

# Swiftworkshop!



# Swift gjør ting lett

```
1 let maximumNumberOfLoginAttempts = 10
2 var currentLoginAttempt = 0
```

```
1 let  $\pi$  = 3.14159
2 let 你好 = "你好世界"
3 let 🐶🐮 = "dogcow"
```

```
var welcomeMessage: String
```

```
welcomeMessage = "Hello"
```

# Arrays i Swift

```
let provinser = ["Manitoba", "Saskatchewan", "Ontario"]  
print("Provinsen er \(provinser[2])"); // Ontario
```

```
var redigerbareProvinser = ["Quebec", "Britisk Columbia", "New Brunswick"]  
redigerbareProvinser[1] = "Nova Scotia"
```

```
print("Canadisk provins: \(redigerbareProvinser[1])"); // Nova Scotia
```

```
redigerbareProvinser.append("Alberta")
```

```
redigerbareProvinser += provinser;
```

```
var territorier:[String] = ["Nuvanut", "Yukon", "Nordvestterritoriene"]
```

# Kontrollflyt

```
for indeks in 1...5 {
    print("\(indeks) ganger 5 er \(indeks * 5)")
}
// 1 ganger 5 er 5
// 2 ganger 5 er 10 ..

for _ in 1...10 {
    print("Jeg skrives ut ti ganger!")
}

let navneliste = ["Gaute", "Roy Olav", "Stein Petter", "Robin"]

for navn in navneliste {
    print(navn);
}

for i in (0..
```

# Funksjoner

```
func personligHilsen(navn:String) {  
    print("Hei \navn)")  
}  
  
personligHilsen("Gaute") // Hei Gaute!
```

```
func minVerdi(arr:[Int]) -> Int {  
    var minstverdi = arr[0]  
  
    for tall in arr {  
        if tall < minstverdi {  
            minstverdi = tall  
        }  
    }  
    return minstverdi  
}  
  
minVerdi([6, 12, 9, -2, 5]) // gir -2
```

# Oppgave

- Lag en funksjon som returnerer den største verdien som er i en array
- Løsningsforslag ligger ute (<https://github.com/applitude/Swift-workshop>)

# Optionals

```
var serverCode: Int? = 404; → {Some 404}  
serverCode = nil;
```

```
if (serverCode? != nil) {  
    println(serverCode!)  
}
```

```
if let kode = serverCode? {  
    println(kode);  
}
```

```
println(serverCode ?? 0);
```

```
println(serverCode != nil ? serverCode! : 0);
```

# Klasser og strukturer

```
class Bil {  
    private var bilmerke:String;  
    private var modell:Int;  
  
    init(bilmerke:String, årsmode:ll:Int) {  
        self.bilmerke = bilmerke;  
        modell = årsmode:ll;  
    }  
}  
  
var lada = Bil(bilmerke: "lada", årsmode:ll: 1984)  
  
struct Punkt {  
    var høyde:Int;  
    var lengde:Int;  
}  
  
var pkt = Punkt(høyde: 30, lengde: 15);  
println(pkt.høyde);  
  
var someInts = [Int]()
```



# Oppgave

- Lag en ToDo-liste som kan hente ut items ut i fra tittel, legge inn items osv
- Hint: La klassen være listen, la et item i listen være representert med en struct
- Løsningsforslag ligger ute (<https://github.com/applitude/Swift-workshop>)

# Operatoroverlasting

```
struct Vektor2D {  
    var x = 0.0  
    var y = 0.0  
}  
  
func + (lhs:Vektor2D, rhs:Vektor2D) -> Vektor2D {  
    return Vektor2D(x: lhs.x + rhs.x, y: lhs.y + rhs.y)  
}  
  
let vektor = Vektor2D(x: 3, y: 1)  
let vektor2 = Vektor2D(x: 2, y: 4)  
  
let vektor3 = vektor+vektor2 // gir oss vektor med x: 5 og y: 5  
  
prefix func ++ (vektor:Vektor2D) -> Vektor2D {  
    return Vektor2D(x: vektor.x+1, y: vektor.y+1)  
}  
  
let nyVektor = ++vektor3 // gir oss vektor med x: 6 og y: 6
```

# Oppgave

- Lag en funksjon som legger sammen 2 TodoLists. La resultatet bli en ny TodoList som kombinerer gjøremålene i de to listene
- Hint: Overlast plussoperatorfunksjonen, la den ta to argumenter. Legg sammen arrayene i objektene og konstruer det nye objektet
- Løsningsforslag ligger ute (<https://github.com/applitude/Swift-workshop>)

# Mer funksjoner

```
1 func sayHelloAgain(personName: String) -> String {
2     return "Hello again, " + personName + "!"
3 }
4 print(sayHelloAgain("Anna"))
5 // prints "Hello again, Anna!"
```

```
1 func swapTwoInts(inout a: Int, inout _ b: Int) {
2     let temporaryA = a
3     a = b
4     b = temporaryA
5 }
```

```
1 var someInt = 3
2 var anotherInt = 107
3 swapTwoInts(&someInt, &anotherInt)
4 print("someInt is now \(someInt), and anotherInt is now \(anotherInt)")
5 // prints "someInt is now 107, and anotherInt is now 3"
```

```
public class Test {
    public static void main(String[] args) {
        boolean boolVar = true;
        tarBool(boolVar);

        // Hvilken verdi vil skrives ut?
        System.out.println(boolVar);
    }
    private static void tarBool(boolean param) {
        param = false;
    }
}
```

# Generics (templates)

```
1 func swapTwoStrings(inout a: String, inout _ b: String) {  
2     let temporaryA = a  
3     a = b  
4     b = temporaryA  
5 }  
6  
7 func swapTwoDoubles(inout a: Double, inout _ b: Double) {  
8     let temporaryA = a  
9     a = b  
10    b = temporaryA  
11 }  
  
1 func swapTwoValues<T>(inout a: T, inout _ b: T) {  
2     let temporaryA = a  
3     a = b  
4     b = temporaryA  
5 }
```

# Et program eksempel

```
import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        funkVerden();
        super.viewDidLoad()
    }

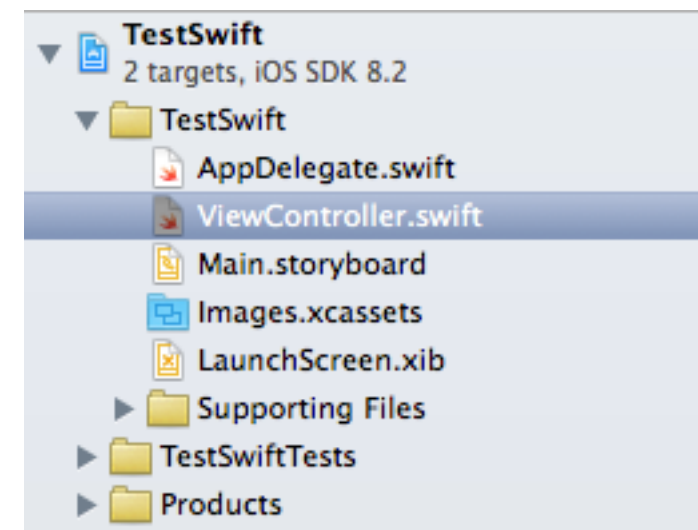
    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
    }

}

class VerdensUndergang {
    func countdown(hilsen:String) {
        for sekunder in reverse(1...5) {
            print("\(sekunder)... ");
        }
        print(hilsen);
    }
}

func funkVerden() {
    var undergang = VerdensUndergang();

    var hilsen:String = "Farewell, cruel world";
    undergang.countdown(hilsen);
}
```



5... 4... 3... 2... 1... Farewell, cruel world

All Output ↕

