

PG5600 IOS

PROGRAMMING

FORELESNING 1

HANS MAGNUS INDERBERG

@HINDERBERG

MADS MOBÆK

@MADSMOBAEK

AGENDA

- ▶ **Praktisk informasjon**
- ▶ **Kort intro til til OSX**
- ▶ **iOS økosystem**
 - ▶ **XCode**
 - ▶ **Swift**

PRAKTISK INFORMASJON

- ▶ Ikke lenger Objective-C, men Swift
- ▶ Utgår: Objective-C Programming: The Big Nerd Ranch Guide
- ▶ Inngår: The Swift Programming Language (fra iBooks)
 - ▶ Deleksamen og innlevering

KORT INTRO TIL TIL OSX

DEMO

IOS ØKOSYSTEM

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

XCODE

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
 - ▶ Fokus på Enkelhet og sikkerhet
 - ▶ Moderne og kraftig
 - ▶ Fungerer side-om-side med C og Objective-C
- ▶ Paradigmer som imperativ, OOP og funksjonelt

**HVORDAN SER
DET UT?**

```
println("Hello fellow coder!")
```

KONSTANTER OG VARIABLER

```
let shortName: String = "NITH"
```

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

TYPE INFERENCE

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

```
let shortName = "NITH"
```

STRENGER

```
let subject = "iOS programmering"  
println("\(countElements(subject)) tegn")  
if(subject == "iOS programmering") {  
    println("😘 " + subject)  
}
```

Og ja, du kan bruke emoticons 😜

STRING INTERPOLATION

```
let n1 = 10, n2 = 8
```

```
let mathString = "\(n1) ganger \(n2) er \(n1 * n2)"
```

```
//10 ganger 8 er 80
```

NUMBERS

```
let heltall = 30 // Int32 / Int64 avhengig av plattform
let heltall2 : Int = 25
let desimaltall = 30.45
let desimaltall2 : Float = 30.456 // 32 bit presisjon
let desimaltall3 : Double = 30.456 // 64 bit presisjon
```


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")  
println(error.1) // Login unavailable
```

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

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

OPTIONALS

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

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

```
if let welcomeMessage = possibleWelcomeMessage {  
    println(welcomeMessage)  
}
```

RANGES

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

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

COLLECTION TYPES

Det finnes to typer collections i Swift

- ▶ Arrays
- ▶ Dictionaries

ARRAY

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

// merk ingen spesifisering av type
var jobs = ["Systemutvikler", "Prosjektleder", "Frontend-Utvikler"]

// Uthenting
jobs[0]

// Iterering
for job in jobs {
    println(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"]

for (name, count) in jobs {
    println("\(name): \(count)")
}

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

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

```
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`

FOR Å KJØRE SWIFT I TERMINALEN

MÅ MAN SETTE NYESTE XCODE MED

xcode-select

**xcode-select -s /Applications/Xcode6-Beta6.app/Contents/
Developererents/Developer**

OPPGAVER
SE ØVINGSOPPGAVERNE