

# **Swift 2 Tutorial: A Quick Start**

quick tour of the Swift language, including variables, control flow, classes, best practices, and more.

# **Introduction to Playgrounds**



## **Get started with a playground**

Explore new ideas quickly and easily.

Choose a template for your new playground:

iOS

tvOS

macOS

Filter

## Playground



Blank



Game



Map



Single View

Cancel

Previous

Next

Save As: SwiftTutorial

Tags:



실습

Search

### Favorites

- [C] Windows 10
- iCloud Drive
- Applications
- Downloads
- Desktop
- Documents
- [C] Windows 10

### Devices

- Remote Disc
- BOOTCAMP

- 트...임프로그래밍
- 트...임프로그래밍
- 트...임프로그래밍
- 트...임프로그래밍
- office 서식 파일
- ...6.External.pdf
- 3.5
- 3.5.tar
- ter
- ...3 오후 4.12.26
- ragon.ipa
- Backups
- s Unitv 5 DVD
- 2018-1 스마트...래밍 강의자료
- 실습
- iOS10 webarchive
- iOS10AppEssentialsPreview
- iOS10BookSamples
- iOS10BookSamples.zip
- Lab
- Ray Wenderlich webarchive
- HelloWorld
- LearnSwift.playground
- MyFirstProject
- MyPlayground.playground
- MyPlayground2.playground
- PeopleDatabase
- Person.playground
- Tap Me
- Tap Me 2

New Folder

Cancel

Create

//swift playground 는 옆 사이드라인에 결과를 보여줌

let swiftTeam = 13

let iOSTeam = 54

let otherTeams = 48

let totalTeam = swiftTeam + iOSTeam + otherTeams

13

54

48

115


---

# **Variables vs. Constants in Swift**



//totalTeam 은 상수로(let) 으로 정의했기 때문에 에러

```
totalTeam += 1
```

 Left side of mutating operator isn't mutable:

//totalTeam 은 상수로(let) 으로 정의했기 때문에 에러

```
totalTeam += 1
```

❖ Left side of mutating operator isn't mutable: 'totalTeam' is a 'let' constant ×

Change 'let' to 'var' to make it mutable

Fix

```
var totalTeam = swiftTeam + iOSTeam + otherTeams
```

//totalTeam 은 상수로(let) 으로 정의했기 때문에 에러

```
totalTeam += 1
```

# Explicit vs. Inferred Typing

```
//swift compiler는 타입을 추론해서 Inferred Type Int로 결정  
let swiftTeam = 13
```

```
//타입을 지정 Explicit Type
```

```
let iOSTeam : Int = 54
```

# **Basic Types and Control Flow in Swift**

# Floats and Doubles



//swift compiler는 실수는 default로 Double로 결정

//실수는 Float 혹은 Double type 가질 수 있음

```
let priceInferred = 19.99
```

```
let priceExplicit : Double = 19.99
```

# Bools

```
//Boolean 타입은 true / false를 가짐  
let onSaleInferred = true  
let onSaleExplicit : Bool = false
```

---

# Strings

//String 타입

let nameInferred = "김영식"

let nameExplicit : String = "오황석"

---

**If statements and string interpolation**

```
29 //String interpolation : 큰 따옴표 안에 변수 값을 함께 쓸 수 있음
30 if onSaleInferred {
31     print("\(nameInferred) on sale for \((priceInferred)!\)")
32 } else {
33     print("\(nameInferred) at regular price: \((priceInferred)!\)")
34 }
35 |
```



**김영식 on sale for 19.99!**

**\(your expression)**



```
28
29 //String interpolation : 큰 따옴표 안에 변수 값을 함께 쓸 수 있음
30 if onSaleInferred {
31     print("\(nameInferred) on sale for \(priceInferred)!")
32 } else {
33     print("\(nameInferred) at regular price: \(priceInferred)!")
34 }
35
```

김영식 on sale  
for 19.99!...

Quick Look



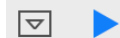
김영식 on sale for 19.99!

4:49  
5:05  
10:31  
10:03  
11:25  
5:44  
11:18  
3:33  
3:34  
3:34

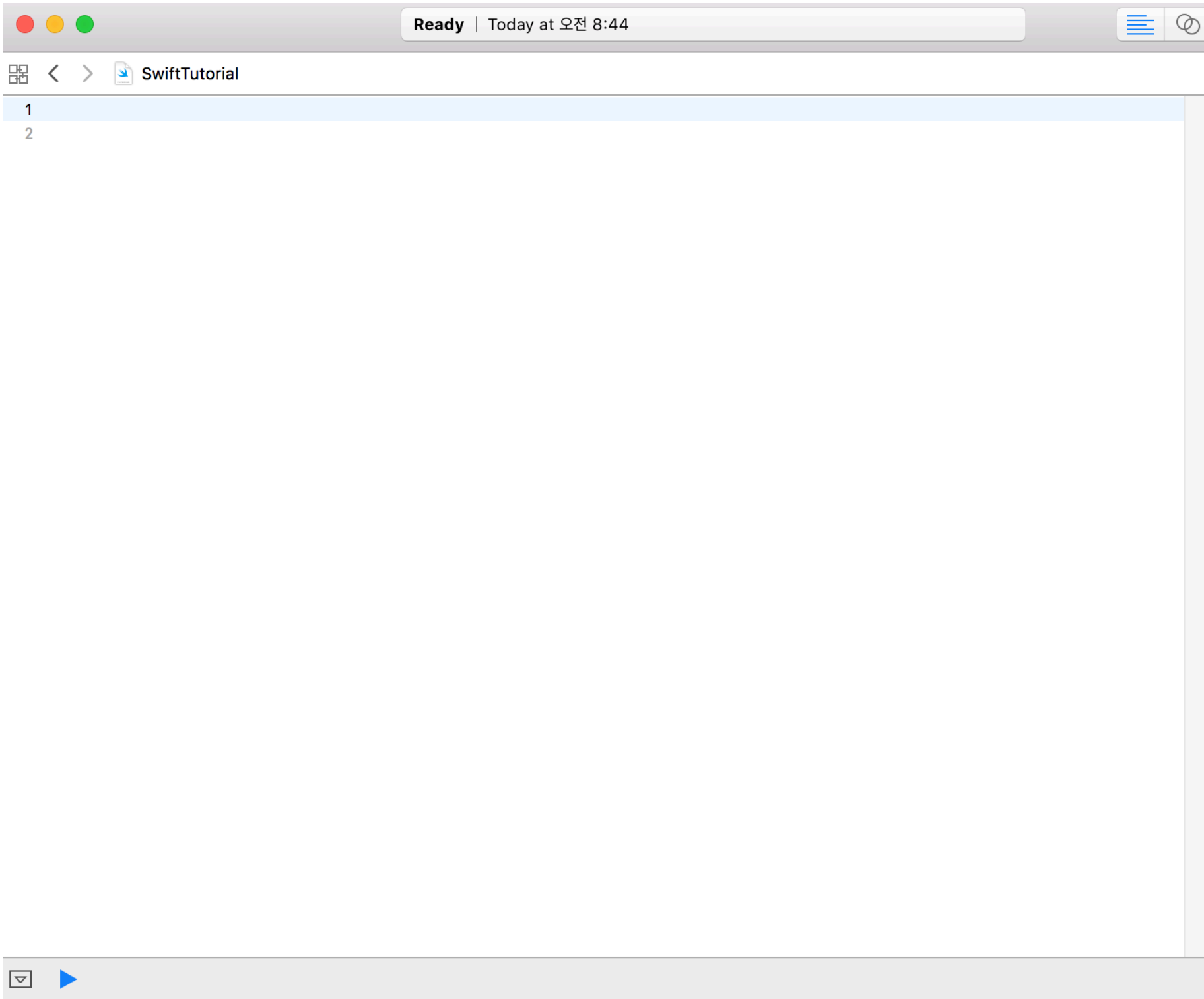
# Classes and Methods

First, delete everything in your playground file so you're at a clean start.

```
1  //: Playground - noun: a place where people can play
2
3  import UIKit
4
5  //swift playground 는 옆 사이드라인에 결과를 보여줌
6  //swift compiler는 타입을 추론해서 Inferred Type Int로 결정
7  let swiftTeam = 13
8  //타입을 지정 Explicit Type
9  let iOSTeam : Int = 54
10 let otherTeams = 48
11 var totalTeam = swiftTeam + iOSTeam + otherTeams
12
13 //totalTeam 은 상수로(let) 으로 정의했기 때문에 에러
14 totalTeam += 1
15
16 //swift compiler는 실수는 default로 Double로 결정
17 //실수는 Float 혹은 Double type 가질 수 있음
18 let priceInferred = 19.99
19 let priceExplicit : Double = 19.99
20
21 //Boolean 타입은 true / false를 가짐
22 let onSaleInferred = true
23 let onSaleExplicit : Bool = false
24
25 //String 타입
26 let nameInferred = "김영식"
27 let nameExplicit : String = "오황석"
28
29 //String interpolation : 큰 따옴표 안에 변수 값을 함께 쓸 수 있음
30 if onSaleInferred {
31     print("\(nameInferred) on sale for \(priceInferred)!")
32 } else {
33     print("\(nameInferred) at regular price: \(priceInferred)!")
34 }
35
```



김영식 on sale for 19.99!



```
//Tip Calculator 클래스 선언  
class TipCalculator {  
  
}
```

```
class TipCalculator {
```


```
  //상수 선언 : Error
```

```
  let total: Double      //post-tax total
```

```
  let taxPct: Double     //tax percentage
```

```
  let subtotal: Double   //pre-tax subtotal
```

```
}
```

 Class 'TipCalculator' has no initializers

//Tip Calculator 클래스 선언

```
class TipCalculator {  
    //상수 선언 : Initializer 에서 초기값을 가지므로 Error 사라짐  
    let total: Double        //post-tax total  
    let taxPct: Double       //tax percentage  
    let subtotal: Double     //pre-tax subtotal
```

//class property 는 선언할 때 혹은 initializer 에서 초기값을 지정해야 함  
//초기값을 가지지 않기 위해서는 optional로 선언해야 함

```
init(total: Double, taxPct: Double) {  
    self.total = total  
    self.taxPct = taxPct  
    subtotal = total / (taxPct + 1)  
}
```

```
}
```



,

```
//tip 계산 함수: tip은 pre-tax subtotal에서 계산해야 함  
func calcTipWithTipPct(tipPct: Double) -> Double {  
    return subtotal * tipPct  
}
```

```
}
```

-  
//15% 18% 20% 팁을 포함한 가격을 리스트 프린트하는 함수

```
func printPossibleTips() {  
    print("15%: \(calcTipWithTipPct(tipPct: 0.15))")  
    print("18%: \(calcTipWithTipPct(tipPct: 0.18))")  
    print("20%: \(calcTipWithTipPct(tipPct: 0.20))")  
}
```

```
}
```

```
27 //클래스 바깥에 인스턴스 생성 및 멤버함수 호출
28 //레스토랑 계산서 total 이 팁을 포함하지 않고 tax만 포함한 것임
29 //팁은 따로 subtotal(tax 포함하지 않는)에서 15~20% 정도 계산해서 테이블 위에 온다고 가정!!
30 let tipCalc = TipCalculator(total: 33.25, taxPct: 0.06)
31 tipCalc.printPossibleTips()
32
```



```
15%: 4.70518867924528
18%: 5.64622641509434
20%: 6.27358490566038
```

```

19 //15% 18% 20% 팁을 포함한 가격을 리스트 프린트하는 함수
20 func printPossibleTips() {
21     print("15%: \(calcTipWithTipPct(tipPct: 0.15))")
22     print("18%: \(calcTipWithTipPct(tipPct: 0.18))")
23     print("20%: \(calcTipWithTipPct(tipPct: 0.20))")
24 }
25 }
26
27 //클래스 바깥에 인스턴스 생성 및 멤버함수 호출
28 //레스토랑 계산서 total 이 팁을 포함하지 않고 tax만 포함한 것임
29 //팁은 따로 subtotal(tax 포함하지 않는)에서 15~20% 정도 계산해서 테이블 위에 온다고 가정!!
30 let tipCalc = TipCalculator(total: 33.25, taxPct: 0.06)
31 tipCalc.printPossibleTips()
32

```

```

"15%: 4.7051886792452...
"18%: 5.6462264150943...
"20%: 6.2735849056603...

```

```

TipCalculator
TipCalculator

```



```

15%: 4.70518867924528
18%: 5.64622641509434
20%: 6.27358490566038

```

# **Arrays and For Loops**

//15% 18% 20% 팁을 포함한 가격을 리스트 프린트하는 함수

//array와 for 루프를 이용해서 다시 작성

```
func printPossibleTips() {
```

```
    //Inferred와 Explicit array 변수 선언: 워닝 무시!
```

```
    let possibleTipsInferred = [0.15, 0.18, 0.20]
```

```
    let possibleTipsExplicit: [Double] = [0.15, 0.18, 0.20]
```



Immutable value 'pos:

```
    //for 루프 : possibleTipsInferred 배열에서 하나씩 꺼내서 루프 실행
```

```
    for possibleTip in possibleTipsInferred {
```

```
        print("\(possibleTip*100)%: \(calcTipWithTipPct(tipPct: possibleTip))")
```

```
    }
```

```
}
```

```
//for 루프 : possibleTipsInferred 배열에서 하나씩 꺼내서 루프 실행
for possibleTip in possibleTipsInferred {
    print("\(possibleTip*100)%: \(calcTipWithTipPct(tipPct: possibleTip))")
}

//다른 스타일의 for 루프
//..< < 연산자는 상한은 포함안함. 반면에 ... 연산자는 상한 포함함
//배열의 원소 개수 연산자: .count, 배열 원소 추출 연산자 : [index]
for i in 0..
```

# Viewing Print Output



to see the print results.





**15.0%: 4.70518867924528**

**18.0%: 5.64622641509434**

**20.0%: 6.27358490566038**

**15.0%: 4.70518867924528**

**18.0%: 5.64622641509434**

**20.0%: 6.27358490566038**

# Dictionaries

Delete the `printPossibleTips()` method

```
//Tip Calculator 클래스 선언
```

```
class TipCalculator {
    //상수 선언 : Initializer 에서 초기값을 가지므로 Error 사라짐
    let total: Double        //post-tax total
    let taxPct: Double        //tax percentage
    let subtotal: Double      //pre-tax subtotal

    //class property 는 선언할 때 혹은 initializer 에서 초기값을 지정해야 함
    //초기값을 가지지 않기 위해서는 optional로 선언해야 함
    init(total: Double, taxPct: Double) {
        self.total = total
        self.taxPct = taxPct
        subtotal = total / (taxPct + 1)
    }
    //tip 계산 함수: tip은 pre-tax subtotal에서 계산해야 함
    func calcTipWithTipPct(tipPct: Double) -> Double {
        return subtotal * tipPct
    }
    /*
    //15% 18% 20% 팁을 포함한 가격을 리스트 프린트하는 함수
    //array와 for 루프를 이용해서 다시 작성
    func printPossibleTips() {
        //Inferred와 Explicit array 변수 선언: 워닝 무시!
        let possibleTipsInferred = [0.15, 0.18, 0.20]
        let possibleTipsExplicit: [Double] = [0.15, 0.18, 0.20]

        //for 루프 : possibleTipsInferred 배열에서 하나씩 꺼내서 루프 실행
        for possibleTip in possibleTipsInferred {
            print("\(possibleTip*100)%: \(calcTipWithTipPct(tipPct: possibleTip))")
        }
        //다른 스타일의 for 루프
        //..< 연산자는 상한은 포함안함. 반면에 ... 연산자는 상한 포함함
        //배열의 원소 개수 연산자: .count, 배열 원소 추출 연산자 : [index]
        for i in 0..
```

```
//tip 계산 함수: tip은 pre-tax subtotal에서 계산해야 함
func calcTipWithTipPct(tipPct: Double) -> Double {
    return subtotal * tipPct
}
```

```
//Dictionary (key/value 쌍을 갖는 자료구조)를 반환하는 함수
```

```
func returnPossibleTips()->[Int: Double] {
    //Inferred array 팁 퍼센트 배열 선언
    let possibleTipsInferred = [0.15, 0.18, 0.20]

    //빈 Dictionary 변수 선언
    var retval = [Int: Double]()

    //for 루프에서 3개 팁 퍼센트에 대한 팁을 계산하고 Dictionary에 추가
    for possibleTip in possibleTipsInferred {
        //배열에서 꺼낸 Double 타입의 possibleTip을 Int로 타입 변환
        let intPct = Int(possibleTip*100)
        //팁 계산한 value값을 Dictionary에 key는 intPct 에 맞춰서 삽입
        retval[intPct] = calcTipWithTipPct(tipPct: possibleTip)
    }
    return retval
}
```

```
let tipCalc = TipCalculator(total: 33.25, taxPct: 0.06)
//tipCalc.printPossibleTips()
tipCalc.returnPossibleTips()
```



```
//클래스 바깥에 인스턴스 생성 및 멤버함수 호출  
//레스토랑 계산서 total 이 팁을 포함하지 않고 tax만 포함한 것임  
//팁은 따로 subtotal(tax 포함하지 않는)에서 15~20% 정도 계산해서 테이블 위에 온다고 가정!!  
let tipCalc = TipCalculator(total: 33.25, taxPct: 0.06)  
//tipCalc.printPossibleTips()  
tipCalc.returnPossibleTips()
```



- ▶ (key 18, value 5.6462264150...
- ▶ (key 15, value 4.7051886792...
- ▶ (key 20, value 6.2735849056...