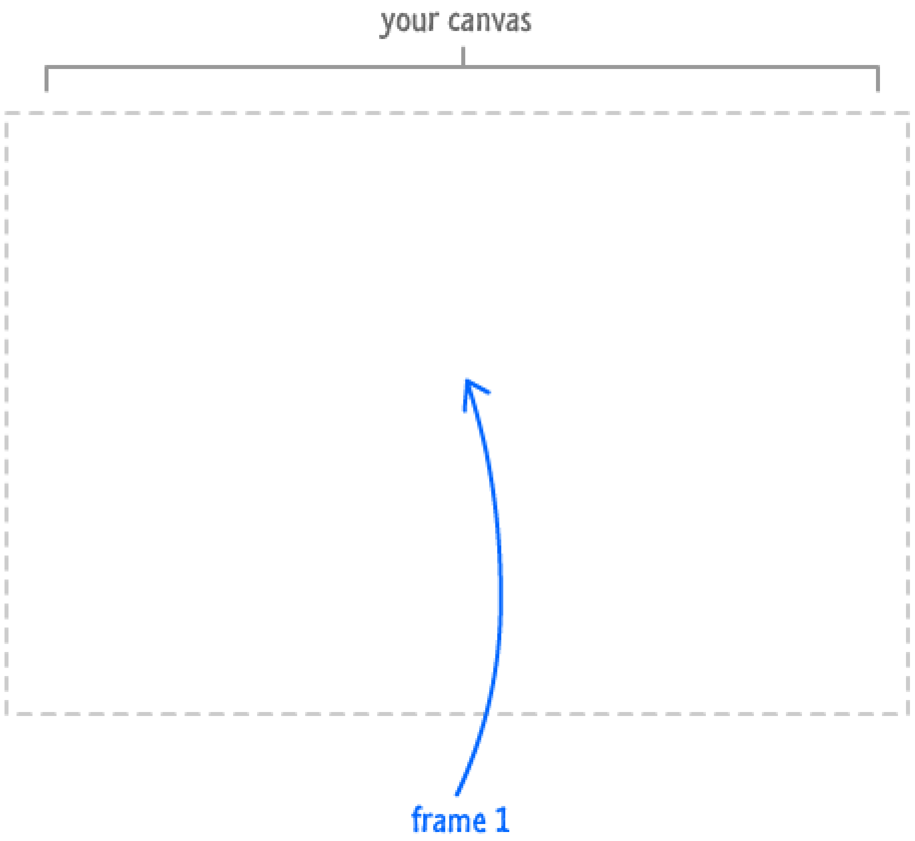
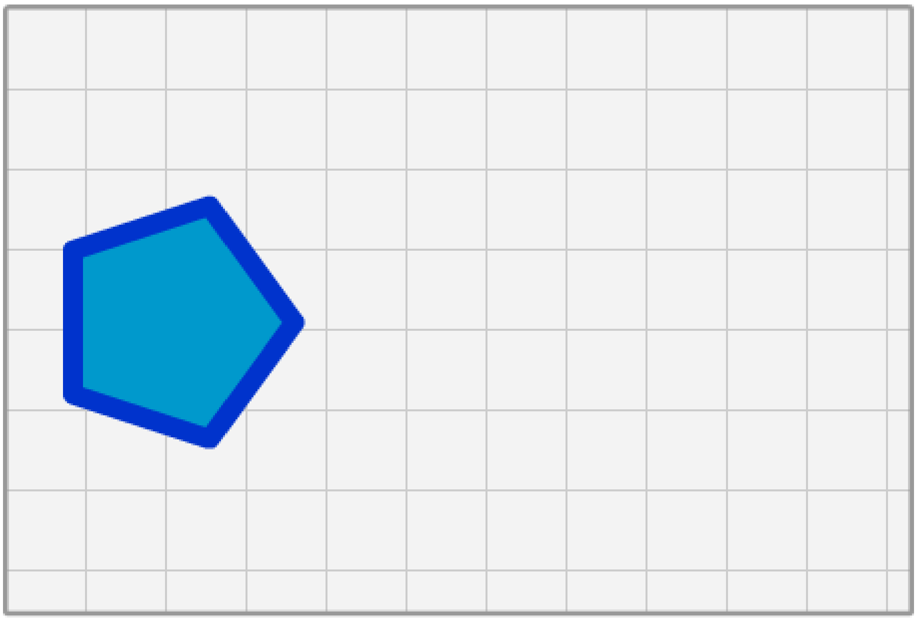
Animating on the Canvas

In the previous sections, we looked at some basic things about animations and how to think about them. It's time to move beyond that and focus a bit more on the details. The way you draw and animate on a `canvas` is very similar to how people created animations manually back in the day.

Initially, your `canvas` is completely blank. Let's call this initial state **frame 1**:



In this blank frame, you draw what you want to show as part of your starting state:



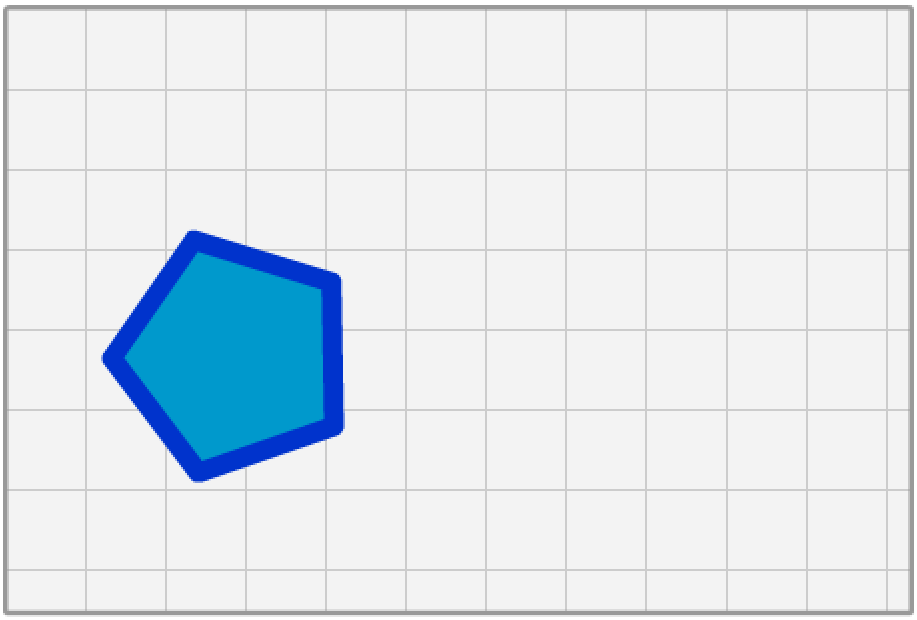
frame 1

You draw everything from the things in the foreground to things that appear in the middle to the things that make up the background. As you know well by now, every single detail is under your control. The `canvas` does nothing for you. It’s really friendly like that.

Once you are happy with how the first frame looks like, you **clear** everything that is shown. What you have now is a new frame:

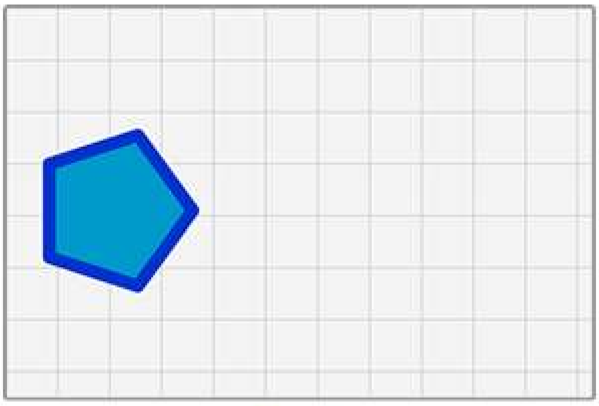
frame 2

In this new frame, which we will affectionately call **frame 2**, you re-draw everything you had in frame 1 but alter what you drew ever-so-slightly:

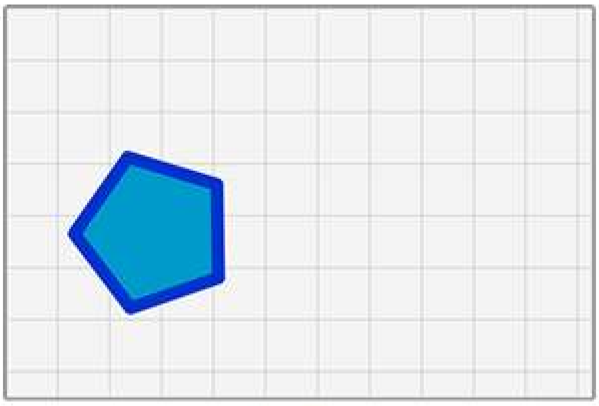


frame 2

In this example, between frame 1 and frame 2, our pentagon shape has rotated slightly and moved over and down a little bit. Everything else is pretty much the same. Now, if you compare frame 1 and frame 2 side by side, you'll be better able to see these two subtle changes:

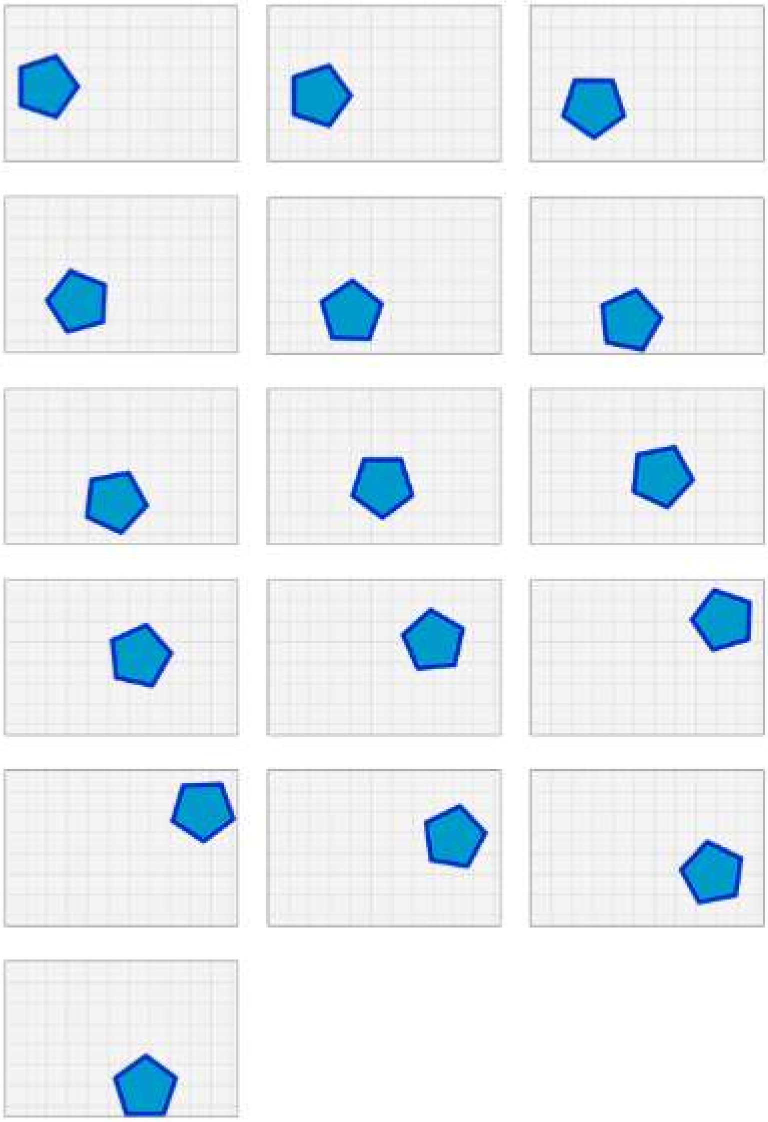


frame 1



frame 2

What you have just defined is an intermediate state. To create a smooth animation, you will need to define many MANY intermediate states. This means that you will need to repeat this **draw** and **clear** process over and over again for frame 3, frame 4, frame 5, and so on. In each frame, your pentagon shape is altered very slightly from the frame that preceded it. The following diagram shows the sequence of frames for this animation:



You will keep creating these intermediate frames until you hit your end state - which also signifies the end of this particular animation sequence. When all of these frames are played back-to-back really quickly, you have an animation.

In the next few sections, we'll go even deeper and turn everything you've learned in this section into working code.