Animation
What does it mean to *animate* something?

- To bring to life
- To cause to appear as if it’s moving or changing

In the context of iOS applications, this means to modify aspects of the user interface in a special way as to produce the appearance of action.

Why would we want to animate something?

- It draws the user towards things that change
- It indicates importance at a particular moment
- It makes your app look cool, fun, or polished – which can be a differentiator
You can animate the following properties of a UIView derived object:

- **frame** – move or scale the view (relative to its superview)
- **bounds** – move the view’s contents within the view
- **center** – move the view relative to the screen
- **transform** - scale, rotate, or translate the view relative to its center point
- **alpha** - gradually change the transparency of the view
- **backgroundColor** - change the view’s background color
- **contentStretch** - change the way the view’s contents are stretched to fill the available space
The basic UIView animation method is `UIView.animate`: 

```swift
UIView.animate(
    withDuration: <duration>,
    delay: <delay>,
    options: <options>,
    animations: {
        <animation code>
    }
    completion: {
        <completion code>
    }
)
```
duration: how long in seconds to run the animation
delay: how long to wait until starting the animation
options:
  .curveEaseInOut  begin slow, accelerate, end slow
  .curveEaseIn    begin slow, accelerate to end
  .curveEaseOut   begin quickly, slow to end
  .curveLinear    even over the duration
  .repeat         make the animation loop forever
  .autoreverse    animate forward, then reverse

animation code:
  identifies the ending value for the selected attribute(s)

completion code:
  code to be executed at the end of the animation
Adjust the *alpha*:

```swift
// Starting alpha value
self.labelName.alpha = 1.0

UIView.animate(
    withDuration: 3.0, 
    animations: {
        self.labelName.alpha = 0.0
    }
)
```
Center Animation: Slide Out to Right

Adjust the *center*:

```swift
// Starting center value
self.labelName.center.x = self.view.center.x

UIView.animate(
    withDuration: 3.0,
    animations: {
        self.labelName.center.x +=
            self.view.bounds.width
    }
)
```
Adjust the `transform`:

```swift
UIView.animate(
    withDuration: 3.0,
    animations: {
        // 180 degree rotation
        self.labelName.transform =
            self.labelName.transform.transform.rotated(
                by: CGFloat(Double.pi))
    }
)
```

Center Animation: Spinning