Swift Study 01

2016. 10.29

Swift 란?

- **wwdc(The Apple Worldwide Developers Conference) 2014**에 발표한 프로그래밍 언어로 애플의 **osx / ios** 등을 개발목적으로 만든 언어

- 기본의 **object-c**보다 성능적으로나 문법적으로나 간편 유연하며 학습하기 편하다는 의견

Swift 란?

- **wwdc 2015**에서 **swift**를 오픈소스**(GPL v3)**로 전환하여 **osx**와 **linux** 환경에서 사용가능하게 되었다.

- 더불어 기존의 **2.2**에서 **3.0**으로 버전업하면서 일부 문법 및 함수 네이밍, 넘버링 등의 전체적인 틀 일부가 변경되었다.

- 현재 **object-c**와 혼용으로 사용가능하며 아직 **osx, ios**의 많은 라이브러리가 **object-c**기반이라 점점 **swift** 코드로 대체 중이다. **(cocoa framwork / cocoa touch framework** 기반 언어가 **object-c)**

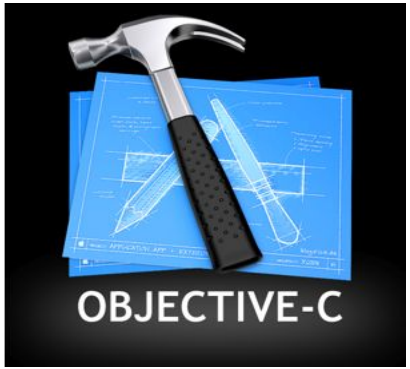
Swift 란?

- object-c와 swift의 컴파일러(백그라운드 컴파일러)인 LLVM의 메인개발자 크리스 래트너(Chris Lattner)에 의해 최초 개발되었다. 현재는 애플을 떠난 상태

- 현재 오픈소스 프로젝트로 github(https://github.com/apple/swift)에 애플이 공개한 상태이다.

- 2016.10 기준으로 3.0버전이 release상태이며, 3.0버전부터는 문법적인 큰 수정이 없다고 공식 발표하였다.

Cocoa Framework 란?



- 스티브잡스가 애플을 떠나 있을 당시 **Next**사의 메인언어인 **Object-C**를 기반으로 한 **UI** 및 전체적인 기능 프레임워크.

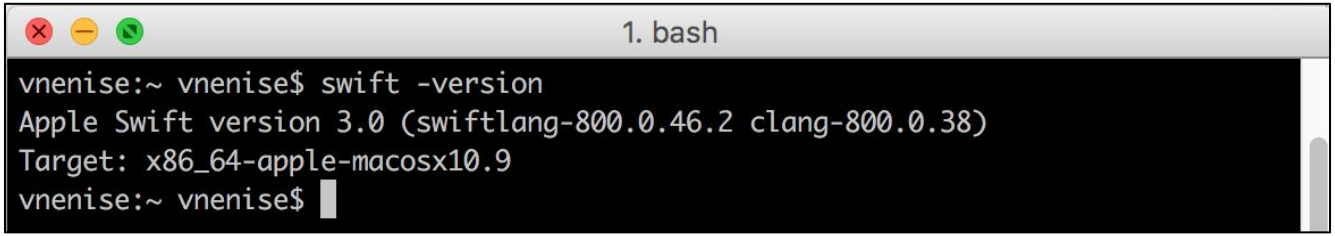
- 라이브러리 네이밍의 **NS**가 붙은 것은 당시 넥스트사의 운영체제인 넥스트스텝(NeXTSTEP)에 따온 약어

- **osx** 나 **ios** 개발시 **cooca framework**의 라이브러리를 사용하여 개발하게 됨

- **osx**는 **cocoa framework, ios/tablet**은 **cocoa touch framework** 사용

(대표적인 **Foundation Kit, UI KIt** 등이 있다.)

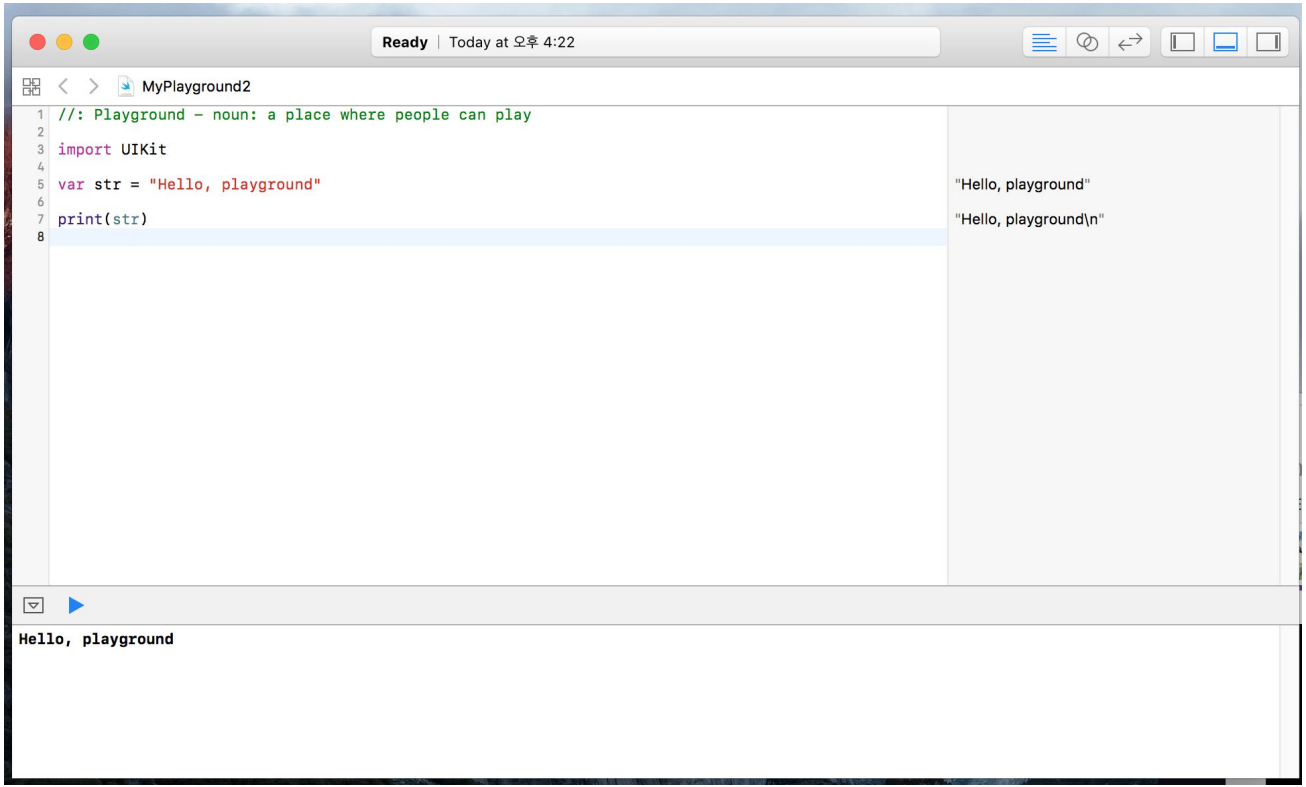
Swift 3.0 & Xcode 8.0



Xcode


Welcome to Xcode
Version 8.0 (8A218a)
Get started with a playground
Explore new ideas quickly and easily.
Create a new Xcode project
Create an app for iPhone, iPad, Mac, Apple Watch or Apple TV.
Check out an existing project
Start working on something from an SCM repository.
ImageView
~/Documents/ios/SwiftStudy/week01
Week01
~/Documents/ios/SwiftStudy
ImageView
~/Documents/ios/doitEasypub
MyPlayground
~/Documents/ios/playground
Audio
~/Documents/ios/doitEasypub
DatePicker
~/Documents/ios/doitEasypub
PickerView
~/Documents/ios/doitEasypub
Test1
~/Documents/ios/test
MeetingRooms
~/Documents/ios/helloworld
Open another project...

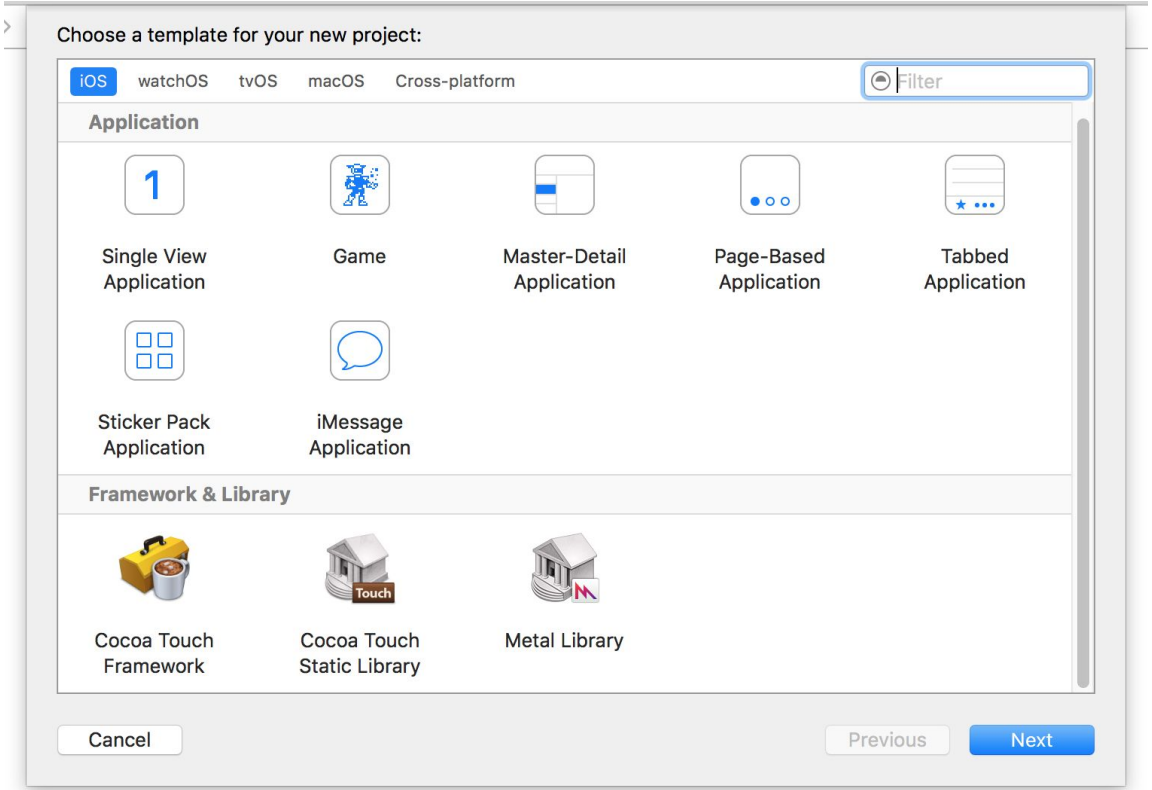
Xcode


Ready | Today at 오후 4:22
MyPlayground2
//: Playground - noun: a place where people can play
import UIKit
var str = "Hello, playground"
print(str)
"Hello, playground"
"Hello, playground\n"
Hello, playground

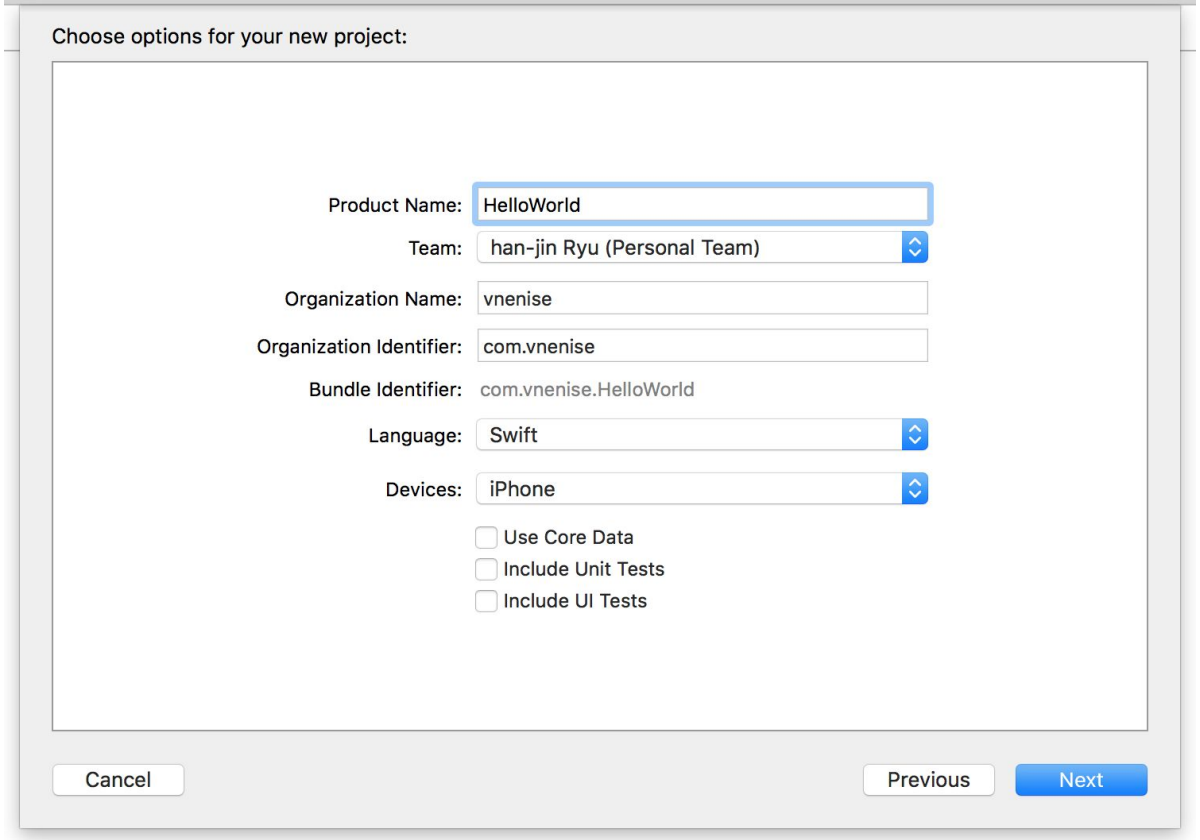
Xcode


Welcome to Xcode
Version 8.0 (8A218a)
Get started with a playground
Explore new ideas quickly and easily.
Create a new Xcode project
Create an app for iPhone, iPad, Mac, Apple Watch or Apple TV.
Check out an existing project
Start working on something from an SCM repository.
ImageView
~/Documents/ios/SwiftStudy/week01
Week01
~/Documents/ios/SwiftStudy
ImageView
~/Documents/ios/doitEasypub
MyPlayground
~/Documents/ios/playground
Audio
~/Documents/ios/doitEasypub
DatePicker
~/Documents/ios/doitEasypub
PickerView
~/Documents/ios/doitEasypub
Test1
~/Documents/ios/test
MeetingRooms
~/Documents/ios/helloworld
Open another project...

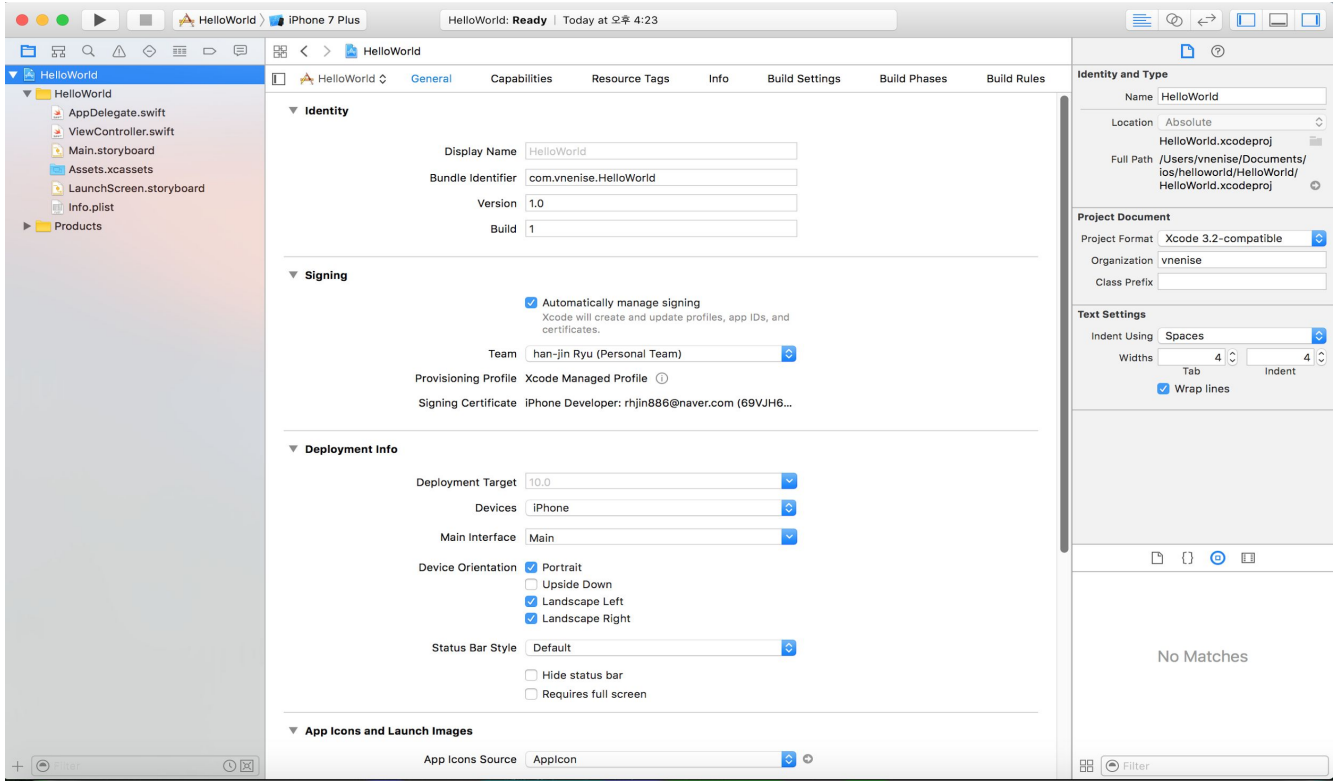
Xcode


Choose a template for your new project:
iOS
watchOS
tvOS
macOS
Cross-platform
Filter
Application
Single View Application
Game
Master-Detail Application
Page-Based Application
Tabbed Application
Sticker Pack Application
iMessage Application
Framework & Library
Cocoa Touch Framework
Cocoa Touch Static Library
Metal Library
Cancel
Previous
Next

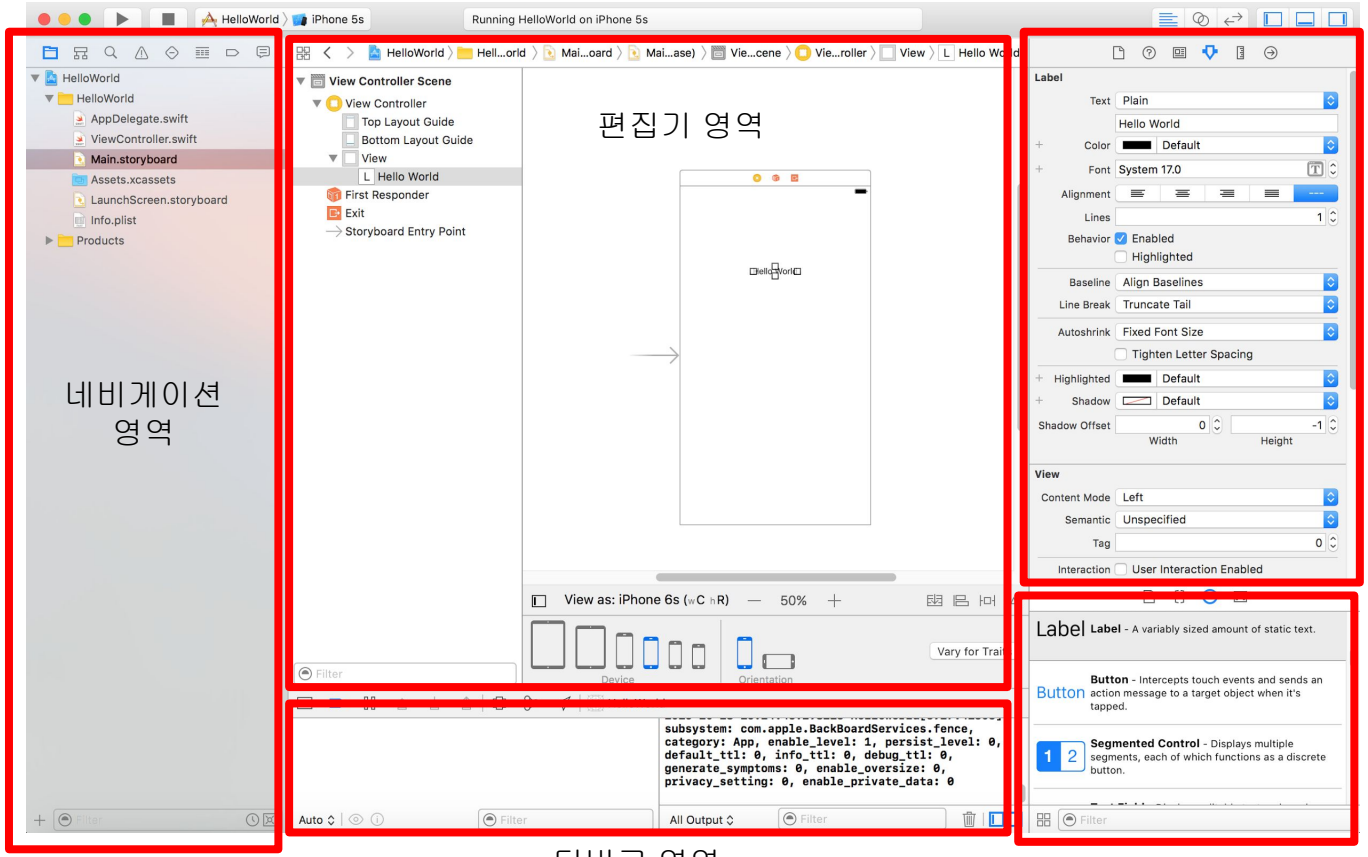
Xcode


Choose options for your new project:
Product Name: HelloWorld
Team: han-jin Ryu (Personal Team)
Organization Name: vnenise
Organization Identifier: com.vnenise
Bundle Identifier: com.vnenise.HelloWorld
Language: Swift
Devices: iPhone
Use Core Data
Include Unit Tests
Include UI Tests
Cancel
Previous
Next

Xcode


HelloWorld
iPhone 7 Plus
HelloWorld: Ready | Today at 오후 4:23
HelloWorld
HelloWorld
HelloWorld
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
HelloWorld
General
Capabilities
Resource Tags
Info
Build Settings
Build Phases
Build Rules
Identity
Display Name HelloWorld
Bundle Identifier com.vnenise.HelloWorld
Version 1.0
Build 1
Signing
Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.
Team han-jin Ryu (Personal Team)
Provisioning Profile Xcode Managed Profile
Signing Certificate iPhone Developer: rhjin886@naver.com (69VJH6...
Deployment Info
Deployment Target 10.0
Devices iPhone
Main Interface Main
Device Orientation
Portrait
Upside Down
Landscape Left
Landscape Right
Status Bar Style Default
Hide status bar
Requires full screen
App Icons and Launch Images
App Icons Source AppIcon
Identity and Type
Name HelloWorld
Location Absolute
HelloWorld.xcodeproj
Full Path /Users/vnenise/Documents/ios/helloworld/HelloWorld/HelloWorld.xcodeproj
Project Document
Project Format Xcode 3.2-compatible
Organization vnenise
Class Prefix
Text Settings
Indent Using Spaces
Widths 4 4
Tab
Indent
Wrap lines
No Matches
Filter

Xcode



Swift 문법

- 기본 자료형 (데이터 타입)
- 변수 및 선언방식
- 옵셔널 (optional)
- 조건문

Swift - 기본 자료형

타입	특징	예
Int, Int8, Int16, Int32, Int64	작은 수 또는 큰 수의 음수, 양수값	**4, 523, -45565, 5342, -28, 54, 234**
UInt, UInt8 ,UInt16, UInt32, UInt64	작은 수 또는 큰 수의 양수값	**5, 123, 3432432, 52, 34, 5, 123**
Float, Double	부동 소수점, 분수의 음수, 양수값	**11.453, 234.23, -123.34, 2.231231123, 0.012345**
Character	단일 문자 (큰따옴표로 묶어서 표현)	**“T”, “한”, “H”, “*”, “3”**
String	문자열 데이터 (큰따옴표로 묶어서 표현)	**“Filasa”, “문장입니다.”, “New York”**
Bool	참/거짓을 표현하는 논리데이터 표현	**true, false**

이 밖에도 Collection 타입: Array, Tuple, Dictionary / 구조체 Struct / 열거형 enum / Class 등 참조타입이 있음.

***값이 없다는 표현으로 swift에서는 nil 이라고 표현함.**

Swift - 변수

***변수: 프로그래밍 언어에서 일반적으로 어떤 공간에 데이터 값을 담고 사용하는 공간 또는 그릇**

var 키워드: 언제든지 값을 변경할 수 있다

<형식>
var [변수명]
var [변수명]:[데이터 자료형]

var [변수명]:[데이터 자료형]
var [변수명]:[데이터 자료형] = 데이터값

(예)
var a
var a:Int

var a = 123
a = 567 (a의 값이 567로 변경됨)

let 키워드: 값을 넣으면 영원히 그 값 유지

<형식>
let [변수명]
let [변수명] = 데이터값

let [변수명]:[데이터 자료형]
let [변수명]:[데이터 자료형] = 데이터값

(예)
let a
let a:Int

let a = 123
a = 567 (x) (값을 변경할 수 없다)

(*var는 타입 추론을 하여 알아서 데이터의 자료형을 판단합니다.)

Swift - 옵셔널

*데이터의 유무를 판별하기 위한 문법적 장치, ! 와 ?를 데이터자료형과 변수 뒤에 명시

* ?를 붙이면 실제데이터는 **Optional(데이터)** 형식으로 데이터형을 감싸게 표현됨
  - 값의 **optional** 낙인을 찍으므로, 이 데이터가 존재하는지의 여부에 대한 문법적인 경고를 준다.
  - **optional** 붙은 자료형은 **!**로 값이 존재함을 알림으로 변수/데이터를 사용하게 됩니다.

* 프로그래밍의 **null point exception**에 대한 문법적 안전장치로 **swift**에 도입되었습니다.

(유형 3가지)

```
var str:String?  //optional wrapping

var str:Int!  //force unwrapping

//implicity unwrapping
if let str2 = str {
        print(str2)
}
```

(예제)

```
var str:String?

str = "String입니다."    //  Optional("String입니다.") 형태


var str2:String = str!  // !로 str의 Optional 낙인을 지워 값을 대입
                                    // str의 값은 "String입니다." 형태
```

Swift - 조건문 if / else if / else

*** 조건의 참/거짓에 따라 해당 구문을 실행하겠금 분기구문**

1) 단독 if문: if문에 조건식이 맞으면 실행

```
if 조건식 {
  실행할 내용
}
```

2) if / else : if문 조건식이 맞지 않으면 else문을 실행

```
if 조건식 {
  실행할 내용
} else {
  실행할 내용
}
```

예제)

```
if 5 == 5 {
  print("5와 5는 같습니다.")
}
```

```
if 3 == 5 {
  print("if문 출력")
}else {
  print("else문 출력") => 3 == 5이 같지 않기에 출력
}
```

Swift - 조건문 if / else if / else

3) if / else if : if문 조건식이 맞지 않으면 else if 조건식을 판별하여 실행

```
if 조건식 {
  실행할 내용
} else if 조건 {
  실행할 내용
}
```

4) if /else if / else

```
if 조건식 {
  실행할 내용
} else if 조건 {
  실행할 내용
} else {
  실행할 내용
}
```

예제)

```
if 3== 5 {
        print(“if문 출력”)
else if 3 == 3 {
        print(“else if문 출력”)
}
```

```
if 3== 5 {
        print(“if문 출력”)
else if 3 == 1 {
        print(“else if문 출력”)
} else {
        print(“else문 출력”)
}
```