# Workshop: Swift for beginners

Ana Nogal
ana@codurance.com
@anainogal
@codurance
http://codurance.com

# Agenda

## What are we learning today?

Today we are going to write an iOS app called City Quiz. We will learn:

- the basics of Xcode 9
- the fundamentals of the Swift language

# Before we start

Make sure you have **Xcode** installed

# City Quiz

# Challenge

Create a TODO list of all the things you need to do to make this app work.

# City Quiz

# Solution

- Ask the user to find a city from a set of images
- Have 4 images of different cities to choose from
- Inform the user if their answer was correct
- Calculate the score
- Know what the current round is
- Be able to restart the quiz

# Demo

# View Controllers

Manage a Screen or a portion of a Screen


# View Controllers

Main.storyboard
(Design)

ViewController.swift
(Code)

```
@IBAction func showAlert() {

}
```

# Add City to the game

- Enter a City in the screen
  - Set the text to London
- Run the app
- Connect the city to the code file
- Change the text of the city label to Barcelona in code
- Run the app

# Walkthrough

- Go to Main.Storyboard
- Open the Utilities panel
- Select the “Label” control and drag the control to the ViewController
- In the attributes inspector, change the color of text to “Light Gray Color”
- Align text to center
- Change the Font of the Text to Custom, the style to Bold and size to 30
- Run the app

# Walkthrough

- Go to Main.Storyboard
- Open the Assistant Editor
- Holding the control key, Drag from the label to the code file
- Set the name to “cityLabel” and click “connect”
- Set the text of the label to “Barcelona” in the method viewDidLoad

# What have we learned so far?

- ViewControllers:
  - they control a screen
  - are made of 2 parts: the design and code
- Labels:
  - can change some attributes like color and size
  - can connect then to code

# Add a button to the game

- Add a Button
- Set the title to “Hit me!”
- Connect the Button to code by adding an action
- Run the app
- Tap the button

# Walkthrough

- Go to Main.Storyboard
- Create a button
- Open the assistant editor
- Holding the control key, Drag from the button to the code file
- Select Action in the connection dropdown
- Name it “showAlert”
- Leave the Event dropdown as it is
- In arguments select “None”

# Walkthrough

- Delete the print statement
- Create an alert
- Present the alert
- Create an action
- Add action to alert
- Run the app

# Conditional Statements

```
let today = "Sunny day"
var action: String

if today == "Sunny day" {
   action = "Go to the park"
} else {
   action = "Stay home"
}
```

# Walkthrough

- Verify if the city in the cityLabel is Barcelona
- If it is then set the message to “You’re Awesome. I am Barcelona”
- If it is not then set the message to “I'm sorry... I'm not London”
- Change the city label.text in viewDidLoad to “London”
- Run the app

# What have we learned so far?

- ViewControllers:
  - they control a screen
  - are made of 2 parts: the design and code
- Labels:
  - Can change some attributes like color and size
  - Can connect then to code

# What have we learned so far?

- Buttons:
  - Can connect to code
  - React to user interaction by showing an alert
- Conditional statements

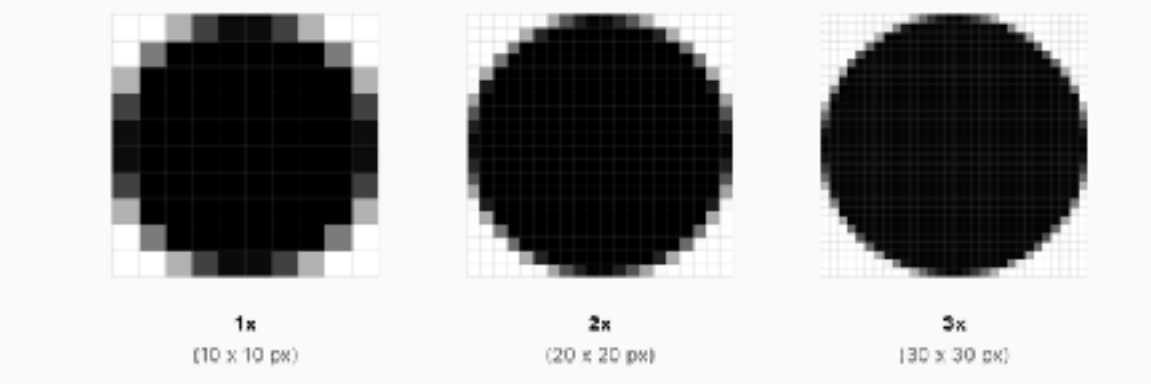
# Solving Problems

- Open the ViewController code file
- Select the name of the IBAction (“showAlert") and substitute it with “showCity1”
- Run the app and tap the button
- Look at the log report
- Disconnect the “showAlert” and connect the “showCity1”
- Run the app and tap the button

# Adding images

Go to https://github.com/ananogal/Workshop-Swift-for-beginners and download the folder
Add images to the Assets catalog.

# Image Size and Resolution



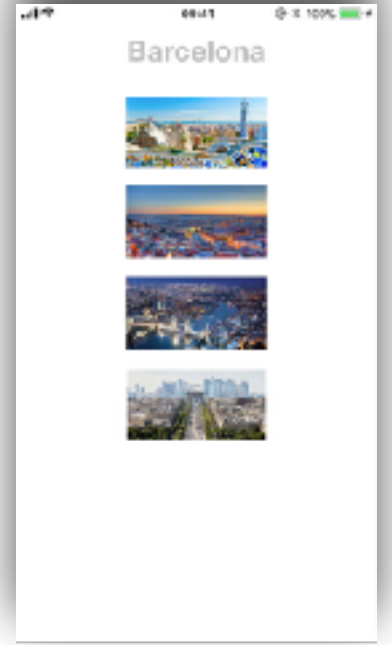

# Walkthrough

- Go to Main.storyboard
- Select the first button
- In the utilities panel, chose the attributes inspector
- Remove the title of the button
- In the image dropdown select the first city

# What have we learned so far?

- ViewControllers
- Labels
- Buttons
- Showing alerts
- If else conditions

# Challenge 1




# Challenge 2

Add 3 more buttons
Add an alert to each button

- Go to Main.Storyboard
- Create a button
- Open the assistant editor
- Holding the control key, Drag from the button to the code file
- Select Action in connection dropdown
- Name it “showCity2”
- Leave Event dropdown as it is
- In arguments select “None”

# Don’t Repeat Yourself

```
@IBAction func showBarcelona() {
    var message = ""
    if cityLabel.text == "Barcelona" {
        message = "Your are Awesome. I am Barcelona!"
    } else {
        message = "I'm sorry... I'm not \(cityLabel.text!)!"
    }

    let alert = UIAlertController(title: "CityQuiz", message:
        message, preferredStyle: .alert)
    let action = UIAlertAction(title: "Awesome", style: .default,
        handler: nil)
    alert.addAction(action)
    present(alert, animated: true, completion: nil)
}
```

# Functions

```
func functionName(parameters) -> ReturnType {
  //body of the function
}
```

# Functions (parameters)

```
func multiply(firstNumber: Int, secondNumber: Int)
{
  let result = firstNumber * secondNumber
  print("The result is \(result).")
}
```

# Functions (Argument Labels)

```
func sayHello(firstName: String) {
  print("Hello, \(firstName)!")
}

sayHello(firstName: "Ana")

sayHello(to: "Ana" and: "Pedro")
```

# Functions (Argument Labels)

```
func sayHello(to: String, and: String) {
  print("Hello, \(to) and \(and)")
}

func sayHello(to person: String, and anotherPerson: String)
{
  print("Hello, \(person) and \(anotherPerson)")
}
```

# Functions (Argument Labels)

```
func sayHello(_ person: String, _ anotherPerson: String) {
  print("Hello, \(person) and \(anotherPerson)")
}


sayHello("Ana", "Pedro")
```

# Functions (Default Values)

```
func display(teamName: String, score: Int = 0) {
    print("\(teamName): \(score)")
}

display(teamName: "Wombats", score: 100) //"Wombats: 100”
display(teamName: "Wombats") //"Wombats: 0"
```

# Challenge 3

Review your code and create functions that can be called with different arguments.

# Resources

RayWenderlich.com

HackingWithSwift.com

Apple Swift Book

Human Interface Guidelines iOS