



Swift & iOS Uygulama Geliştirme Eğitimi

Eğitimci



- Süleyman Çalık
 - suleyman.calik@wissenakademie.com
 - suleymancalik@gmail.com
- Gebze Yüksek Teknoloji Enstitüsü (Bilgisayar Mühendisliği)
- 2008'den bu yana mobil uygulama geliştirir.
- Trendyol, Markafoni, Akmerkez, Wallit, Yellow Pages
- ve 20+ mobil uygulama

Eğitim



- iOS mimarisi ve SDK'leri
 - Swift programlama dili
 - Mobil yazılım gereksinimleri
 - Mobil kullanıcı deneyimi
-
- <http://tinyurl.com/wissen-sdu>

iOS



- iPhone, iPad, iPod Touch ve Apple Watch isimli cihazlar üzerinde çalışan işletim sistemi
- iPhone OS 1,2,3
- iOS 4,5,6,7,8
- En güncel versiyon iOS 8

Cocoa, Cocoa Touch & Xcode



- Cocoa: Uygulama geliştirme framework'ü
- Cocoa Touch : Cocoa'nın iOS için özelleşmiş formu
- Xcode : Geliştirme Ortamımız (IDE)
 - Mac App Store'dan ücretsiz indirilebilir
 - Güncel versiyon Xcode 6.1
 - Gerekli bütün araçları içerisinde barındırıyor
 - Oldukça gerçekçi bir simlatöre sahip

iPhone Ailesi



- iPhone & iPhone 3G & iPhone 3GS

iOS 7 ve sonrası

- iPhone 4

iOS 8 ve sonrası

- iPhone 4S
- iPhone 5 & iPhone 5C & iPhone 5S
- iPhone 6 & iPhone 6 Plus

iPhone & iPad & iPod Touch



- iPad
 - iPad 2
 - iPad 3rd Gen. & iPad 4th Gen.
 - iPad Mini & iPad Mini Retina
 - iPad Air
-
- iPod Touch 1-5. Nesil

Apple Watch



- WatchKit SDK Kasım 2014'te yayınlandı
- Cihaz 2015'te piyasaya çıkacak



Objective - C



- Cocoa ve Cocoa Touch platformlarının geliştirme dili
- 80'lerde Brad Cox tarafından oluşturuldu.
- Steve Jobs'ın şirketi Next tüm haklarını satın aldı
- Apple'ın 1996 yılında Next'i satın almasıyla Apple'ın resmi dili haline geldi
- 2014 Eylül ayına kadar ☺

Swift



- Apple'ın taptaze programlama dili
- Çıkış sebepleri,
 - Obj-C 'nin garip syntax'ı
 - Manuel hafıza yönetimi
 - Giriş bariyerinin diğer dillere göre yüksek olması
 - Obj-C'de bir türlü çözilemeyen problemler
- Getirileri
 - Hız
 - Öğrenme ve programlama kolaylığı
 - Daha basit mimari
 - Hafıza yönetimi ile uğraşmama

Swift'e Giriş



```
println("Hello World")
```

Temel Değerler



- Değişken ve sabitler
 - var
 - let
- Tipler
 - Int - Int32 – Int64 - UInt – UInt32 – UInt64
 - Float (32 bit)
 - Double (64 bit)
 - Bool (true, false)
 - String (text)

Değişken & Sabitler



- Değişken tanımlarken ***var***
- Sabit tanımlarken ***let***

Değişken & Sabitler



- Değişken tanımlarken ***var***
- Sabit tanımlarken ***let***

1. `var myVariable = 42`

Değişken & Sabitler



- Değişken tanımlarken ***var***
 - Sabit tanımlarken ***let***
-
1. `var myVariable = 42`
 2. `let myConstant = 123`

Değişken & Sabitler



- Değişken tanımlarken ***var***
 - Sabit tanımlarken ***let***
-
1. `var myVariable = 42`
 2. `let myConstant = 123`
 3. `myVariable = 120`

Değişken & Sabitler



- Değişken tanımlarken ***var***
 - Sabit tanımlarken ***let***
-
1. `var myVariable = 42`
 2. `let myConstant = 123`
 3. `myVariable = 120`
 4. `myConstant = 10`

Değişken & Sabitler



- Değişken tanımlarken ***var***
- Sabit tanımlarken ***let***

1. `var myVariable = 42`
2. `let myConstant = 123`
3. `myVariable = 120`
4. ~~`myConstant = 10`~~

Type Safety and Type Inference



- Değişkenlerin tipini her zaman yazmak zorunda değilsiniz

Type Safety and Type Inference



- Değişkenlerin tipini her zaman yazmak zorunda değilsiniz
 1. `let myDoubleConst:Double = 70`

Type Safety and Type Inference



- Değişkenlerin tipini her zaman yazmak zorunda değilsiniz
 1. `let myDoubleConst:Double = 70`
 2. `let otherDoubleConts = 70.0`

Type Safety and Type Inference



- Değişkenlerin tipini her zaman yazmak zorunda değilsiniz
 1. `let myDoubleConst:Double = 70`
 2. `let otherDoubleConts = 70.0`
 3. `let myIntConst = 70`

Type Safety and Type Inference



- Fakat bir değişkenin tipi sonradan değiştirilemez.

Type Safety and Type Inference



- Fakat bir değişkenin tipi sonradan değiştirilemez.
1. `var customerName = "Süleyman"`

Type Safety and Type Inference



- Fakat bir değişkenin tipi sonradan değiştirilemez.
 1. `var customerName = "Süleyman"`
 2. `customerName = 1000`

Type Safety and Type Inference



- Fakat bir değişkenin tipi sonradan değiştirilemez.
 1. `var customerName = "Süleyman"`
 2. ~~`customerName = 1000`~~

Strings



- Değerler String'lerin içine kolayca yerleştirilebilir.

Strings



- Değerler String'lerin içine kolayca yerleştirilebilir.
- \()

Strings



- Değerler String'lerin içine kolayca yerleştirilebilir.
- \()
- `let balance = 10000`

Strings



- Değerler String'lerin içine kolayca yerleştirilebilir.
- `\()`
 1. `let balance = 10000`
 2. `let balanceSummary = "Bakiyeniz: \(balance) TL"`

Optional Tipler



- Bu değişkenin içinde değer olabilir de, olmayabilir de

Optional Tipler



- Bu değişkenin içinde değer olabilir de, olmayabilir de
- Değer yoksa nil

Optional Tipler



- Bu değişkenin içinde değer olabilir de, olmayabilir de
- Değer yoksa nil
- Optional olmayan değişkene nil atamaya çalışırsanız, uygulama kapanır. (crash)

Optional Tipler



- Bu değişkenin içinde değer olabilir de, olmayabilir de
- Değer yoksa nil
- Optional olmayan değişkene nil atamaya çalışırsanız, uygulama kapanır. (crash)
- Bir değişkene atanacak değerın nil olma ihtimali var onu optional olarak tanımlamak gerekir.

Optional Tipler



- Bu değişkenin içinde değer olabilir de, olmayabilir de
- Değer yoksa nil
- Optional olmayan değişkene nil atamaya çalışırsanız, uygulama kapanır. (crash)
- Bir değişkene atanacak değerın nil olma ihtimali var onu optional olarak tanımlamak gerekir.
- ?

Optional Tipler



- Bu değişkenin içinde değer olabilir de, olmayabilir de
- Değer yoksa nil
- Optional olmayan değişkene nil atamaya çalışırsanız, uygulama kapanır. (crash)
- Bir değişkene atanacak değerın nil olma ihtimali var onu optional olarak tanımlamak gerekir.
- Optional değişkene ilk değer atamak zorunda değilsiniz.

Optional Tipler



• ?

Optional Tipler



- ?
- `var serverResponseCode : Int ?`

Optional Tipler



- ?
- `var serverResponseCode : Int ?`
- `var code:Int = serverResponseCode!`

Optional Tipler



- ?
- `var serverResponseCode : Int ?`
- `var code:Int = serverResponseCode!`
- !

Optional Tipler



- ?
- `var serverResponseCode : Int ?`
- `var code:Int = serverResponseCode!`
- !
- `var serverResponseCode : Int !`

Optional Tipler



- ?
- `var serverResponseCode : Int ?`
- `var code:Int = serverResponseCode!`
- !
- `var serverResponseCode : Int !`
- `var code:Int = serverResponseCode`

Optional Binding



```
if let constantName = someOptional {  
    statements  
}
```

Optional Binding



```
let possibleNumber = "123"
let convertedNumber = possibleNumber.toInt()

if let actualNumber = convertedNumber {

    println(" \'\'(possibleNumber)\'\' içerikli String, sayıya
çevrilebildi: \'\'(actualNumber)\'\'")

}
else {

    println(" \'\'(possibleNumber)\'\' içerikli String, sayıya
çevrilemedi!")

}
```

Xcode & Playground Demo

