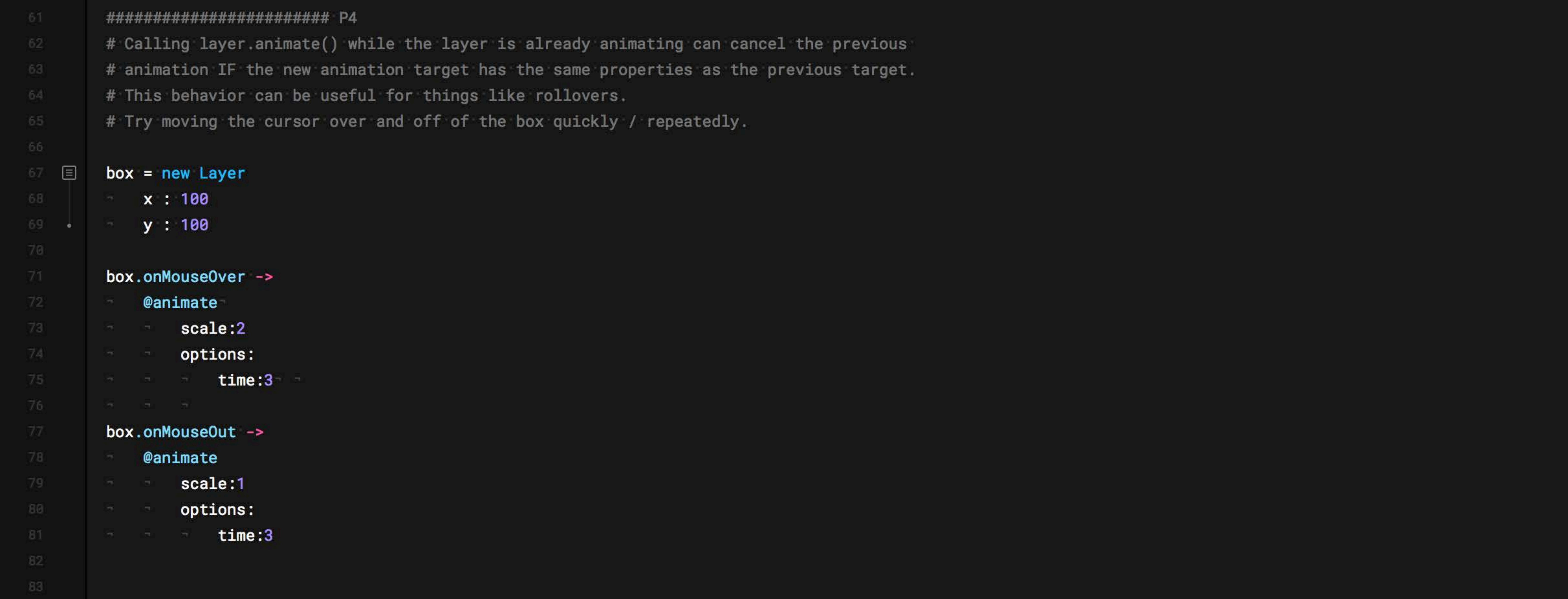
```coffeescript
###

	FramerJS introduction:

		• Basic Triggered Animation

###

######################## P1
# Any UI development framework should include a way to animate things.
# Framer is no exception.
# Framer layers have an animation method.
# You call the method and pass it an object with details about the animation.
# Framer will change the layer's properties, over time, to match the details object.
# The details object is a kind of 'animation target' or 'keyframe data'.
# You can think of this approach as 'fire and forget'. (This is common in many animation systems.)


box = new Layer
box.animate({x:300,y:300})


######################## P1 (alternate)
# Same as above, but using shortcuts.
# This is super common and we'll use this approach hereafter.

box = new Layer
box.animate
	x:300
	y:300


```

```
######################## P2
# The details object can include an options property that contains more animation details.
# Two common options are time and delay.
# Time is the duration of the animation and delay is, well, a delay.

box = new Layer

box.animate
	scale:1.2
	rotation:45			# Rotation is in degrees. Note that layers rotate around their center by default.
	backgroundColor:"#ff0000"
	options:
		time:0.5		# The animation will last for 1/2 of a second.
		delay:2			# The animation will start in 2 seconds.

# There are more options in the documentation. (cmd+D)


######################## P3
# Animations are commonly used inside of event handler functions.

box = new Layer

box.onMouseDown ->
	@animate						# Remember, '@' means 'this.' and 'this' refers to the layer instance that generated the event.
		rotation:180 + @rotation		# When the handler function runs, '@animate' translates to 'box.animate'
```



```
######################## P4
# Calling layer.animate() while the layer is already animating can cancel the previous
# animation IF the new animation target has the same properties as the previous target.
# This behavior can be useful for things like rollovers.
# Try moving the cursor over and off of the box quickly / repeatedly.

box = new Layer
    x : 100
    y : 100

box.onMouseOver ->
    @animate
        scale:2
        options:
            time:3

box.onMouseOut ->
    @animate
        scale:1
        options:
            time:3

```

```
######################### P5
# Any animation system must "interpolate" between the starting value and the end values of an animation.
# Interpolation is when a set of values between two other values are calculated using an equation.
# Different equations can be used for interpolation.
# Different equations result in different sets of values that lead to different looks, or easing, to an animated change.
# Because of this, sometimes these equations are called "easing equations".
# The values of these equations, when graphed, will look like different kinds of curves:
# Because of this, "easing equations" are sometimes just called "curves".
# In animation software like After effects you adjust these curves by hand.
# In UI development frameworks you select from a set of pre-made curves. (Sometimes you can insert math for your own curves.)
# Framer's options include: "ease-in", "ease-out", "ease-in-out".
# There are more options in the documentation.

box1 = new Layer
box2 = new Layer
    y:200

box3 = new Layer
    y:400


box1.onMouseDown ->         # I put these animations in the mouse down handler so you have control over triggering them.
    box1.animate
        x:400
        options:
            curve:"ease-in"

    box2.animate
        x:400
        options:
            curve:"ease-out"

    box3.animate
        x:400
        options:
            curve:"ease-in-out"
```