How to Make an App with Databases

Module 2
Lesson 4
Worksheet

Welcome!

In this worksheet, you'll practice writing some closures in Xcode with the help of autocomplete.

Step #1: We'll be doing these exercises in a new Xcode project.

Open Xcode and create a new project.
(File Menu->New->Project).
From the list of templates, select "**Single View Application**"

Step #2: We're going to add a label to the storyboard so we have something to see. Go to Main.storyboard and add a label to the view. Then do the following:

- Add two constraints so that the label is horizontally and vertically centered.
- Set the text to "Animating"
- Connect the label to the view controller as an IBOutlet property
- Set the alpha value of the label to 0

Step #3: Now go into ViewController.swift and we're going to create a timer object that will fade the label into view. In order to do this, we'll need to supply a closure that changes the alpha property of the label.

```
override func viewDidLoad() {
    super.viewDidLoad()

    var timer = Timer.scheduledTimer(withTimeInterval: 0.1,
        repeats: true, block: (Timer) -> Void)

}
```

Use **0.1** for the time interval and set the repeats parameter to **true**. When you get to the closure parameter, double click or press **enter**. It'll open up a closure for you and you'll get this:

```
override func viewDidLoad() {
    super.viewDidLoad()

    var timer = Timer.scheduledTimer(withTimeInterval: 0.1,
        repeats: true) { (Timer) in
        code
    }

}
```

Give the parameter a name such as "timer" or simply "t" and then for the code inside the closure, write this:

```
self.label.alpha += 0.1

if self.label.alpha >= 1 {
    t.invalidate()
}
```

Step #4: At this point, your viewDidLoad method should look like the image below. Notice that we had to use the **self** keyword to reference the label property.

```
override func viewDidLoad() {
    super.viewDidLoad()

    var timer = Timer.scheduledTimer(withTimeInterval: 0.1,
        repeats: true) { (t) in

        self.label.alpha += 0.1

        if self.label.alpha >= 1 {
            t.invalidate()
        }
    }

}
```

Run your project and you should see the label fade in!

Step #5: If you got that working, we're going to erase that timer statement and start all over; this time using UIView.animate() Go ahead and delete that code so all you have is an empty viewDidLoad method like this:

```
override func viewDidLoad() {
    super.viewDidLoad()

}
```

Step #6: Open a new UIView.animate method. Choose the one with these three parameters: a **time interval**, **animations** and **completion**.

```
override func viewDidLoad() {
    super.viewDidLoad()

    UIView.animate(withDuration: TimeInterval, animations: () -> Void,
        completion: ((Bool) -> Void)?)

}
```

Put **1** for the **time interval**. Then for the **animations** parameter, press “enter” or double click it to open up a closure. Inside the closure, type the following code:

```
self.label.alpha = 1
```

For the **completion** parameter, open up a closure, give the closure parameter a name such as “completed” and type this code into it:

```
self.label.text = "done"
```

Your completed method should look something like this:

```
override func viewDidLoad() {
    super.viewDidLoad()

    UIView.animate(withDuration: 1, animations: {

        self.label.alpha = 1

    }) { (completed) in

        self.label.text = "done"
    }

}
```

Run your project and you'll see your label fade in and then the text change to "done".

You're done! I hope doing these simple exercises gave you a little more confidence with working with closures.

If you have any questions, remember to join the Students Only forum and get help anytime!
https://facebook.com/groups/CodeWithChris

To get the solutions, visit the lesson page.