```
//: Playground - noun: a place where people can play

import UIKit
// functions  - Lecture 3


//*****************************
// IN-CLASS PRACTICE
//*****************************

func firstFun1(x: Int, y: Int = 5) -> Int {
    var s = "hello"
    return x + y
}

var f = firstFun1

firstFun1(x:3, y:3)
firstFun1(x:3)



func maxAndAvg(array: [Int]) -> (m:Int, Double){
    var max = array[0]
    var sum = array[0]
    for each in array[1..<array.count]{
        if max < each {
            max = each
        }
        sum += each
    }

    let avg = Double(sum)/Double(array.count)
    return (max, avg)
}



var ar = [1,2,5,8,4]

maxAndAvg(array: ar)


func f(a b: Int, _ b1 : Int){

}

f(a: 4, 3)


func add(x: Int, y:Int) -> Int {
    return x+y
}
```

```swift
func multiply(x x1: Int, y y1:Int) -> Int {
    return x1*y1
}

var mathFunction = multiply

mathFunction(5, 4)


func increaseByOne(value: Int = 5) -> Int {
    return value+1
}


// example closure
var clos = { (value: Int) -> Int in
    return value - 1
}

// function definition with a closure in parameter list
func someFunction(x: Int, closure: (Int) -> Int){
    closure(x)
}

//function calls with function and closure in parameter list
someFunction(x: 5, closure: increaseByOne)
someFunction(x: 5, closure: clos)

// function call without trailing closure
someFunction(x: 5, closure: {
    (value: Int) -> Int in return value + 5
})

// function call with trailing closure
someFunction(x: 5) {
    (value: Int) -> Int in return value + 10
}




//******************************
// BEFORE CLASS PRACTICE
//******************************


increaseByOne(value: 8)
increaseByOne()

func sayHelloOrBye(name: String, whichOne: Bool){

    if whichOne {
        print("Hello "+name, terminator:"")
```

```
    }
    else{
        print("Bye "+name, terminator:"")
    }
}

sayHelloOrBye(name: "David", whichOne: false)

//different order nor accepted
//sayHelloOrBye(whichOne: false, name: "David")


var a = 1
var b = 2
var c = 3

print(a,b,c)
//prints "1 2 3\n"

print(a,b,c,separator:"-")
//prints "1-2-3\n"

print(a,b,c,separator:"-", terminator:"")
//prints "1-2-3"


// give me max and avg



let res = maxAndAvg(array: ar)

print(res.0, res.1)

func printName(name: String){
    var name = name
    name = "John"
    print(name)
}

printName(name: "Ben")


// closures by sort example
ar.sort()

// ar.sorted(by: (Int, Int) -> Bool)

let names = ["Chris", "Alex", "Ewa", "Barry", "Daniella"]

var reversedNames = names.sorted(by: { (s1: String, s2: String) -> Bool in return s1
 > s2 } )

var ss = 0
```

```
for i in stride(from: 0, through: 10, by: 2){
    ss += i
}

print(ss)
```