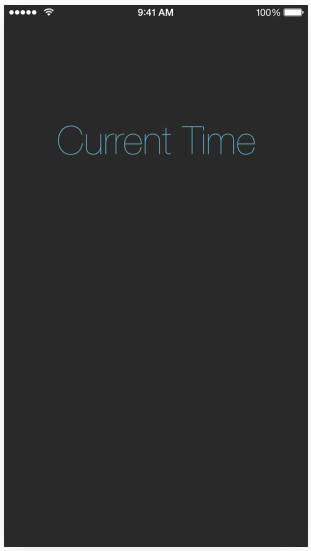
Clock
Lesson 1

Description

Add a `UILabel` to the screen and apply visual position and style. Declare the label as a controller's `@IBOutlet` property, and update the label's text programatically.



Learning Outcomes

- Apply Interface Builder to build a user interface.
- Use Auto Layout constraints to position a `UILabel`.
- Connect interface elements to controller code via an `IBOutlet`.
- Write and recognize the syntax of property declarations.
- Recognize controller lifecycle events and relate them to methods such as `viewDidLoad`.
- Apply `UILabel` properties and string literals.

Vocabulary

Interface Builder	Auto Layout	constraint
inspector	view	controller
Assistant Editor	outlet	property
property attribute	`@IBOutlet`	type annotation
connection well	Document Outline	life cycle event
`println`	Xcode console	string literal

Materials

- **Clock Lesson 1** Xcode project

Opening

How do we display a placeholder for text on the screen?

Agenda

- Using Interface Builder and the Object Library (⌥⌘L), drag a text label for displaying the current time onto the view.
- Use the Attributes Inspector (⌥⌘4) to apply styles, adjust position, and to set the initial text value to **00:00**.
- Add constraints by control-dragging upwards from the label to the view to center the label horizontally, and by using the Pin control (⊢□⊣) to set a constraint for the vertical position.
- Discuss how, if a constraint indicator is orange, Interface Builder is communicating that the visible position or size of the label does not match its constraints. With the label selected and the orange constraint indicators visible, correct the position and size of the label with the menu item *Editor* > *Resolve Auto Layout Issues* > *Update Frames* (⌥⌘=).
- Explain the relationship between views and controllers, and the concept of "outlets."
- Using Interface Builder and the Assistant Editor (⌥⌘↩), connect the label as an `@IBOutlet` property of the `ViewController` class.

```
@IBOutlet weak var timeLabel: UILabel!
```

- Explain the anatomy of a property declaration, emphasizing the `@IBOutlet` attribute, the property name, and type annotation.
- Experiment with removing the `@IBOutlet` keyword from the property declaration, and notice the connection well disappear. Restore the `@IBOutlet` keyword, and witness the connection well reappear.
- Demonstrate seeing the connection between the controller and view by using Interface Builder and the Assistant Editor (⌥⌘↩), to right-click (Control-click) the label itself, right-click the text label in the Document Outline, hover over the connection well with the mouse, and use the Connections Inspector (⌥⌘6).
- Explain the concept of application life cycle events and their relationship to controller methods, such as `viewDidLoad`.
- Experiment with generating an explicit console message during `viewDidLoad`.

```
override func viewDidLoad() {
   super.viewDidLoad()
   println("The view did load.")
}
```

- Display the Xcode console (⇧⌘C).
- Run the app (⌘R), and witness the `println` message on the console.
- Change the label text during `viewDidLoad`, to demonstrate the ability to manipulate a controller's `@IBOutlet` property with code.

```
override func viewDidLoad() {
   super.viewDidLoad()
   timeLabel.text = "Current Time"
}
```

- Explain the property access syntax and Swift string literals.
- Run the app (⌘R), and witness how the content of the text label has changed via controller code.

Closing

Now that we can display what we want with code, how do you think we can get the app to display the current time?

Modifications And Extensions

- Programmatically create a `UILabel` and set its position and text property.

Resources

Cocoa Core Competencies: Model-View-Controller https://developer.apple.com/library/ios/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html

Interface Builder Help: Creating and Connecting an Outlet https://developer.apple.com/library/ios/recipes/xcode_help-IB_connections/chapters/CreatingOutlet.html

Cocoa Application Competencies for iOS: Outlet https://developer.apple.com/library/ios/documentation/General/Conceptual/Devpedia-CocoaApp/Outlet.html

UIKit Constants Reference: Interface Builder Constants https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIKitConstantsReference/index.html#//apple_ref/doc/constant_group/Interface_Builder_Constants

The Swift Programming Language: Attributes https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/Attributes.html

UIViewController Class Reference https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIViewController_Class/index.html

View Controller Programming Guide for iOS https://developer.apple.com/library/ios/featuredarticles/ViewControllerPGforiPhoneOS/Introduction/Introduction.html

View Controller Programming Guide for iOS: Resource Management in View Controllers https://developer.apple.com/library/ios/featuredarticles/ViewControllerPGforiPhoneOS/ViewLoadingandUnloading/ViewLoadingandUnloading.html

The Swift Programming Language: Accessing Properties https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/ClassesAndStructures.html#//apple_ref/doc/uid/TP40014097-CH13-ID86

UILabel Class Reference https://developer.apple.com/library/ios/documentation/UIKit/Reference/UILabel_Class/