Swift講習会 #1

Yuhel Tanaka

今回やること

- 変数と定数の宣言
- 配列と辞書の宣言
- for文・if文
- クラスの宣言・インスタンス生成

変数と定数

変数は**var**, 定数は**let**で定義します

```
var hoge = 100 // 変数!
hoge = 120

let huga = 100 // 定数!
huga = 120 // 定数なのでエラー
```

↑こいつらの型は？？

型推論

変数を初期化する際に代入した値を見て型を推測してくれる！

```
var hoge = 100 // Intだな!
var huga:Int = 100 // 型を明示することもできる
var piyo:Int // これでもOK!(使う前に値を代入する必要あり)

NSNumber *hoge = @(100); // Objective-C
```

こんなことはできないよ

Swiftは**静的型付け言語**なので...

```
var hoge = "テスト" // hogeはString型
hoge = 111 // hogeはString型変数なのでInt型は代入できない

## Python(動的型付け言語)の場合
hoge = "テスト"
hoge = 1
## 変数にどの型のオブジェクトが入っているのかわからない
## JS や Pythonなどが動的型付け言語
```

配列と辞書

```
let array = ["a", "i", "u", "e", "o"]
let array2:[String] = ["a", "i", "u", "e", "o"] // 型を明示す

let dict = ["a": 1, "b": 2, "c": 3] // keyがString,valueがIn
let dict2:[String:Int] = ["a": 1, "b": 2, "c": 3] // 型を明示
```

if文

条件式に括弧をつける必要がなくなった

```
let name = "tanaka"
if name == "tanaka" {
  print("tnk")
}
if (name == "tanaka") { // 括弧をつけてもよい
  print("(tnk)")
}
```

for文(1)

for-in文を使う

```
let numbers = [3, 5, 7]
for number in numbers { // 3, 5, 7
    print(number)
}
```

for文(2)

```
for number in 0...5 { // 0 ~ 5
    print(number)
}
for number in 0..<5 { // 0 ~ 4
    print(number)
}
for (int i=0;i<5;i++) { // Swiftでこの書き方は出来ない
    print(i)
}
```

クラス定義

```
class Test { // Testクラスを定義
}

let ins = Test() // Testクラスのインスタンスを生成
```

プロパティ

型に紐付いた値のこと(車classだったら馬力とか)

- Stored Property
- Computed Property

Stored Property

変数や定数を格納している

```
class Test { // Testクラスを定義
    var name:String = "tanaka"
    let age:Int = 25
}

let ins = Test() // Testクラスのインスタンスを生成
print(ins.name) // "tanaka"
```

Computed Property?

Computed Propertyとは???

Computed Property

値を保持していない

- get -> 読み込み時に呼ばれる
- set -> 書き込み時に呼ばれる

```
class Test { // Testクラスを定義
    var height:Double
    var weight:Double
    var bmi { // Computed Property
        get {
            return weight / height / height
        }
    }
}
```

メソッド定義

func 関数名(引数名:型) -> 戻り値の型 {}

```
class Hello {
  func say() { // 引数も戻り値もないメソッド
    print("hello!")
  }
  func say2(message:String) { // 引数が1つあるメソッド
    print(message)
  }
  func plus(a:Int, b:Int) -> Int { // 引数が2つでInt型の戻り値が
    return a+b
  }
}
```

イニシャライザ

インスタンスを生成する際に呼び出されるメソッド

```
class Man {
    let height:Double
    init(height:Double) { // イニシャライザ
        self.height = height
    }
}

let man = Man(height: 1.93) // インスタンス生成
print(man.height) // 1.93
```

終わり

第1回は終わりです。