

WordCollage

Lesson 3

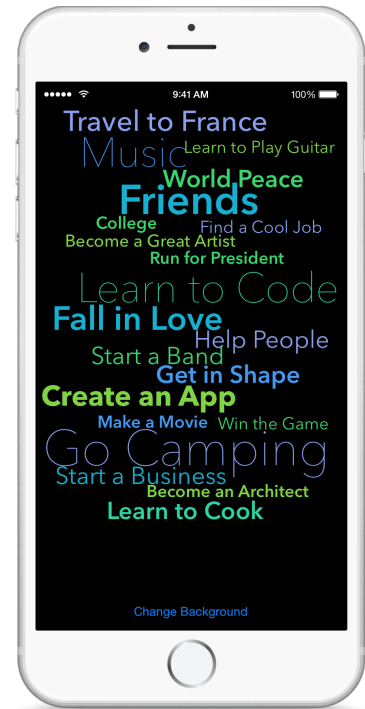


Description

Introduce Apple development tools, including Xcode, iOS Simulator, Swift and frameworks. Add a button that changes the background color.

Learning Outcomes

- Recognize the Xcode Identity, Attributes and Size inspectors.
- Identify the tools and technologies used to create iOS apps.
- Discover how to add user interface elements to an iOS app, and how to connect interface behavior to code.
- Practice adding layout constraints to user interface elements.
- Explain that Interface Builder relies on the @IBAction attribute to connect interface elements to code.
- Practice using the Xcode Documentation and API Reference.



Vocabulary

Xcode	IDE	project
source code	Interface Builder	storyboard
user interface	Xcode Documentation and API Reference	Model-View-Controller
view	controller	Assistant Editor
connection well	@IBAction	UIColor

Materials

- **WordCollage Lesson 3** Xcode project
- **Tools and Technologies** presentation

Opening

How do you create a button that changes the background color?

Agenda

- Open and run (⌘R) the **WorldCollage Lesson 3** app.
- Using Interface Builder, ensure that the Any Width | Any Height size class is active, and use the Object Library (⇧⌘L) to place a Button on the interface.
- With the button selected, briefly demonstrate the Identity (⇧⌘3), Attributes (⇧⌘4) and Size (⇧⌘5) Inspectors.
- Using Interface Builder, change the text of the button to "Change Background."
- Run the app (⌘R) and observe how the button appears in a different location within the iOS Simulator.
- Using Interface Builder, Control-drag from the Button downward to the View, and select Bottom Space to Bottom Layout Guide to create a Vertical Space constraint.
- With the Button still selected, use the Align control and select Horizontal Center in Container to create a Center X Alignment constraint.
- Run the app (⌘R), tap the button, and observe that nothing happens.
- Present the iOS tools and technologies, including Swift, Cocoa Touch / iOS SDK, Instruments, the iOS Simulator, LLVM/Clang Compiler, Xcode anatomy, Xcode shortcuts, the components of an Xcode project, storyboards and interface components, and MVC.
- While viewing the storyboard in Interface Builder, open the Assistant Editor (⇧⌘↔).
- Using the Show Document Outline control (⌘O) in the lower left corner of the canvas, ensure that the document outline is visible.
- Using the Document Outline, Control-click the button and drag a connection from the Touch Up Inside connection well to the controller, to create an Action connection. Use the name `changeBackgroundColor` and the Type `UIButton`.

```
@IBAction func changeBackgroundColor(sender: UIButton) {  
}
```

- Drawing attention to the connection well next to the method, explain the how Interface Builder relies on the `@IBAction` attribute to establish connections between interface components and controller code.
- Experiment with removing the `@IBAction` attribute, and witness the connection well disappear. Undo the change, and witness the connection well reappear
- Implement the `changeBackgroundColor:` method.

```
@IBAction func changeBackgroundColor(sender: UIButton) {  
    view.backgroundColor = UIColor.blackColor()  
}
```

- Using the Xcode Documentation and API Reference (⇧⌘0), demonstrate searching for `UIColor` to discover other "easy" colors.
- Run the app (⌘R), tap the button, and witness the background color change.

Closing

How might we add more buttons that change the background to different colors?

Modifications and Extensions

- Investigate the `UIColor` class, and use the `init(red:green:blue:alpha:)` initializer to create a specific color.
- Enhance the `changeBackgroundColor:` action so the background color toggles between light and dark colors.
- Experiment with simulating different devices in the iOS Simulator, and explore the different Size Classes within Interface Builder so the interface is usable on different devices.

Resources

iOS Developer Program <https://developer.apple.com/programs/ios/>

Start Developing iOS Apps Today <https://developer.apple.com/library/ios/referencelibrary/GettingStarted/RoadMapiOS/>

iOS Technology Overview <https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/>

iOS App Programming Guide: About iOS App Programming <https://developer.apple.com/library/ios/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/Introduction/Introduction.html>

Xcode Overview: Build a User Interface https://developer.apple.com/library/ios/documentation/ToolsLanguages/Conceptual/Xcode_Overview/dit_user_interface.html

Adding an Object to Your Interface https://developer.apple.com/library/ios/recipes/xcode_help-IB_objects_media/Chapters/AddingObject.html

Xcode Overview: Connect User Interface Objects to Code https://developer.apple.com/library/ios/documentation/ToolsLanguages/Conceptual/Xcode_Overview/edit_user_interface.html#/apple_ref/doc/uid/TP40010215-CH6-SW3

Cocoa Application Competencies for iOS: Target-Action <https://developer.apple.com/library/ios/documentation/General/Conceptual/Devpedia-CocoaApp/TargetAction.html>

Using Swift with Cocoa and Objective-C: Working with Outlets and Actions https://developer.apple.com/library/ios/documentation/Swift/Conceptual/BuildingCocoaApps/WritingSwiftClassesWithObjective-CBehavior.html#/apple_ref/doc/uid/TP40014216-CH5-XID_62

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

Start Developing iOS Apps Today: Finding Information <https://developer.apple.com/library/ios/referencelibrary/GettingStarted/RoadMapiOS/FindingInformation.html>