## Overview

CSS animations consist of two components: a style describing the CSS animation and a set of keyframes that indicate the start, intermediate, and end states of the animation's style.

Keyframe animations are frequently used to build an engaging interface that delights users.

In this activity, you’ll build some common animations that you’ve probably seen across the internet.

## Instructions

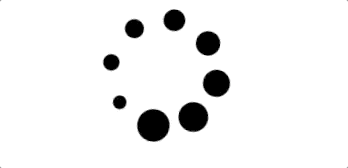
1. Open index.html located in Activities / 03\_keyframes and index.css located in Activities / 03\_keyframes / css.

**Animation 1: Bouncing Arrow**

****

1. Add the property animation: bounceArrow .7s infinite to the .bounceMe selector.
   * Note the name of the animation must be the same as the keyframe specified later in the file.
   * There are three things we just specified. A name of bounceArrow, a duration of .7s and infinite repetition.
2. Now it’s time to modify the keyframes. Let’s start with the start state.
   * Add the property margin-top: 0px; to the start state for bounceArrow.
   * The bounceArrow animation will now start with a 0px top margin.
3. Add margin-top: 7px; to the intermediary keyframe.
   * This indicates our bounceArrow animation will have a top margin of 7px at its halfway point.
4. Lastly, add a property to the end state so our animation returns to its original starting position.

**Animation 2: spinMe**

****

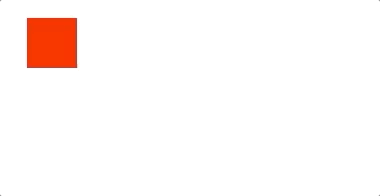
1. Next, you will animate a loading circle. Add the property animation: spin 2s infinite; to the .spinMe selector.
   * What are the three things you specified in this property?
2. Modify the start state by adding the property transform: rotate(180deg);.
3. How can you bring the animation full circle?
   * Add the property transform: rotate(-180deg); to the end state.
4. Take a minute to inspect these two states and see how we only used two states to make an animation.

**Animation 3: Loading Bar**

****

1. Next, you will animate a loading bar. Add the properties animation: loadBar 4s; and animation-fill-mode: forwards; to the #progressBar selector.
   * The first property names and adds a duration to our animation.
   * The second states the direction of our animation.
2. Modify the start state by adding the property width: 40%;.
3. Finish the loading bar by adding the property width: 100%; to the end state.

**Animation 4: Move a box!**

****

1. Animate the red box by adding the values to the .movingBox selector that declares the name of the keyframes (movingBox) and duration (5s).
2. Modify the start state by adding the property transform: translate3d(0px, 0px, 0px);.
   * There’s a lot happening here, so let’s review the syntax.
   * The first value is the X-axis, the second value is the Y-axis and the third value is the Z-axis. We’re not transforming along the Z-axis in this case, but we still have to declare it.
   * You can read more about translate3d [here](https://developer.mozilla.org/en-US/docs/Web/CSS/transform-function/translate3d).
3. Add the following properties to the remaining keyframes in the same order.
   * transform: translate3d(100px, 0px, 0px);
   * translate3d(50px, 300px, 0px);
   * translate3d(200px, 100px, 0px);
   * translate3d(0px, 0px, 0px);

**Animation 5: Bonus!**

1. It's time to be creative. If you're feeling adventurous and think you have a solid understanding of keyframe animations, you should practice!
2. Be creative and test yourself!
3. Try animations with images or try to recreate an animation you’ve seen somewhere.