

Swift 프로그래밍

1. Swift 소개와 개발 환경

CONTENTS

1

Swift 개요

2

개발 환경

3

개발자 사이트

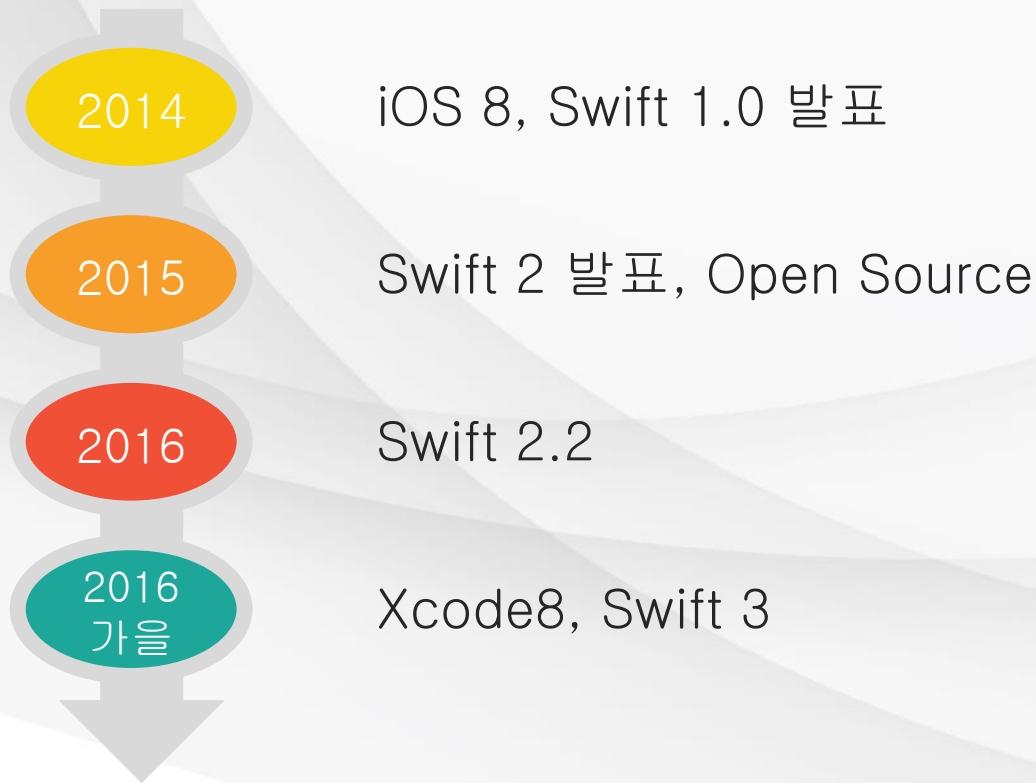
학습 목표

- Swift 언어의 등장 배경과 특징을 이해할 수 있다.
- Swift 언어 실습을 위한 개발 환경을 알아보고 세팅할 수 있다.
- 관련 사이트를 방문해보고 관련된 정보와 진행 중인 프로젝트, 진행 방향을 알 수 있다.



1. Swift 개요

Swift 언어 역사



■ Swift 언어를 사용하는 이유

❖ Objective C의 단점

- ◉ 오래된 언어 기반(C) - 헤더 파일, 구현 파일
- ◉ nil 처리에 대한 부담
- ◉ 타입 체크 약함
- ◉ 성장 한계

■ Modern, Safety, Fast and Powerful

- ◉ 클로저, 튜플, 제네릭, 함수형 프로그래밍
- ◉ **nil**에 대한 강박증에 가까운 처리
- ◉ 타입 세이프, 타입 체크 강화
- ◉ 플레이그라운드



2. 개발 환경

■ Swift 언어 실습 환경

플레이그라운드
(Playground)

iOS, MacOS
프로젝트

콘솔 기반 실행

온라인
swift 환경

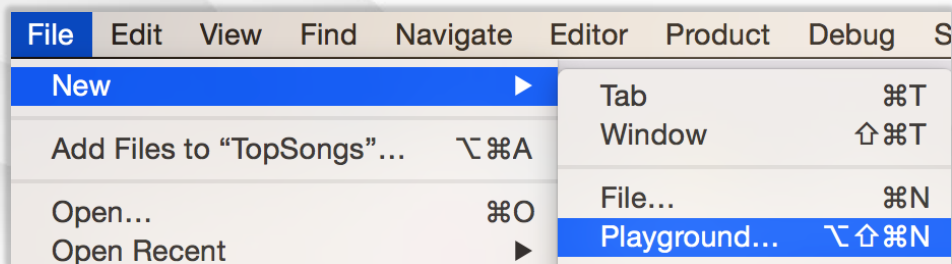
■ 플레이 그라운드

- ◉ **REPL(Read - Eval - Print - Loop)**
- ◉ 프로젝트 생성 불필요
- ◉ 코드 작성/변경 - 바로 결과 확인
- ◉ 바로 짜고 바로 확인하자

■ 플레이 그라운드 작성

웰컴 윈도우

File → New → Playground 메뉴



■ 플레이 그라운드 시작하기

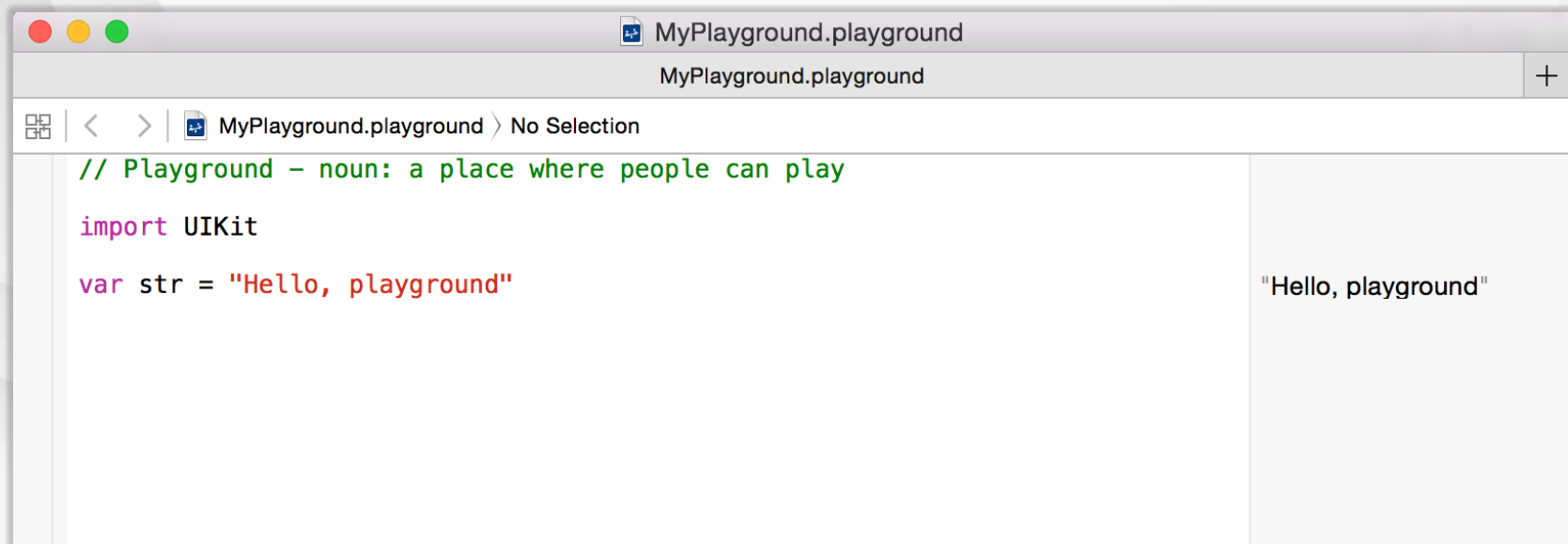
- 플레이그라운드 파일 이름
- 실행 플랫폼 선택 - **iOS, macOS**
- 플랫폼 별 동작 환경이 다르다.

Choose options for your new file:

Name

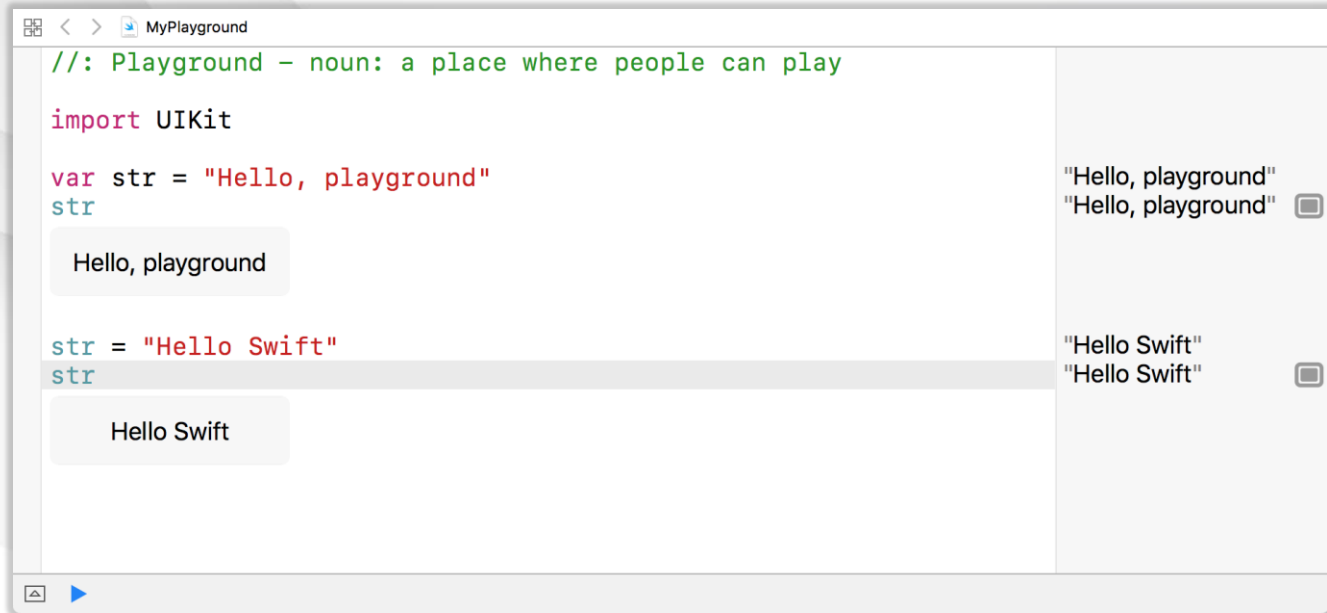
Platform:

■ 플레이 그라운드 실행 화면



플레이 그라운드 값, 결과 보기

- 에디터 내 결과 출력
- 퀵룩(Quick Look)



■ 플레이 그라운드 - 콘솔 결과 보기

- ◉ 하단 **Debug Area**
- ◉ **print** 함수 결과 출력



The screenshot shows the Swift Playground interface. The top bar displays the title 'MyPlayground'. The main editor area contains the following code:

```
str = "Hello Swift"
str

Hello Swift

print(str)
print(1+2)
print("str : \(str)")
```

Below the code editor, there is a button labeled 'Hello Swift'. To the right of the code editor, the results of the code execution are displayed:

```
"Hello Swift"
"Hello Swift"

"Hello Swift\n"
"3\n"
"str : Hello Swift\n"
```

At the bottom of the interface, the console output is shown:

```
Hello Swift
3
str : Hello Swift
```

■ 앱 프로젝트

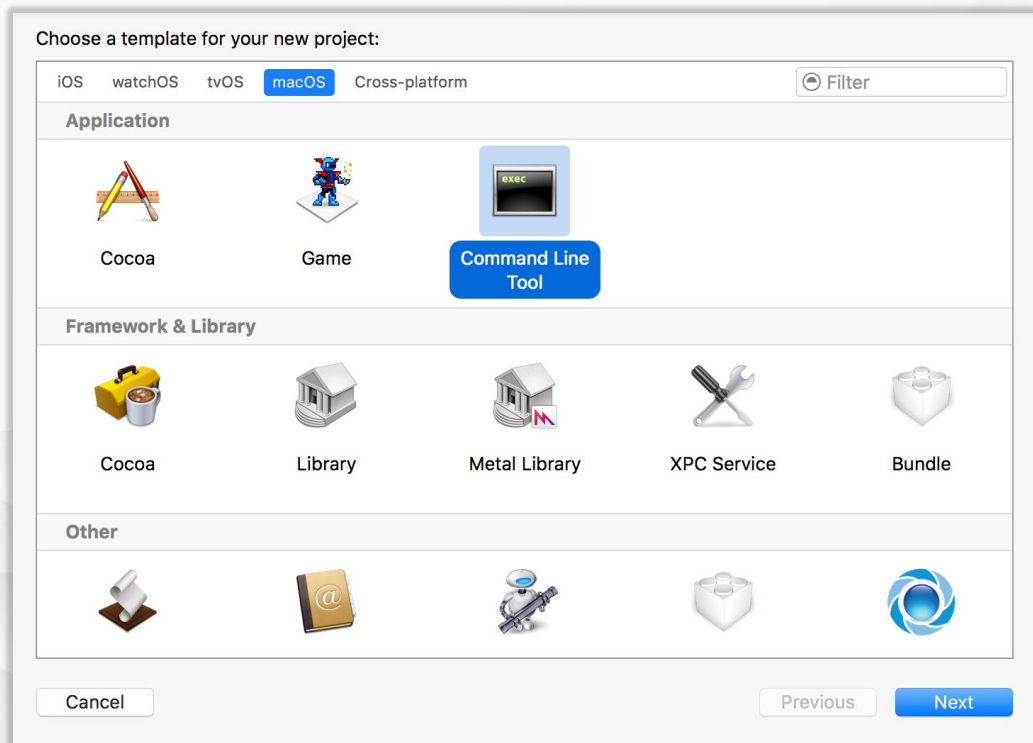
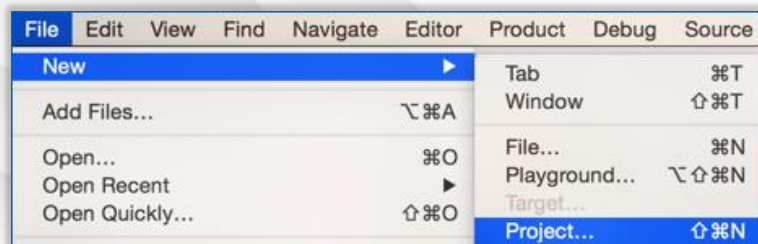
❖ Swift 언어 개발 환경

❖ 앱 개발 프로젝트

- iOS/macOS 개발 프로젝트
- 선택 환경마다 실행 환경이 조금 다름
- 개발 프로젝트 내 플레이 그라운드 포함 가능

■ 프로젝트 생성 메뉴

- 프로젝트 템플릿
- **Command Line Tool**



■ 프로젝트 정보

- 제품 이름
- 조직 이름과 정보(URL)
- **Bundle ID**
- 언어 : **Swift**

Choose options for your new project:

Product Name: HelloWorld

Team: None

Organization Name: Jaehoon Lee

Organization Identifier: com.vanillastepdev.example

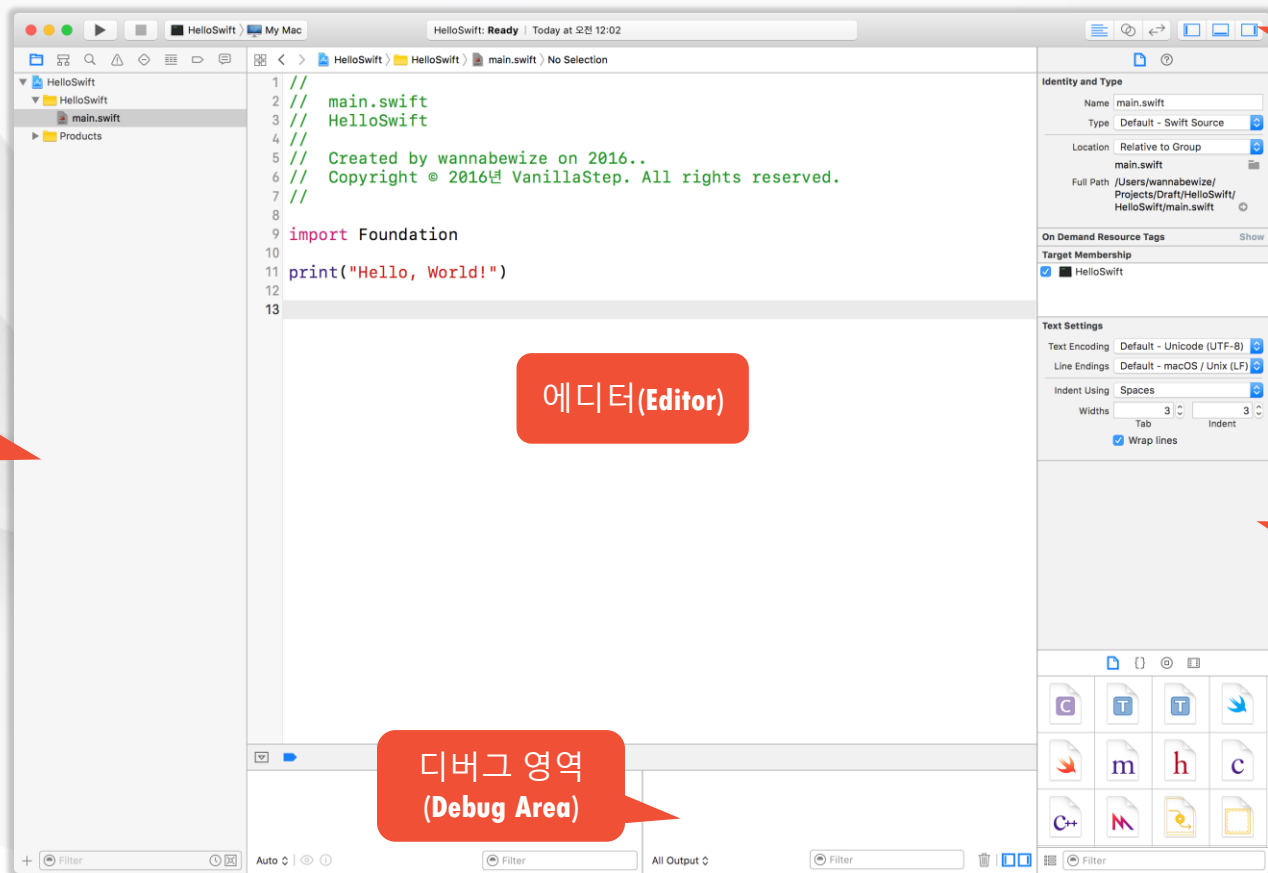
Bundle Identifier: com.vanillastepdev.example.HelloWorld

Language: Swift

Cancel Previous Next

Xcode

프로젝트 생성 완료 - Xcode



네비게이터
(Navigator)

에디터(Editor)

툴바
(Toolbar)

유tility
(Utility)

디버그 영역
(Debug Area)

■ Xcode 영역, 에디터

◉ Xcode 툴바 오른쪽 부분



◉ 영역 조절

네비게이터 영역

디버그 영역

유틸리티 영역

◉ 에디터

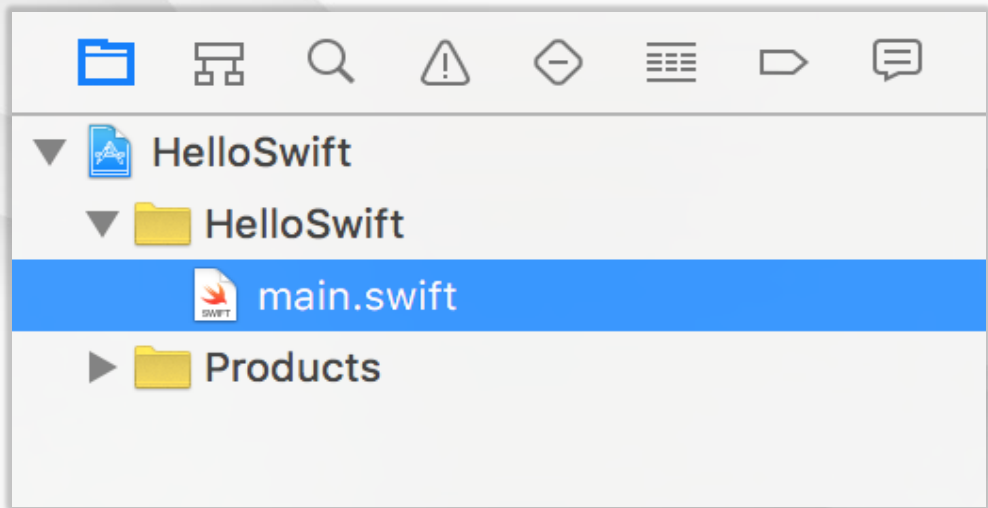
표준 모드

보조 모드

버전 모드

네비게이터 영역

- ◉ 네비게이터 영역
- ◉ 프로젝트 내 파일 목록



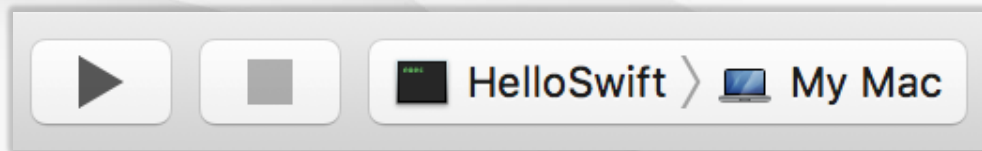
■ 코드 작성과 실행

❖ 코드 작성

```
import Foundation  
print("Hello, Swift!")
```

❖ 실행

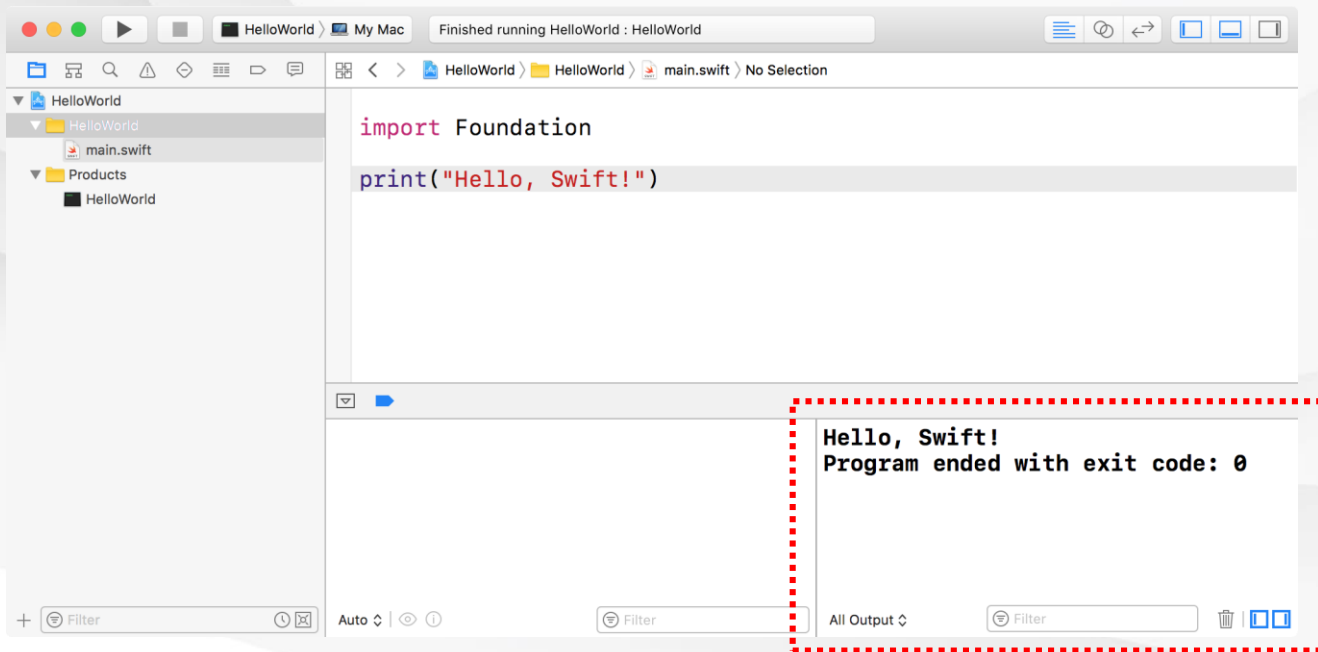
◉ Xcode 툴바 내 실행 버튼



■ 실행 결과

❖ 실행 결과 보기

👁 **print** 함수 → 디버그 영역에 출력



■ 콘솔 실행

❖ REPL 실행

\$ swift

❖ 콘솔에서 **swift** 실행

\$ swift helloWorld.swift

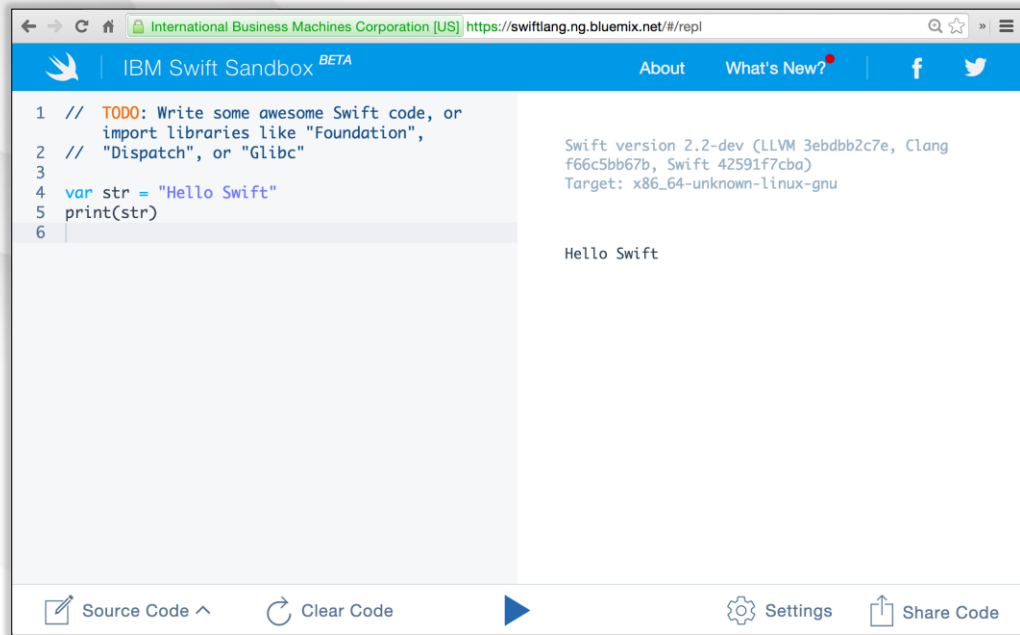


```
wannabewizeui-iMac:~ wannabewize$ swift
Welcome to Apple Swift version 2.1.1 (swiftlang-700.1.101.15 clang-700.1.81). Ty
pe :help for assistance.
1> var str = "Hello Swift"
str: String = "Hello Swift"
2> print(str)
Hello Swift
3> :help
```


■ Online Swift

❖ Online으로 실행하기

❖ ~~IBM Bluemix Swift 검색~~ online.swiftplayground.run





3. 개발자 사이트

■ 개발 지원 사이트

❖ **Swift** 개발 지원 사이트

❖ **오픈소스(swift.org)**

- ◉ 플랫폼 별 배포(현재 **ubuntu**)
- ◉ 소스 코드
- ◉ **Swift** 관련 프로젝트

❖ **애플 개발자 사이트(developer.apple.com/swift)**

- ◉ **iOS/macOS** 애플리케이션 개발 자료
- ◉ 개발자 컨퍼런스 동영상

■ swift.org

❖ 도큐먼트

❖ 다운로드

● MacOS


● Ubuntu

❖ 소스코드

❖ 프로젝트

● 패키지 매니저

● 서버

 **Swift**

ABOUT SWIFT

BLOG

DOWNLOAD

GETTING STARTED

- Installing Swift
- Using the REPL
- Using the Package Manager
- Using the LLDB Debugger

DOCUMENTATION

MIGRATING TO SWIFT 3

SOURCE CODE

COMMUNITY

CONTRIBUTING

CONTINUOUS INTEGRATION

PROJECTS

- COMPILER AND STANDARD LIBRARY
- PACKAGE MANAGER
- CORE LIBRARIES
- REPL AND DEBUGGER
- XCODE PLAYGROUND SUPPORT

Getting Started

Here, you'll find information about the how to use the Swift programming language.

If you're new to Swift, check out [A Swift Tour](#) in *The Swift Programming Language*, for a quick introduction to the most important concepts and features of the language.

Installing Swift

The first step to using Swift is to download and install the compiler and other required components. Go to the [Download](#) page and follow the instructions for your target platform.

In order to follow along with the examples below, make sure to add Swift to your `$PATH`.

On macOS

The default location for the downloadable toolchain on macOS is `/Library/Developer/Toolchains`. You can make the tools available for use from the terminal with the following command:

```
$ export PATH=/Library/Developer/Toolchains/swift-latest.xctoolchain/usr/bin:$PATH
```

On Linux

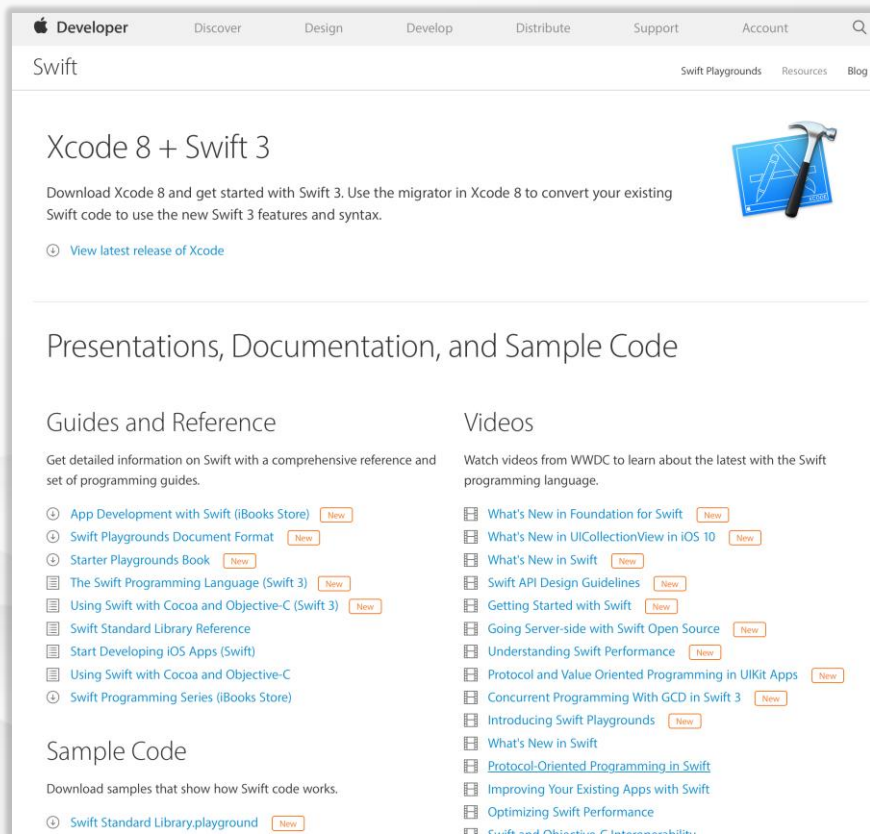
First, install clang:

```
$ sudo apt-get install clang
```

■ 애플 개발자 사이트

❖ 애플 개발자 사이트

- iOS, macOS 개발 자료
- 개발자 컨퍼런스 자료



Standard Library

❖ Swift 표준 라이브러리 문서

API Reference

Swift Standard Library > String

Structure

String

A Unicode string value.

Language

Swift

On This Page

[Overview](#)

[Nested Types](#)

[Symbols](#)

[Relationships](#)

[See Also](#)

Overview

A string is a series of characters, such as "Swift". Strings in Swift are Unicode correct, locale insensitive, and designed to be efficient. The `String` type bridges with the Objective-C class `NSString` and offers interoperability with C functions that works with strings.

You can create new strings using string literals or string interpolations. A string literal is a series of characters enclosed in quotes.

```
let greeting = "Welcome!"
```

String interpolations are string literals that evaluate any included expressions and convert the results to string form. String interpolations are an easy way to build a string from multiple pieces. Wrap each expression in a string interpolation in parentheses, prefixed by a backslash.

```
let name = "Rosa"
let personalizedGreeting = "Welcome, \(name)!"

let price = 2
let number = 3
let cookiePrice = "\(number) cookies: \$(price * number)."
```

■ Swift Evolution

❖ Swift 커뮤니티와 Swift Evolution

- ◉ Swift 언어의 개선을 위한 제안/토론/승인 과정
- ◉ Swift 3에도 많은 제안이 반영
- ◉ <https://github.com/apple/swift-evolution>
- ◉ API Guide Line - SE0023



학습정리

지금까지 [Swift 언어]에 대해서 살펴보았습니다.

Swift 언어 배경

Swift 언어는 개발 과정에서 타입 오류나 nil 오류 방지와 클로저, 제네릭스 등의 최신 언어 패러다임이 적용된 언어

개발 환경

Swift 언어를 사용하는 개발 환경으로
플레이그라운드, 앱 프로젝트, 콘솔, 온라인 코딩 사이트

개발 관련 사이트

Swift 언어를 배우는데 알아야 하는 공식 사이트와 API 문서 보기
Swift 언어 발전 커뮤니티와 과정인 Swift Revolution