Swift

Jeremy Day
Steven Dao
Nicole Lazar
Overview

• Swift is primarily used to create applications for Apple’s iOS platform, as well as the OS X operating system.

• Swift is a multi-paradigm language. It features: the protocol-oriented programming paradigm, object-oriented programming, functional programming, imperative programming, and block structuring.

• Swift was introduced as an alternative to using Objective-C for iOS development. According to the June 2015 RedMonk Programming Language rankings, Objective-C is still more popular than swift (#10 vs #18), but Swift’s ranking is still impressive due to the fact it was only announced in June of last year.
History of Swift

• Swift was first announced by Apple at WWDC on June 2, 2014.

• Apple also released their manual on Swift, *The Swift Programming Language* at the same time.

• Swift left beta on September 9th, 2014. Several new iterations of the version have been released since then, with Swift 2.0 being announced at WWDC this year.

• Development of Swift was started in 2010 by Chris Lattner, but many engineers at Apple had a part in Swift’s development.

• According to Chris Lattner, Swift draws ideas from “Objective-C, Rust, Haskell, Ruby, Python, C#, CLU, and far too many others to list.”
Problem Domain

• Swift’s primary use is in building applications for iOS, which is the operating system on most Apple’s mobile devices including the iPhone and iPad. Programs written in Swift will also work on watchOS and tvOS, the operating systems for Apple Watch and Apple TV, respectively.

• Swift started out as a proprietary language, but will transition into open source later this year. After its transition to open source, Swift applications will be able to run on OS X and Linux as well.
Language Design & Concepts

Objective-C Interoperability:

- Because Swift still has access to Objective-C libraries and APIs, the two languages can co-exist together.

Syntactic sugar:

- Use the `var` keyword to define a variable
- Use the `let` keyword to define a constant

Type Inference:

- Infers the type based on the value

<table>
<thead>
<tr>
<th>Java</th>
<th>Swift</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>String x = &quot;Hello&quot;;</code></td>
<td><code>var x = &quot;Hello&quot;</code></td>
</tr>
<tr>
<td></td>
<td><code>var x: String = &quot;Hello&quot;</code></td>
</tr>
</tbody>
</table>
Optionals:

- This new feature in Swift allows references or values to operate in a manner similar to the common pattern in C, where a pointer may refer to a value or may be null.
  - This implies that non-optional types cannot result in a null-pointer error; the compiler can ensure this is not possible.
- To indicate that a variable is optional, the question mark operator is placed after the variable name.
  - The exclamation point operator is used to force unwrap the optional, only if it is known to not be empty or null.

```swift
var age: Int? = nil
var height: Int? = 180
```
Language Design & Concepts

Protocols:

• Also a feature of Objective-C, protocols are used to ensure that a class implements a set of methods, just like interfaces do in other languages.

Tuples:

• Because Swift contains advanced types, such as tuples, it allows for functions to return more than one value.

• Swift also supports pattern matching in switch statements.
Access Denied
Invalid username and/or password.

Title
Message
Ok Button
import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    var username = TextField()
    var password = TextField()

    override func touchUpInside() {
        // Handle the touch event
    }
}
class ViewController: UIViewController {
    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }
    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }
}

Drag 1 Button to Storyboard
Connection from Storyboard to Code: Ctrl + Drag Text Field to Swift File
Connection from Storyboard to Code:
Ctrl + Drag Button to Swift File
Select Action for Connection Type
import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var username: UITextField!
    @IBOutlet weak var password: UITextField!

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

    @IBAction func login(sender: AnyObject) {
        if (username.text == "admin" && password.text == "admin") {
            let title: String = "Access Approved"
            let message: String = "You may now proceed."
            createAlert(title, message: message)
        } else {
            let title: String = "Access Denied"
            let message: String = "Invalid username and/or password."
            createAlert(title, message: message)
        }
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarningMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    func createAlert(title: String, message: String) {
        let alertController = UIAlertController(title: title, message: message, preferredStyle: UIAlertControllerStyle.Alert)
        let confirmed = UIAlertAction(title: "OK", style: UIAlertActionStyle.Default, handler: nil)
        alertController.addAction(confirmed)
        self.presentViewController(alertController, animated: true, completion: nil)
    }
}
Comparison (Objective-C)

• Swift is easier to read

<table>
<thead>
<tr>
<th>Objective-C</th>
<th>Swift</th>
</tr>
</thead>
<tbody>
<tr>
<td>var x = 1</td>
<td>var x: Int</td>
</tr>
<tr>
<td>var y = &quot;Hello&quot;</td>
<td>x = 2</td>
</tr>
</tbody>
</table>

• No semicolons or parentheses for if-else statements

• Swift requires less code

• Swift is faster and safer

• Optional typing
References

- [http://www.toptal.com/swift/from-objective-c-to-swift](http://www.toptal.com/swift/from-objective-c-to-swift)
- [http://nondot.org/sabre/](http://nondot.org/sabre/)
- [https://developer.apple.com/swift/](https://developer.apple.com/swift/)
- [https://redmonk.com/sogrady/2015/07/01/language-rankings-6-15/](https://redmonk.com/sogrady/2015/07/01/language-rankings-6-15/)
Swift

Jeremy Day
Steven Dao
Nicole Lazar