

A Swift Kickstart

Daniel H Steinberg
@dimsumthinking

Functions

Create a playground

We'll do this first section together

Start Xcode

Create an iOS Playground named Functions

No Parameters

```
func helloWorld1() {  
    print("Hello, ")  
    println("World!")  
    println("From beautiful Las Vegas, NV")  
}  
  
helloWorld1()
```

One Parameter

```
func helloWorld2(name: String) {  
    print("Hello, ")  
    println("\(name)")  
    println("From beautiful Las Vegas, NV")  
}  
  
helloWorld2("CocoaConf")
```

Multiple Parameters

```
func helloManyTimes(name: String,  
                    numberOfTimes: Int) {  
    for i in 1 ... numberOfTimes {  
        helloWorld2(name)  
    }  
}  
  
helloManyTimes("Someone", 7)
```

Overloading

Change the function names to hello() and call the second from the first and third

```
func hello() {  
    hello("World")  
}  
  
func hello(name: String) {  
    print("Hello, ")  
    println("\ (name)")  
    println("From beautiful Las Vegas, NV")  
}  
  
func hello(name: String,  
    numberOfTimes: Int) {  
    for i in 1 ... numberOfTimes {  
        hello(name)  
    }  
}
```

Default Values

```
func helloWorld3(name: String = "World") {  
    print("Hello, ")  
    println("\ (name)!")  
    println("From beautiful Las Vegas, NV")  
}
```

```
helloWorld3()  
helloWorld3(name: "CocoaConf Friends")
```


Default Values (cont)

```
func helloManyTimes2(name: String,  
    numberOfTimes: Int = 1) {  
    for i in 1...numberOfTimes {  
        helloWorld2(name)  
    }  
}
```

```
helloManyTimes2("CocoaConf Pals")  
helloManyTimes2("CocoaConf Buddies", numberOfTimes: 4)
```

External Names

```
func hello(personWithName name: String,  
           #numberOfTimes: Int) {  
    for i in 1 ... numberOfTimes {  
        helloWorld2(name)  
    }  
}
```

```
hello(personWithName: "Anabelle",  
      numberOfTimes: 3)
```

External Names (cont)

```
func hi(_ name: String = "World") {  
    println("Hello, \(name)!")  
}
```

```
hi("CocoaConf Companion")
```

Variadic Parameters

```
func hello(people: String...) {  
    for person in people {  
        hi(person)  
    }  
}  
  
hello()  
hello("Friend", "Buddy", "Companion")
```

Return Values

```
func hello(name: String = "World") -> String {  
    return "Hello, \(name)"  
}
```

```
println(hello())  
println(hello(name: "CocoaConf Cats"))
```

Return Values in the Playground

```
func hello(name: String = "World") -> String {  
    return "Hello, \(name)"  
}
```

```
hello()  
hello(name: "CocoaConf Cats")
```

Tuples

```
func helloFriends(people: String...) -> (numberOfFriends: Int, greeting: String) {  
    var tempGreeting = "Hello, "  
    for person in people {  
        tempGreeting += person + "\n"  
    }  
    return (people.count, tempGreeting)  
}
```

```
helloFriends("A","B","C").numberOfFriends  
helloFriends("A","B","C").greeting
```

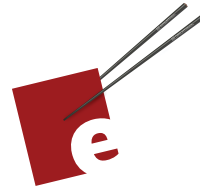
Try this

- Create a function named `display()` that takes a single String parameter named `theCharacter` and returns a String
- `theCharacter` has default value `"\u{2615}"`
- Call the function without any arguments
- Call the function with the argument `"\u{266C}"`

Try this

```
func display(theCharacter: String = "\u{2615}") -> String {  
    return theCharacter  
}
```

```
display()  
display(theCharacter: "\u{266C}")
```



A Swift Kickstart

DANIEL H STEINBERG



Introducing
the Swift Programming Language

Editors Cut

<https://itunes.apple.com/us/book/a-swift-kickstart/id891801923?mt=11&uo=4&at=11156E>