Swift

BY KARINA BERÇAN AND SYMONNE FONTENOT

Introduction

General-purpose, high-level programming language

Used to write apps for Apple products

Protocol-oriented, object-oriented, functional, procedural

Compiled

IDEs

History

Chris Lattner of Apple, Inc.

Revealed in June of 2014 and released version 1.0 September 9th, 2014

Built on Objective-C and can be used alongside it but aims to replace it

Intended to be easier to use than Objective-C

Open-source

The Basics

Declaring Variables

- Naming conventions
- Strongly typed
- Statically typed

Supports Narrowing and Widening Conversion

Data Types

```
/***** examples *****/

// name is a constant
let name = "Karina"
// age and favColor are variables
var age: Int = 20
var favColor = "yellow"

// x is a Double
var x = Double(age) + 1.5
// y is an Int (and it rounds down to 21)
var y = age + Int(1.5)
```

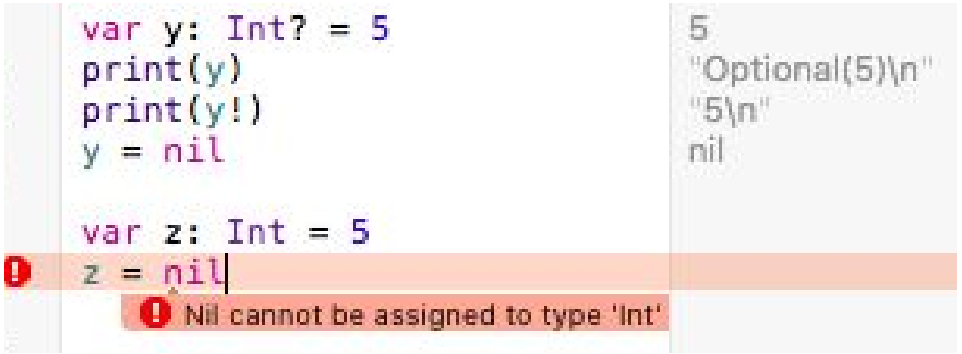
```
// decompose a tuple by indexing
var height = (5,8)
print(height.0)
print(height.1)
// or by naming each component
var (feet, in) = height
print(feet)
print(in)
```

Optionals

When is it used?

How do you create one?

How do you use it?



Collection Types

Array: ordered collection of values

Set: unordered collection of values that are not repeated

Dictionary: unordered collection of key-value pairs

```
/**** examples *****/

// declaring arrays
var arry = [Int]()
var aRRY = [1, 2, 3]

// declaring sets
var catNames = Set<String>()
var genres: Set = ["romance", "science fiction",
"comedy"]

// declaring dictionaries
var seasons = [String: String]()
var airports: [String: String] = ["BOS": "Logan",
"GRU": "Guarulhos", "GIG": "Galeao"]
```

Flow of Program Control

Conditionals

- If/else if/else
- Switch
- Guard

Loops

- For-in
- While/repeat-while

```
// switch-case example
var state = "MA"
switch state{
case "CT","ME","MA","NH","VT":
    print("New England")
    fallthrough
default:
    print("I was too lazy to type all the abbreviations. Sorry.")}

// for-in example
for _ in 1...3{
    print("Betelgeuse")}
```

Flow of Program Control continued

Functions

```
//function that takes in two parameters and returns a boolean
func intro(friend1: String, friend2: String) -> Bool{
      print("\(friend1), this is \(friend2).")
      return(true)
}
friends = intro(“Symonne”, friend2: “Karina”)
```

Scope

- Global and local variables
- Files within the same folder