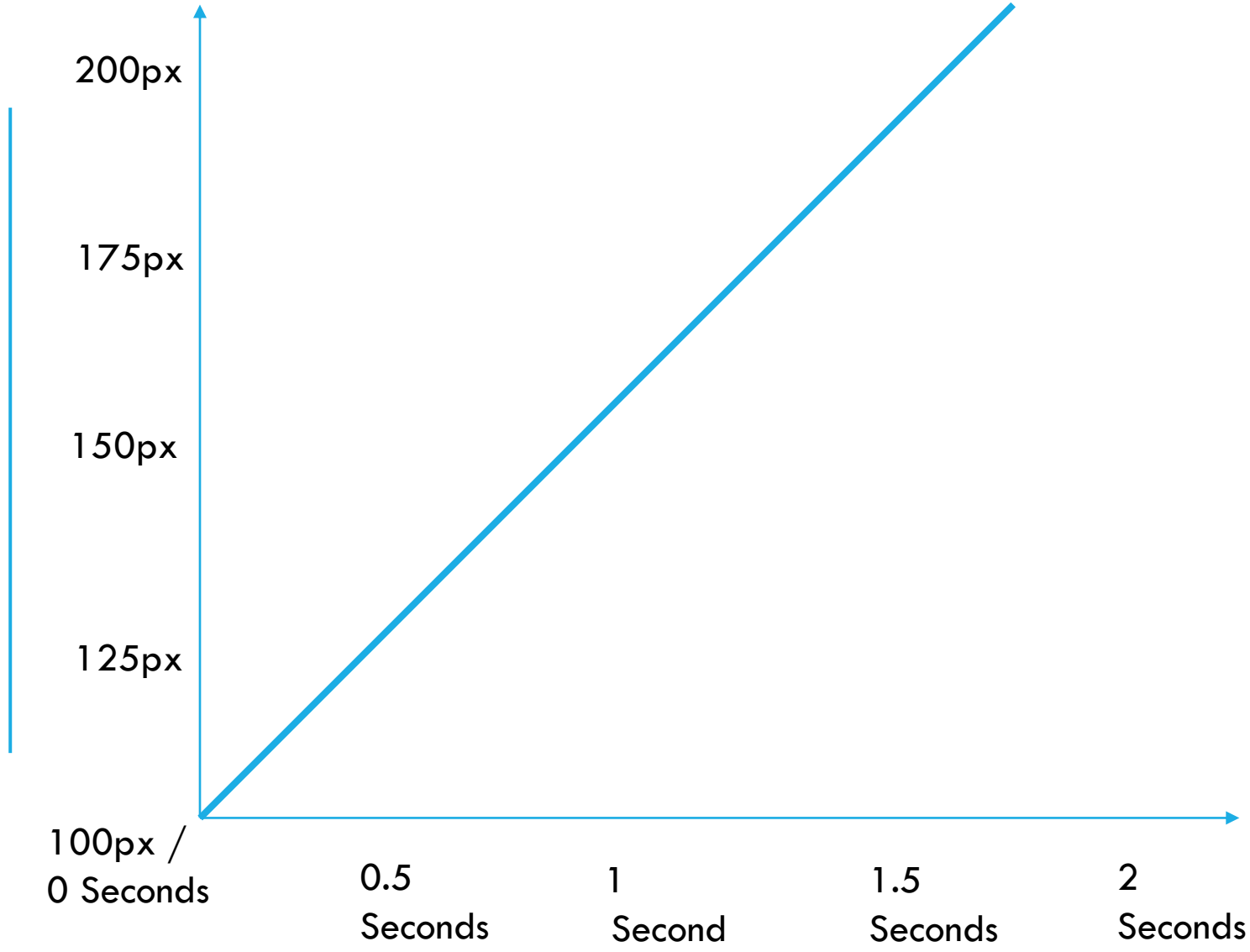
ANIMATION CURVES

Let's imagine we want to animate the **height** of an object from 100px to 200px over a duration of **2 seconds.** With just that information, we **don't** have a fully described animation.

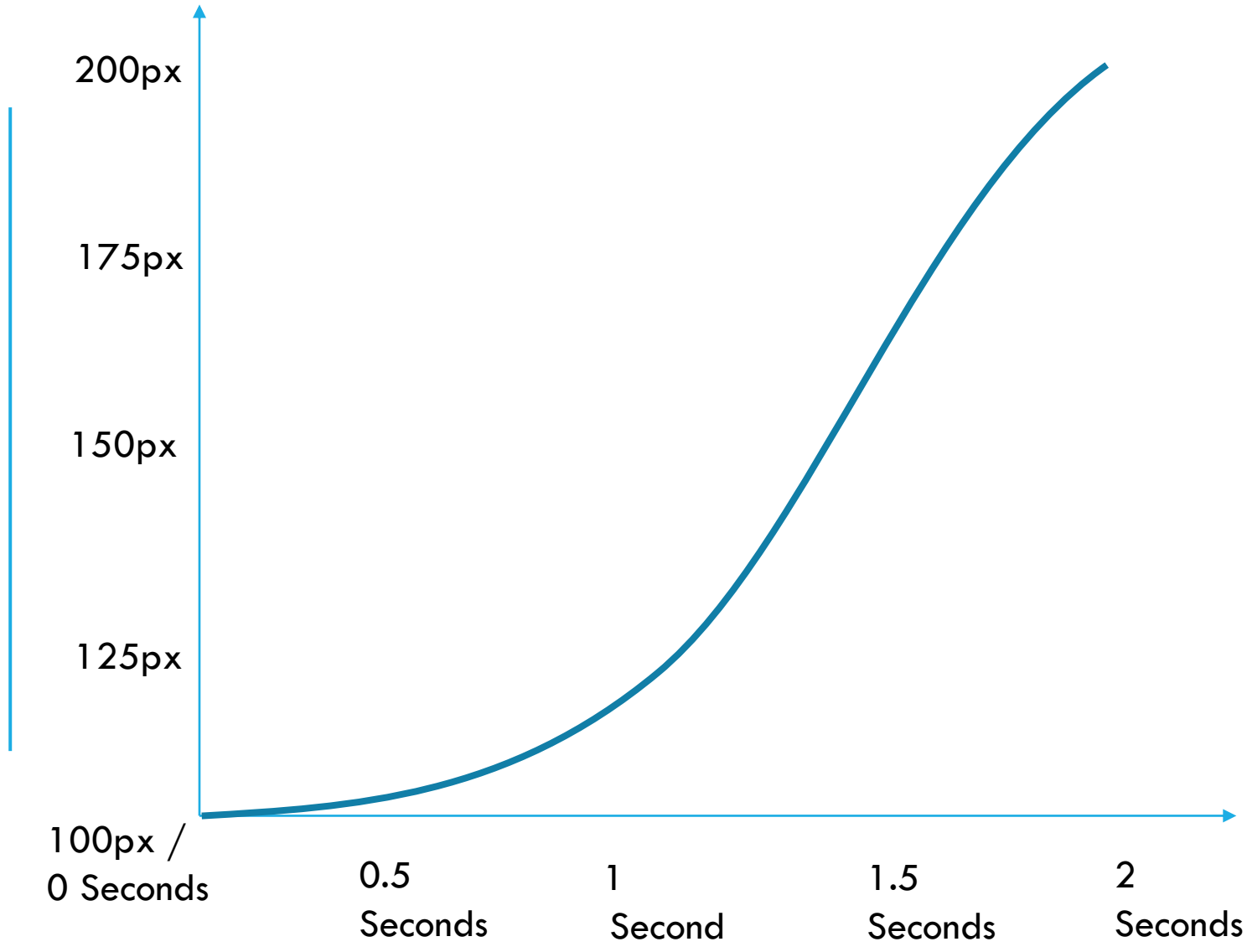
How should the animation execute? Should it evenly grow the height over the 2 seconds, should it start by growing really fast for 1 second and then slower for the last second? This is the problem a **curve** addresses. It describes how our changes should take place over the given duration.

Let's take a look at some different curves for our animation.

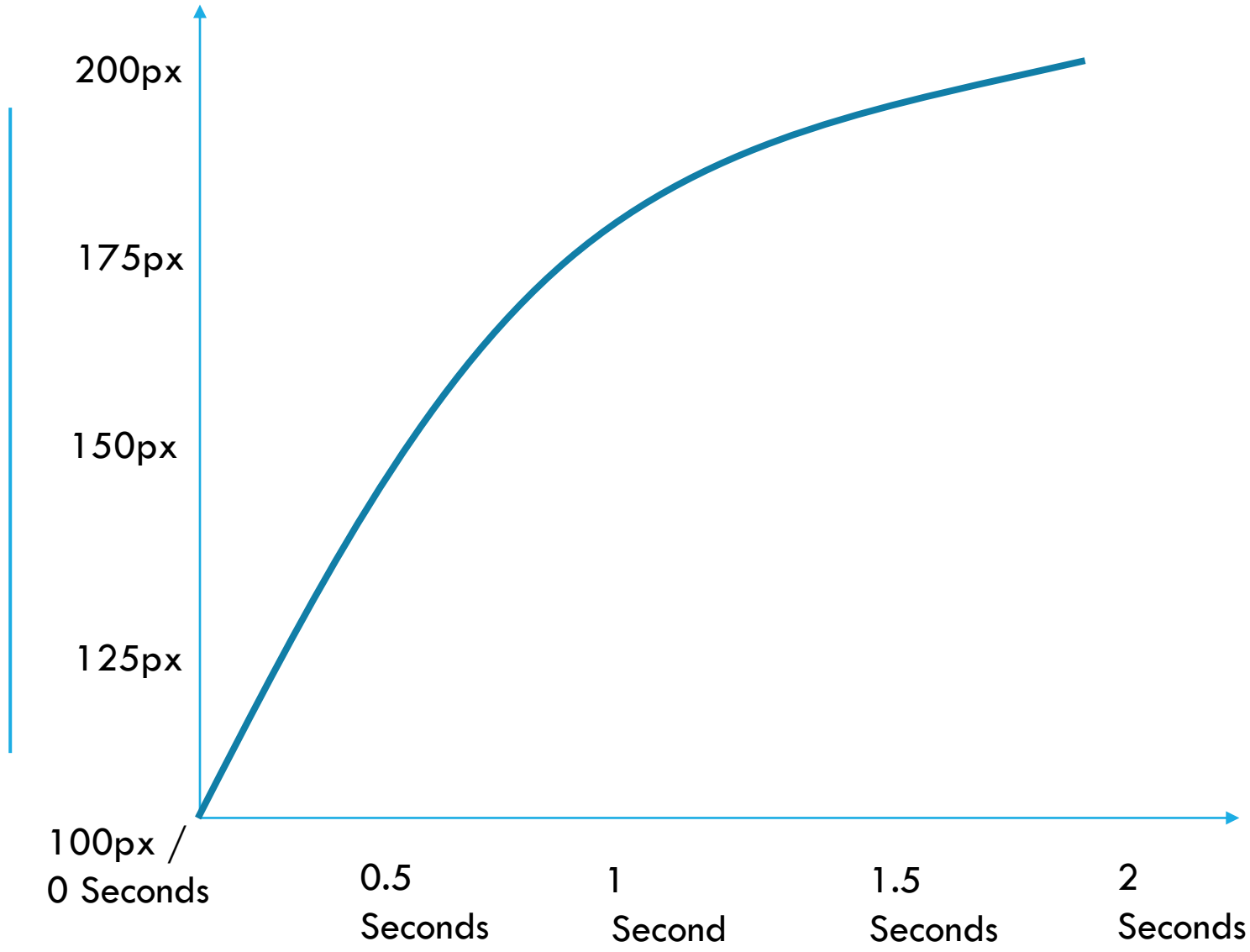
LINEAR


200px
175px
150px
125px
100px /
0 Seconds
0.5
Seconds
1
Second
1.5
Seconds
2
Seconds

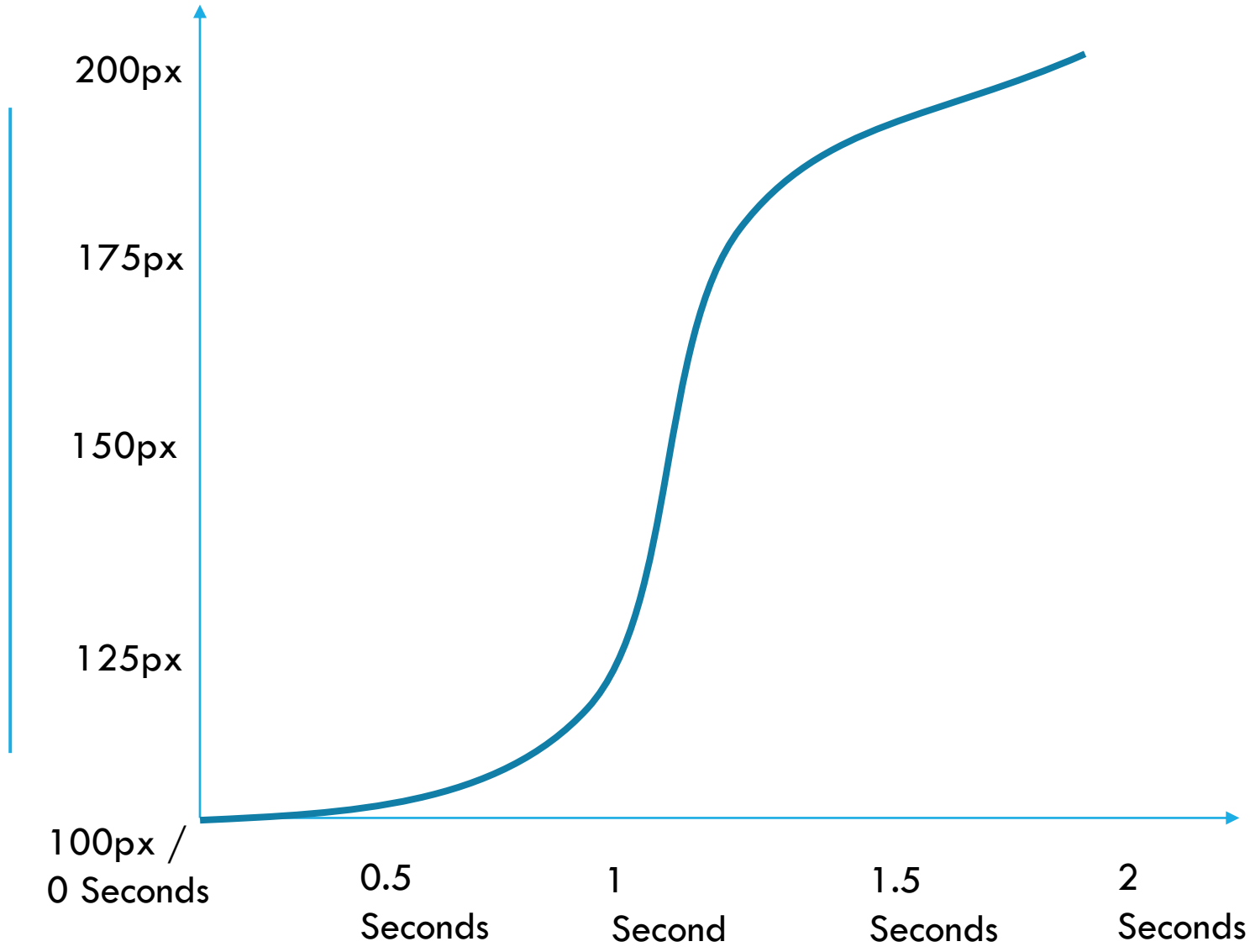
EASE IN


200px
175px
150px
125px
100px /
0 Seconds
0.5
Seconds
1
Second
1.5
Seconds
2
Seconds

EASE OUT


200px
175px
150px
125px
100px /
0 Seconds
0.5
Seconds
1
Second
1.5
Seconds
2
Seconds

EASE IN OUT


200px
175px
150px
125px
100px /
0 Seconds
0.5
Seconds
1
Second
1.5
Seconds
2
Seconds

ANIMATION CURVES

You can even make your own custom curves! We will see how in the upcoming slides. You can make the animation **bounce** beyond the starting and ending points to give even more life to our animations.

Now that we understand the very basics of animation, we can move on to seeing how they are implemented in CSS.

CSS ANIMATION

There are two ways we can add animations to our site in CSS:

1. Transitions
2. Key-Frame animation

Transitions are a great introduction to getting some movement on the page and key-frame animation can give us much more fine detail control over our animations with customization options.

TRANSITIONS

To trigger a CSS transition, a **state change** must occur. For example, the hover or active state being triggered.

To trigger our transition we need:

- A CSS rule defined in the standard state (starting point)
- A transition defined in the standard state (the animation)
- A CSS rule defined for a changed state using pseudo selectors (ending point)

TRANSITIONS

A very simple example:

```
#my-div {
    width: 10vw;
    height: 10vh;
    background-color: blue;
    transition-property: width;
    transition-duration: 1s;
    transition-timing-function: ease-in-out;
}

#my-div:hover {
    width: 90vw;
}
```

TRANSITION EXTRAS

We can also add in a **transition-delay** to trigger the animation after a certain amount of time. This is a great way to **chain** different animations one after the other.

As with many properties of CSS, there is a short-hand way to write our transitions:

```
transition: width 1s ease-in-out;
```

This will do the exact same thing as our example. We give the property, duration, and curve. Optionally you can give a delay duration as a 4th argument.

```
transition: width 1s ease-in-out 0.5s;
```

You can also set a transition for multiple properties you want to change using the **all** keyword:

```
transition: all 1s ease-in-out;
```

KNOWLEDGE CHECK

Let's see if we can get some simple transitions to add life to our page:

1. Create a folder called TransitionAnimation in your Scratch directory
2. Connect this folder to Git/GitHub
3. Add in at least 3 content tags
4. Give these content tags the following in CSS (you choose the values):
    1. width
    2. background-color
5. Create a transition that changes both of these properties over 1 second when **hovered** using an ease-in-out curve
6. Add commit and push your code

KEY-FRAME ANIMATION

Transitions are great and can create some pretty powerful animation effects, but there are some instances that you will need more control over your animation.

To create a **keyframe animation** we need two things:

1. Our animation definition
2. Our animation rules.

KEY-FRAME ANIMATION DEFINITION

Our animation definition allows us to define the properties we would like to animate. This is very similar to the **transition-property** from before but much more powerful.

To create an animation definition we use the following syntax:

```
@keyframes myAnimationName {
    0% {
        width: 100px;
    }

    100% {
        width: 200px;
    }
}
```

KEY-FRAME ANIMATION DEFINITION

A more complicated example:

```
@keyframes myAnimationName {
    0% {
        width: 100px;
        background-color: red;
    }
    50% {
        width: 400px;
        background-color: green;
    }
    100% {
        width: 200px;
        background-color: blue;
    }
}
```

KEY-FRAME ANIMATION

So with our previous example, we have the CSS properties we want to animate, but we have yet to:

1. Attach it to an actual item on our page
2. Define a duration
3. Define a curve

As we remember, these are things we need to do if we want to have an animation take place on our page. For key frame animation, we place these in **the CSS rule for the object we want to animate**.

KEY-FRAME ANIMATION

This is shown best with an example. Let's **attach** the animation we created to a div on our page.

```
#my-div:hover {
    animation-name: myAnimationName;
    animation-duration: 2s;
    animation-timing-function: ease-in-out;
}
```

KEY-FRAME ANIMATION EXTRAS

There are many more properties we can set up for our key-frame animation:

```
#my-div:hover {
    animation-name: myAnimationName;
    animation-duration: 2s;
    animation-timing-function: ease-in-out;
    animation-delay: 1s;
     animation-iteration-count: 4;
}
```

KNOWLEDGE CHECK

Let's see if we can get some simple animations to add more life to our page:

1. Add in 1 more content tag
2. Give the content tag the following in CSS (you choose the values):
    1. background-color
    2. font-size
3. Create an animation that changes both of these properties over 1 second when **hovered** using an ease-in-out curve
    1. Have the animation change the background-color and font-size at least 3 times over the life of the animation
4. Add commit and push your code

ANIMATION

And that's it! I will warn you, please do not overuse animation. It is something that is very eye catching which means when it is used too much, your users can get a little bit overwhelmed.