

# An Introduction to the Swift Programming Language

Sarah Reichelt

# What is Swift?

- Language introduced by Apple in 2014
- Open-sourced 2015
- Used for all Apple devices
- Now extending to Linux and server-side apps

*Swift is a general-purpose programming language built using a modern approach to safety, performance, and software design patterns.*

# How to follow along

- If you have a Mac, use Xcode and a Swift Playground
- If you do not have a Mac, use the “We ❤️ Swift” online sandbox
- Go to <http://troz.net/ncss2018> for a direct link



```
1 print("👋 🌍 !")
```

```
2
```

```
3
```

```
4
```

👋 🌍 !

We ❤️ Swift



# Using the sandbox:

- Swift is designed for writing apps.
- In apps, the input comes from controls: buttons, sliders, text fields etc.
- You will set inputs using variables & constants. Just imagine that these are coming from user input and then change them manually to test different things.
- You will display your results using the "print" command.

**Data**

# Variables

```
var temperature = 29  
print(temperature)
```

```
temperature = 12  
print(temperature)
```

```
29  
12
```

# Constants

```
let cityName = "Sydney"  
print(cityName)
```

```
cityName = "Melbourne"
```

```
code:4:10: error: cannot assign to value:
```

```
'cityName' is a 'let' constant
```

```
cityName = "Melbourne"
```

```
~~~~~ ^
```

```
code:1:1: note: change 'let' to 'var' to make it  
mutable
```

```
let cityName = "Sydney"
```

```
^~
```

```
var
```



# Naming variables & constants

- Variable names can contain almost any character, including emojis.
- They cannot contain spaces, punctuation or maths symbols.
- They cannot start with a number.

# Names

```
// Valid Names
```

```
let myVariable = 42  
let next_number = 43  
let 🐛 = "pile of poo"  
let phase_1_active = true
```

```
// Invalid names
```

```
let first number = 37  
let 2Much = 4_000_000  
let a+b = "AB"
```

*Strings MUST be surrounded by double quotes.  
Single quotes are not valid for Swift strings.*

# Types

```
let anotherCityName = "Sydney"  
print(type(of: anotherCityName))
```

```
var anotherTemperature = 29  
print(type(of: anotherTemperature))
```

```
let distance = 13.54  
print(type(of: distance))
```

```
var isRaining = false  
print(type(of: isRaining))
```

String  
Int  
Double  
Bool

# Comments

```
// This is a single line comment
```

```
let x = 3    // Comment on the end of a line
```

```
/*  
This is a multi-line comment.  
You start and end these with  
slashes & asterisks.  
*/
```

*Comments are very important when you have to come back to code  
or are trying to understand someone else's code.*

# Strings

# Strings

```
let string1 = "I am a string!"  
print(string1)
```

```
let robotHead = "🤖"  
print("Here is a \(robotHead) inside  
another string.")
```

```
let theAnswer = 42  
print("The answer is \(theAnswer)!")
```

```
I am a string!  
Here is a 🤖 inside another string.  
The answer is 42
```

# Looking at Strings

```
let titleString = "NCSS"  
for character in titleString.characters {  
    print(character)  
}  
  
print(titleString.characters.count)  
print(titleString.isEmpty)
```

```
N  
C  
S  
S  
4  
false
```

# Parts of Strings

**Swift 4 syntax - works in Xcode, not in sandbox!**

```
let welcome = "Welcome to Swift"
```

```
let firstWord = welcome.prefix(7)  
print(firstWord)
```

```
let lastWord = welcome.suffix(5)  
print(lastWord)
```

```
let words = welcome.components(separatedBy: " ")  
print(words)
```

Welcome

Swift

["Welcome", "to", "Swift"]



# Collections

# Arrays

```
var cities = ["Sydney", "Brisbane", "Melbourne"]  
print(cities)
```

```
cities.append("Canberra")  
print(cities)
```

```
let brisIndex = cities.index(of: "Brisbane") // 1  
let darwinIndex = cities.index(of: "Darwin") // nil
```

```
cities.remove(at: brisIndex)  
print(cities)
```

```
["Sydney", "Brisbane", "Melbourne"]
```

```
["Sydney", "Brisbane", "Melbourne", "Canberra"]
```

```
["Sydney", "Melbourne", "Canberra"]
```

# Sets

```
var citySet: Set = ["Sydney", "Brisbane", "Perth"]  
print(citySet)
```

```
citySet.insert("Canberra")  
print(citySet)
```

```
citySet.insert("Sydney")  
print(citySet)
```

```
citySet.remove("Perth")  
print(citySet)
```

```
["Sydney", "Brisbane", "Perth"]  
["Sydney", "Brisbane", "Perth", "Canberra"]  
["Sydney", "Brisbane", "Perth", "Canberra"]  
["Sydney", "Brisbane", "Canberra"]
```

# Dictionaries

```
var cityTemperatures = [  
    "Sydney": 29, "Brisbane": 36, "Melbourne": 10  
]  
print(cityTemperatures)  
  
cityTemperatures["Canberra"] = 23  
print(cityTemperatures)
```

```
["Sydney": 29, "Brisbane": 36, "Melbourne": 10]  
["Sydney": 29, "Brisbane": 36, "Canberra": 23,  
"Melbourne": 10]
```

# Dictionaries 2

```
let melbourneTemp = cityTemperatures["Melbourne"]
print(melbourneTemp)

let perthTemp = cityTemperatures["Perth"]
print(perthTemp)

cityTemperatures.removeValue(forKey: "Brisbane")
print(cityTemperatures)

cityTemperatures.removeValue(forKey: "Perth")
print(cityTemperatures)
```

Optional(10)

nil

["Sydney": 29, "Melbourne": 10]

["Sydney": 29, "Melbourne": 10]

# Decisions

# if

```
var temperature = 32  
  
if temperature > 30 {  
    print("It is hot today.")  
}
```

It is hot today.

# if ... else

```
var anotherTemperature = 24

if anotherTemperature > 30 {
    print("It is still hot today.")
} else {
    print("Not too hot now.")
}
```

Not too hot now.



# if ... else if ... else

```
var snowIsForecast = false
var rainIsForecast = true

if snowIsForecast == true {
    print("Better stay at home.")
} else if rainIsForecast == true {
    print("Take an umbrella!")
} else {
    print("Good weather expected today...")
}
```

Take an umbrella!

# switch

```
var windDirection = "S"

switch windDirection {
case "N":
    print("Northerly wind")
case "E":
    print("Easterly wind")
case "S":
    print("Southerly wind")
case "W":
    print("Westerly wind")
default:
    print("Wind from some other direction")
}
```

Southerly wind

# Loops

# for

```
for i in 0 ..< 3 {  
    print(i)  
}
```

```
let cities = ["Sydney", "Brisbane", "Melbourne"]  
for city in cities {  
    print(city)  
}
```

```
0  
1  
2  
Sydney  
Brisbane  
Melbourne
```

# while

```
var counter = 0  
  
while counter < 4 {  
    print(counter)  
    counter += 1  
}
```

```
0  
1  
2  
3
```

# repeat ... while

```
var repeatCounter = 0
repeat {
    print(repeatCounter)
    repeatCounter += 2
} while repeatCounter < 10
```

```
0
2
4
6
8
```

# Looping through Dictionaries

```
var cityTemperatures = [  
    "Sydney": 29, "Brisbane": 36, "Melbourne": 10  
]  
  
var totalTemperatures = 0  
for (key, value) in cityTemperatures {  
    print("In \(key) it is \(value)°")  
    totalTemperatures += value  
}  
  
let averageTemperature = totalTemperatures /  
    cityTemperatures.count  
print(averageTemperature)
```

```
In Sydney it is 29°  
In Brisbane it is 36°  
In Melbourne it is 10°  
25
```

# Functions



# Simple Function

```
func showWeather() {  
    print("Sunny")  
}  
  
showWeather()
```

Sunny

# Function parameters

```
func showWeather(weather: String) {  
    print("The weather is \"(weather)\")  
}  
showWeather(weather: "Cold")  
  
func showWeather(weather: String,  
                 temperature: Int) {  
    print("The weather is \"(weather)\")  
    print("and it is \"(temperature) degrees.\"")  
}  
showWeather(weather: "Wet", temperature: 12)
```

```
The weather is Cold  
The weather is Wet  
and it is 12 degrees.
```

# Better function calls

```
func changeTemperature(newTemperature: Int) {  
    print("Changing temperature to  
          \ (newTemperature)")  
}  
changeTemperature(newTemperature: 12)  
  
func adjustTemperature(to newTemperature: Int) {  
    print("Adjusting temperature to  
          \ (newTemperature)")  
}  
adjustTemperature(to: 34)
```

Changing temperature to 12  
Adjusting temperature to 34

# Returning values

```
func square(of number: Int) -> Int {  
    return number * number  
}
```

```
let result = square(of: 4)  
print(result)
```

# Un-named parameters

```
func convertCtoF(_ degreesC: Double) -> Double {  
    let degreesF = degreesC / 5 * 9 + 32  
    return degreesF  
}
```

```
let boiling = convertCtoF(100)  
let human = convertCtoF(37)
```

```
print(boiling)  
print(human)
```

212.0

98.6

# Default parameters

```
func showWeather(_ weather: String,  
                 useUpperCase: Bool = false) {  
    var weatherInfo = "It is \(weather)."  
    if useUpperCase {  
        weatherInfo = weatherInfo.uppercased()  
    }  
    print(weatherInfo)  
}
```

```
showWeather("Snowing", useUpperCase: true)  
showWeather("Cloudy")
```

```
IT IS SNOWING.  
It is Cloudy.
```

# Optionals

# Optional Variables

```
var weatherReport: String?  
print("\(weatherReport)")  
  
weatherReport = "Mostly sunny"  
print("\(weatherReport)")
```

```
nil  
Optional("Mostly sunny")
```



# Un-wrapping Optionals

```
var number: Int?
```

```
let triple1 = number * 3
```

```
// error: value of optional type 'Int?' not  
unwrapped; did you mean to use '!' or '?'?
```

```
let triple2 = number! * 3
```

```
// fatal error: unexpectedly found nil while  
unwrapping an Optional value
```

```
number = 7
```

```
let triple3 = number! * 3
```

```
print(triple3)
```

# Checking for optionals

```
var weatherReport: String?  
  
if let report = weatherReport {  
    print("\(report)")  
} else {  
    print("No weather report available.")  
}
```

No weather report available.

# Programming Style

# Structure

- Pick a naming style and stick to it: camelCase or snake\_case.
- Use descriptive names even if they are long.
- Be consistent with how you place your braces.
- USE WHITE SPACE!
- Comment "why", not "what".
- **Remember that the person who has to work out this code in 6 months time might be you!!!**

# Apple's Guidelines

- Clarity at the point of use is your most important goal.
- Clarity is more important than brevity.
- Concise code is a consequence of using contextual cues.