

***PG5600* ios** **programming**

Forelesning 1

Sider å lese: 3-11 i TSPL (Swift 2 prerelease)

Håkon bogen

github.com/haakon

www.shortcut.no

Agenda

- Praktisk informasjon
- iOS økosystem
- XCode
- Swift

Praktisk informasjon

- Ikke lenger Objective-C, men Swift
- The Swift Programming Language (Swift 2 prerelease) - Apple
- iOS-programming with Swift - O'Reilly
- en Deleksamen og en innlevering

iOS økosystem

- Portal: *developer.apple.com*
- Enheter: *iPhone, iPad, Apple Watch, (Apple TV?)*
- Plattform: *iOS / Cocoa Touch / Objective-C/Swift*
- Utviklingsverktøy: *XCode, Instruments, Bots*
- Distribusjon: *App Store, iTunes Connect*

XCode 7 (Beta 6)

- last ned på <https://developer.apple.com/xcode/downloads/>

DEMO

Swift

Hva er Swift?

- *Et programmeringsspråk for iOS og OSX laget av Apple*
- *De har fjernet C avhengigheten og Swift tar over for Objective-C*
- *Moderne og kraftig - inspirert av andre moderne språk - Python, Ruby, C#*
- *Fungerer side-om-side med C og Objective-C*
- *Objektorientert, Imperativt, Funksjonelt*

Hvordan ser det ut?

Skrive til console

```
print("Hello world")
```

Konstanter og Variabler

```
let schoolName: String = "Westerdals ACT"
```

```
var numberOfemployees: Int = 320  
numberOfemployees = 500
```

Type inference

```
var numberOfemployees = 350  
numberOfemployees = 500
```

```
let schoolName = "Westerdals ACT"
```

Strenger

```
let subject = "iOS programming"

if(subject == "iOS programming") {
    println("😘 " + subject)
}
```

Og ja, du kan bruke emojis 🤪

```
let 🚀 = "falcon 9 rocket"
```

String interpolation

```
let n1 = 10, n2 = 8
let mathString = "\n1) ganger \n2) er \n1 * n2)"
//10 ganger 8 er 80
```

```
let rocket = "Falcon 9"
print("Det ble skutt opp en \n(rocket)")
print("\n(rocket.characters.length) // 8
```

Numbers

```
let heltall = 30 // Int
let heltall2 : Int = 25
let desimaltall = 30.45
let desimaltall2 : Float = 30.456 // 32 bit presisjon
let desimaltall3 : Double = 30.456 // 64 bit presisjon // Double er default type inferred
```

Tuples

Grupperer flere verdier inn i en verdi. Godt egnet i funksjoner som skal returnere flere verdier - ikke så godt for komplekse datastrukturer.

```
let (x, y) = (50, 300)
```

```
let error = (455, "Login unavailable")  
print(error.1) // Login unavailable
```

```
let (statusCode, message) = error  
print(statusCode) // 455
```



```
// Navngitte felter i tupler  
let error = (code: 455, message: "Login unavailable")  
print(error.message) // Login unavailable
```

Optionals

```
var possibleWelcomeMessage: String? = "Halloo"

if possibleWelcomeMessage == nil {
    println("Det er ingen streng her!")
} else {
    // unwrapping med !
    let welcomeMessage = possibleWelcomeMessage!
    println(welcomeMessage)
}
```

```
<br />
```

```
if let welcomeMessage = possibleWelcomeMessage {  
    println(welcomeMessage)  
} else {  
    // Håndter at verdien var nil  
}
```

Ranges

```
for nr in 1...10 { // 1 t.o.m. 10
    print("\(nr) ganger 2 er \(nr*2)")
}
```

```
for nr in 1...<10 { // 1 t.o.m. 9
    print("\(nr) ganger 2 er \(nr*2)")
}
```

Collection types

Det finnes tre typer collections i Swift

- *Arrays*
- *Dictionaries*
- *Set*

Array

```
// Deklaring
```

```
let jobs = [String]()
```

```
let jobs = Array<String>()
```

```
// merk ingen spesifisering av type
```

```
var jobs = ["Systemutvikler", "Prosjektleder", "Frontend-Utvikler"]
```

```
// Uthenting
```

```
jobs[0]
```

```
// Iterering
```

```
for job in jobs {
```

```
    print(job)
```

```
}
```

```
// Modifisering
```

```
jobs.append("Rådgiver")
```

```
jobs += ["Rådgiver", "Backend-Utvikler"]
```

```
jobs[0] = "Backend-Utvikler"
```

```
jobs[2..<5] = ["Rådgiver", "Backend-Utvikler", "Prosjektleder"]
```


Dictionary

```
// Deklaring
let emptyDictionary = [String: Float]()
var jobs = ["Rådgiver" : 35, "Systemutvikler" : 21, "Prosjektleder" : 32]

// Uthenting
jobs["Systemutvikler"] // = 35

for (key, value) in jobs {
    println("\(key): \(value)")
}

// Modifisering
people["Rådgiver"] = 45
people["Prosjektleder"] = 81
```

Set

```
var people = Set(["Elon Musk", "Neil DeGrasse Tyson", "Bill Nye"])  
people.remove("Elon Musk")    // returnerer nil hvis den ikke finnes  
people.insert("Carl Sagan")
```

Løkker

Det finnes fire typer løkker

- *while*
- *do-while*
- *for*
- *for-in*

Control flow

- *if*
- *switch*

if

- *Ikke noe nytt, parenteser er valgfritt*

```
if age <= 10 {  
    println("Barn")  
} else if age >= 80 {  
    println("Gammel")  
} else {  
    println("I mellom en plass?;)")  
}
```

switch

Her har de gått banans ;-)

- *Du trenger ikke bruke break etter hver case*
- *Bruk hvilke som helst object i en switch*
- *Hvis du utelater default case må switch'ene matche alle case, hvis ikke blir det compile error*

```
switch age {  
case 1,2,3,4,5,6,7,8,9,10:  
    println("Du er et barn!")  
case 80...100:  
    println("Du er gammel!")  
default:  
    println("I mellom en plass?;)")  
}
```

Du kan switche på instanser

```
let childButton = UIButton()  
let oldisButton = UIButton()  
  
switch sender {  
    case childButton:  
        println("Du er et barn!")  
    case oldisButton:  
        println("Du er gammel!")  
    default:  
        println("I mellom en plass?;)")  
}
```


Playground

- *Kjører kode forløpende*
- *Test ut nye algoritme, tester, utforsk apier, rett en bestemt bug*

Playground Demo

REPL

read-eval-print loop støtte

Startes med `xcrun swift`

Swift online editor

<http://swiftstub.com/>

**For å kjøre Swift i terminalen
må man sette nyeste xcode beta med
xcode-select**

*xcode-select -s /Applications/Xcode-beta.app/Contents/
Developer*

Videre lesning

- 3-11 i TSPL (Swift 2 prerelease)
- <http://bit.do/the-basics> - Apple-dokumentasjon om basics i Swift
- <http://bit.do/control-flow> - Apple-dokumentasjon om control flow

Oppgaver

<http://bit.do/forelesning1>

**Forelesningen er basert på fjorårets foiler, laget av
Hans Magnus Inderberg og Mads Mobæk**