

SWIFT 101

SWIFT

- ▶ Lansert i 2014, nå i versjon 3.0
- ▶ Ment på sikt å erstatte det gamle språket Objective-C som hovedspråk for Apple-plattformene
- ▶ Sterkt typet (tenk Java mot Python eller JavaScript)
- ▶ Ikke et tolket, men et compilert språk
- ▶ Tilgjengelig for macOS og Linux, med Windows- og Android-støtte drevet av utviklermiljøet

HELLO WORLD I JAVA

```
1 class HelloWorld {  
2     public static void main(String[] args) {  
3         System.out.println("Hello world");  
4     }  
5 }
```

=> Hello world

HELLO WORLD I SWIFT

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

=> Hello world

VARIABLES

```
1 var name: String = "NTNU"
```

KONSTANTER

```
1 let name: String = "NTNU"
```

KONSTANTER

```
1 let name: String = "NTNU"  
2  
3 name = "UiO"
```

=> **error:** cannot assign to value: 'name' is a 'let' constant

KONSTANTER

```
1 let name: String = "NTNU"
```


TYPE INFERENCE

```
1 let name = "NTNU"
```

UTSKRIFT

```
1 let name = "NTNU"  
2 print("Hello \(name)")
```

=> Hello NTNU

FUNKSJONER

```
1 func square(num: Int) -> Int {  
2     return num*num  
3 }  
4  
5 print( square(num: 3) )
```

=> 9

KLASSER OG OBJEKTER

```
1 class Person {
2     var name: String
3     var age: Int
4
5     init(name: String, age: Int) {
6         self.name = name
7         self.age = age
8     }
9 }
10
11 let p = Person(name: "Øyvind", age: 27)
```

KLASSER OG OBJEKTER

```
1 class Person {
2     var name: String
3     var age: Int
4
5     init(name: String, age: Int) {
6         self.name = name
7         self.age = age
8     }
9 }
10
11 let p = Person(name: "Øyvind", age: 27)
```

```
1 class Person {
2     var name: String
3     var age: Int
4
5     init(name: String, age: Int) {
6         self.name = name
7         self.age = age
8     }
9
10    func birthday() {
11        age += 1
12    }
13
14    func prettyPrint() {
15        print("\(name) er \(age) år gammel")
16    }
17
18    func eats(_ food: String) -> Bool {
19        if (food.lowercased() == "pizza") {
20            return true
21        }
22        return false
23    }
24 }
```

LISTER

```
1 var liste: [String] = []
```

LISTER

```
1 var liste = [String]()
```


LISTER

```
1 var liste = [String]()  
2  
3 liste.append("Hei")  
4 liste.append("NTNU!")
```

LISTER

```
1 var liste = [String]()  
2  
3 liste.append("Hei")  
4 liste.append("NTNU!")  
5  
6 print(liste)
```

=> ["Hei", "NTNU!"]

LISTER

```
1 var liste = [String]()
2
3 liste.append("Hei")
4 liste.append("NTNU!")
5
6 liste += ["Hvordan", "går", "det?"]
7
8 print(liste)
```

=> ["Hei", "NTNU!", "Hvordan", "går", "det?"]

LISTER

```
1 var liste = [String]()
2
3 liste.append("Hei")
4 liste.append("NTNU!")
5
6 liste += ["Hvordan", "går", "det?"]
7
8 for element in liste {
9     print(element)
10 }
```

```
=> Hei
    NTNU!
    Hvordan
    går
    det?
```

LØKKER

```
1 for element in liste {
2     print(element)
3 }
4
5 for (index, element) in liste.enumerated() {
6     print(index)
7     print(element)
8 }
9
10 for index in 1...5 {
11     print(index)
12 }
```

IF-ELSE

```
1 if mat == "pizza" {  
2     print("yes")  
3 } else if mat == "sushi" {  
4     print("yeeees")  
5 } else {  
6     print("no")  
7 }
```

SWITCH

```
1 switch(mat) {  
2     case "pizza":  
3         print("yes")  
4     case "sushi":  
5         print("yeeees")  
6     default:  
7         print("no")  
8 }
```

LISTER

```
1 var liste = [String]()
2
3 liste.append("Hei")
4 liste.append("NTNU!")
5
6 liste += ["Hvordan", "går", "det?"]
7
8 liste.forEach{ print($0) }
```

```
=> Hei
    NTNU!
    Hvordan
    går
    det?
```


OPTIONALS

```
1 func capitol(country: String) -> String? {  
2     switch country {  
3     case "Norway":  
4         return "Oslo"  
5     case "Sweden":  
6         return "Stockholm"  
7     default:  
8         return nil  
9     }  
10 }
```

OPTIONALS

```
1 func capitol(country: String) -> String? {  
2     switch country {  
3     case "Norway":  
4         return "Oslo"  
5     case "Sweden":  
6         return "Stockholm"  
7     default:  
8         return nil  
9     }  
10 }  
11  
12 let capitolOfNorway = capitol(country: "Norway")
```

OPTIONALS

```
1 func capitol(country: String) -> String? {  
2     switch country {  
3     case "Norway":  
4         return "Oslo"  
5     case "Sweden":  
6         return "Stockholm"  
7     default:  
8         return nil  
9     }  
10 }  
11  
12 let capitolOfNorway = capitol(country: "Norway")  
13 print(capitolOfNorway)
```

=> Optional("Oslo")

OPTIONALS

```
1 func capitol(country: String) -> String? {  
2     switch country {  
3     case "Norway":  
4         return "Oslo"  
5     case "Sweden":  
6         return "Stockholm"  
7     default:  
8         return nil  
9     }  
10 }  
11  
12 let capitolOfNorway = capitol(country: "Norway")  
13 print(capitolOfNorway!)
```

=> Oslo

OPTIONALS

```
1 func capitol(country: String) -> String? {  
2     switch country {  
3     case "Norway":  
4         return "Oslo"  
5     case "Sweden":  
6         return "Stockholm"  
7     default:  
8         return nil  
9     }  
10 }  
11  
12 if let capitolOfNorway = capitol(country: "Norway") {  
13     print(capitolOfNorway)  
14 }
```

=> Oslo

OPTIONALS

```
1 func capitol(country: String) -> String? {
2     switch country {
3     case "Norway":
4         return "Oslo"
5     case "Sweden":
6         return "Stockholm"
7     default:
8         return nil
9     }
10 }
11
12 guard let capitolOfNorway = capitol(country: "Norway") else {
13     // Håndter feil dersom capitolOfNorway er nil
14 }
15 print(capitolOfNorway)
```

=> Oslo

MYE MYE MER...

- ▶ Computed properties
- ▶ Extensions
- ▶ Ingen garbage collector, men Automatic Reference Counting (ARC) som minnemodell

PLAYGROUND

SWIFT-OPPGAVER OM KARDASHIANS

- ▶ `git clone https://github.com/bekk/iOS-Lynkurs.git`
- ▶ Åpne Lynkurs-01-Swift.xcodeproj i Xcode
- ▶ I Xcode: Product -> Scheme -> Lynkurs-01-Swift-Tests
- ▶ CMD + U for å kjøre tester og se at det ikke funker
- ▶ Deres oppgave: Fiks koden i Person.swift slik at testene i PersonTests.swift kjører