

스위프트를 이용한 아이폰 앱 개발 시작하기

2020년 2월

월	화	수	목
3일 [--Swift 초급 Day!--]	4일	5일	6일 [--Swift 중급 Day!--]

```
let 초급 = "3일-4일(월-화) 오후 1시-4시"  
let 중급 = "5일-6일(수-목) 오후 1시-4시"  
  
let 장소 = "@Mac실습실 (공과대학 5호관4층 414호)"  
let 신청방법 = "https://qrgo.page.link/HETGH"
```



초급 강의 | DAY #01

<https://swift.org>

왜 스위프트인가?

영어 ▾



한국어 ▾

swift

swift



빠른

ppaleun



'**swift**'의 번역

명사

스위프트

swift

동작이 빠른 사람

swift

동작이 빠른 것

swift



Google 번역에서 열기

사용자 의견



Cocoa and Cocoa Touch

Build with LLVM compiler

Optimizer and Autovectorizer

ARC memory management

Same runtime as Objective-C



Objective-C를 대체하여 현대 언어의 요소를 추가한

클로저/ 다중리턴 타입/ 네임스페이스/
제네릭/ 타입 인터페이스 등

2014년 WWDC에서 공개

애플의 새 프로그래밍 언어

- 실행 환경: iOS / Mac OS / iPadOS
- 개발 타겟: 아이폰, 맥, 아이패드, 애플워치



Objective-C

```
// First program example
#import <Foundation/Foundation.h>
int main (int argc, const char * argv[])
{
    NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];
    NSLog(@"%@", @"Hello, World!");
    [pool drain];
    return 0;
}
```



Objective-C

Objective-C

- C를 기반으로 객체지향 형식의 Smalltalk 문법을 도입
1983년, 브래드콕스/ 톰 라브
- NeXTSTEP 운영체제의 모든 라이브러리 구현을 위해 사용됨
- 잡스가 애플에서 퇴출됐을 때, 설립한 NeXT에서 작성
- 이후 애플에 복귀할 때, 해당 라이브러리를 애플에 도입
- 애플의 기본 언어 및 라이브러리가 됨
 - NSLog, NSString, NSValue, ...

should be adopted when starting to build a new software system. The definition of the TIOBE index can be found [here](#).

<https://www.tiobe.com/tiobe-index/>

Jan 2019	Jan 2018	Change	Programming Language	Ratings	Change
1	1		Java	16.904%	+2.69%
2	2		C	13.337%	+2.30%
3	4	▲	Python	8.294%	+3.62%
4	3	▼	C++	8.158%	+2.55%
5	7	▲	Visual Basic .NET	6.459%	+3.20%
6	6		JavaScript	3.302%	-0.16%
7	5	▼	C#	3.284%	-0.47%
8	9	▲	PHP	2.680%	+0.15%
9	-	▲	SQL	2.277%	+2.28%
10	16	▲	Objective-C	1.781%	-0.08%
11	18	▲	MATLAB	1.502%	-0.15%
12	8	▼	R	1.331%	-1.22%
13	10	▼	Perl	1.225%	-1.19%
14	15	▲	Assembly language	1.196%	-0.86%
15	12	▼	Swift	1.187%	-1.19%
16	19	▲	Go	1.115%	-0.45%
17	13	▼	Delphi/Object Pascal	1.100%	-1.28%
18	11	▼	Ruby	1.097%	-1.31%
19	20	▲	PL/SQL	1.074%	-0.35%
20	14	▼	Visual Basic	1.029%	-1.28%

Jan 2020	Jan 2019	Change	Programming Language	Ratings	Change
1	1		Java	16.896%	-0.01%
2	2		C	15.773%	+2.44%
3	3		Python	9.704%	+1.41%
4	4		C++	5.574%	-2.58%
5	7	▲	C#	5.349%	+2.07%
6	5	▼	Visual Basic .NET	5.287%	-1.17%
7	6	▼	JavaScript	2.451%	-0.85%
8	8		PHP	2.405%	-0.28%
9	15	▲	Swift	1.795%	+0.61%
10	9	▼	SQL	1.504%	-0.77%
11	18	▲	Ruby	1.063%	-0.03%
12	17	▲	Delphi/Object Pascal	0.997%	-0.10%
13	10	▼	Objective-C	0.929%	-0.85%
14	16	▲	Go	0.900%	-0.22%
15	14	▼	Assembly language	0.877%	-0.32%
16	20	▲	Visual Basic	0.831%	-0.20%
17	25	▲	D	0.825%	+0.25%
18	12	▼	R	0.808%	-0.52%
19	13	▼	Perl	0.746%	-0.48%
20	11	▼	MATLAB	0.737%	-0.76%



6406 open source UI components for iOS, iPadOS, macOS, tvOS, and watchOS.

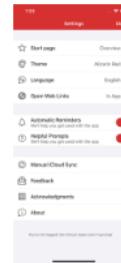
Filter Platform: All ▾ Language: All ▾ Language: All ▾



react-native-wowza-gocoder

React Native wrapper for the Wowza GoCoder SDK

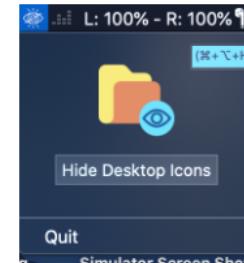
January 30, 2020 • MIT License



ToDoList

A simple ToDoList written in Swift 5. Supports syncing using CloudKit, Realm, Local not...

January 21, 2020 • Apache 2.0 License



HideAndSeek Desktop

Hide/Show your Desktop Icons on your Mac Easily with one

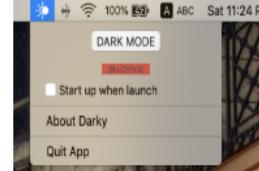
January 14, 2020 • MIT License



R.swift

Get strong typed, autocompleted resources like

January 29, 2020 • MIT License



Darky

Switch from Light Mode to Dark Mode on Mac OS

January 20, 2020 • MIT License



MDatePickerView

Quick and easy date picker.

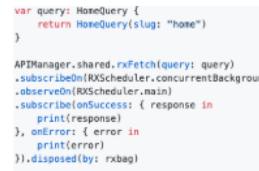
January 13, 2020 • MIT License



EMTNeumorphicView

UIKit views with Neumorphism style design.

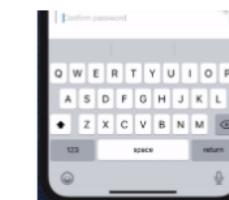
January 28, 2020 • MIT License



RRApolloRxAPIClient

GraphQL (Apollo) API Request by RxSwift

January 19, 2020 • MIT License



KeyboardAvoider

The missing interactive keyboard in SwiftUI for iOS

January 28, 2020 • MIT License



Swift

Swift Programming Language

	Architecture	Master	Package
macOS	x86_64	build passing	build failing
Ubuntu 16.04	x86_64	build passing	build passing
Ubuntu 18.04	x86_64	build passing	build passing

Swift Community-Hosted CI Platforms

OS	Architecture	Build
Ubuntu 16.04	PPC64LE	build failing
Ubuntu 16.04	AArch64	build failing
Android	ARMv7	build passing
Android	AArch64	build passing
Windows 2019	x86_64	build passing

Swift TensorFlow Community-Hosted CI Platforms

OS	Architecture	Build
Ubuntu 16.04	x86_64	build passing
macOS 10.13	x86_64	build failing
Ubuntu 16.04 (GPU)	x86_64	build failing

COLLECTION

Programming languages

A list of programming languages that are actively developed on GitHub

[Suggest edits](#)



The Swift Programming Language

★ 46475 7357 C++



<https://github.com/collections/programming-languages>



ABOUT SWIFT

BLOG

DOWNLOAD

GETTING STARTED

DOCUMENTATION

SOURCE CODE

COMMUNITY

CONTRIBUTING

CONTINUOUS
INTEGRATION

SOURCE
COMPATIBILITY

FOCUS AREAS

SERVER WORK
GROUP

PROJECTS

COMPILER AND
STANDARD LIBRARY

PACKAGE MANAGER

CORE LIBRARIES

REPL, DEBUGGER &
PLAYGROUNDS

Welcome to Swift.org

Welcome to the Swift community. Together we are working to build a programming language to empower everyone to turn their ideas into apps on any platform.

Announced in 2014, the Swift programming language has quickly become one of the fastest growing languages in history. Swift makes it easy to write software that is incredibly fast and safe by design. Our goals for Swift are ambitious: we want to make programming simple things easy, and difficult things possible.

For students, learning Swift has been a great introduction to modern programming concepts and best practices. And because it is open, their Swift skills will be able to be applied to an even broader range of platforms, from mobile devices to the desktop to the cloud.

Swift History

1.0 (Sep 9, 2014)

~

5.1 (Sep 10, 2019)

Version	Release Date	macOS	Linux
Swift 1.0	September 9, 2014	Yes	No
Swift 1.1	October 22, 2014	Yes	No
Swift 1.2	April 8, 2015	Yes	No
Swift 2.0	September 21, 2015	Yes	No
Swift 2.1	October 20, 2015	Yes	No
Swift 2.2	March 21, 2016	Yes	Yes
Swift 2.2.1	May 3, 2016	Yes	Yes
Swift 3.0	September 13, 2016	Yes	Yes
Swift 3.0.1	October 28, 2016	Yes	Yes
Swift 3.0.2	December 13, 2016	Yes	Yes
Swift 3.1	March 27, 2017	Yes	Yes
Swift 3.1.1	April 21, 2017	Yes	Yes
Swift 4.0	September 19, 2017	Yes	Yes
Swift 4.0.2	November 1, 2017	Yes	Yes
Swift 4.0.3	December 5, 2017	Yes	Yes
Swift 4.1	March 29, 2018	Yes	Yes
Swift 4.1.1	May 4, 2018	No	Yes
Swift 4.1.2	May 31, 2018	Yes	Yes
Swift 4.1.3	July 27, 2018	No	Yes
Swift 4.2	September 17, 2018	Yes	Yes
Swift 4.2.1	October 30, 2018	Yes	Yes
Swift 4.2.2	February 4, 2019	No	Yes
Swift 4.2.3	February 28, 2019	No	Yes
Swift 4.2.4	March 29, 2019	No	Yes
Swift 5.0 ^[44]	March 25, 2019	Yes	Yes
Swift 5.0.1	April 18, 2019	Yes	Yes
Swift 5.0.2	July 15, 2019	No	Yes
Swift 5.0.3	August 30, 2019	No	Yes
Swift 5.1	September 10, 2019	Yes	Yes

← 스위프트가 세상에 출시한 날

← xCode 8

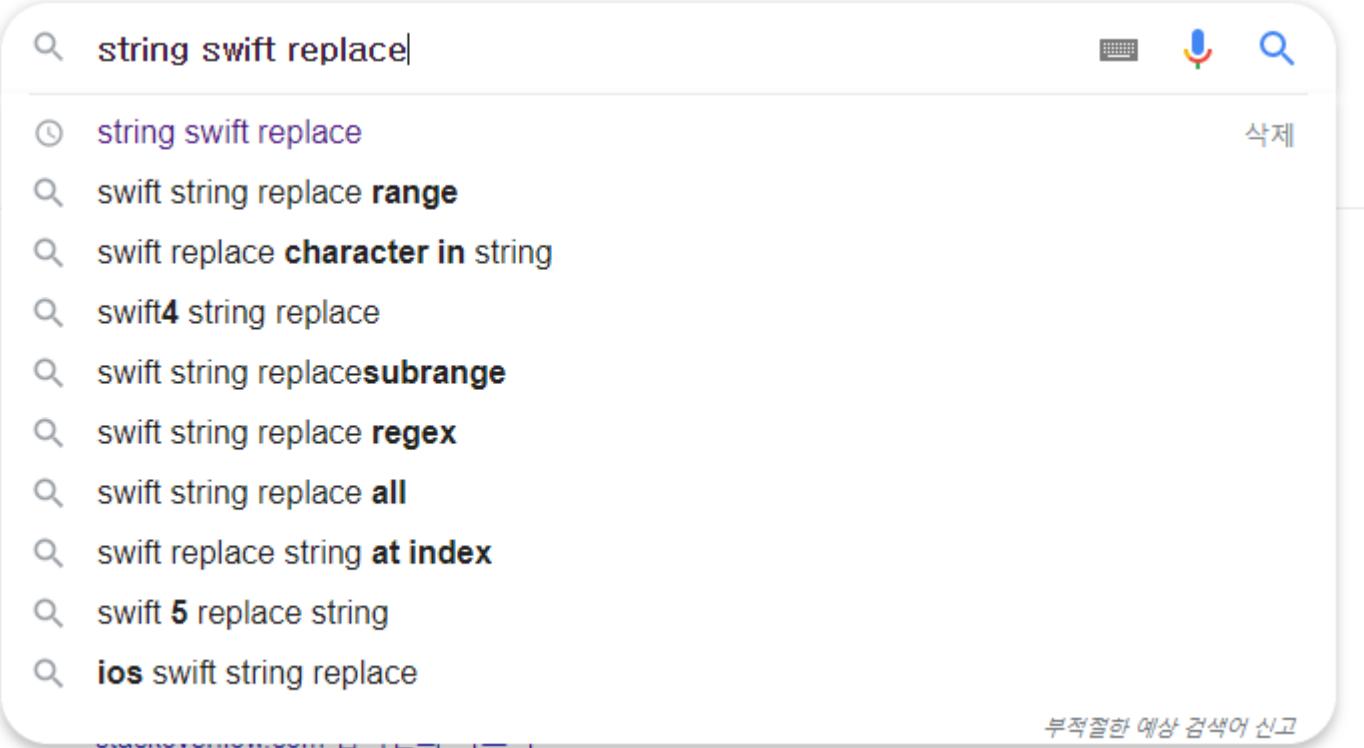
← xCode 9

← xCode 10, 3.0, 4.1, 4.0 버전 코드 호환

← xCode 10.2, 하위 호환 불가

← xCode 11

버전별로 얼마나 다른가?



A screenshot of a Google search bar showing suggestions for the query "string swift replace". The suggestions are listed below the search bar, each preceded by a magnifying glass icon. The suggestions are:

- string swift replace
- string swift replace
- swift string replace **range**
- swift replace **character** in string
- swift4 string replace
- swift string replacesubrange
- swift string replace **regex**
- swift string replace **all**
- swift replace string **at index**
- swift 5 replace string
- ios swift string replace

At the bottom right of the suggestions, there is a link "부적절한 예상 검색어 신고" (Report inappropriate suggested search terms) and a small note "부적절한 예상 검색어 신고" (Report inappropriate suggested search terms).

You can use this:

64
let s = "This is my string"
let modified = s.replace(" ", withString:"+")

If you add this extension method anywhere in your code:

```
extension String  
{  
    func replace(target: String, withString: String) -> String  
    {  
        return self.stringByReplacingOccurrencesOfString(target, withString: withString, options:  
    }  
}
```

Swift 3:

```
extension String  
{  
    func replace(target: String, withString: String) -> String  
    {  
        return self.replacingOccurrences(of: target, with: withString, options: NSString.CompareOptions.literal, range: nil)  
    }  
}
```

Swift 4:

13
let abc = "Hello world"
let result = abc.replacingOccurrences(of: " ", with: "_",
options: NSString.CompareOptions.literal, range: nil)
print(result :\\(result))

Output:

```
result : Hello_world
```

share improve this answer

edited Jan 10 '18 at 6:06

clemens
12.4k ● 11 ● 29 ● 46

answered Jan 10 '18 at 5:47

Garine
249 ● 2 ● 8

add a comment

13 Answers

active oldest votes

Swift 4.2

204

```
var urlString = originalString.addingPercentEncoding(withAllowedCharacters: .urlQueryAllowed)
```

Swift 3.0

45

```
var address = "American Tourister, Abids Road, Bogulkunta, Hyderabad, Andhra Pradesh, India"
let escapedAddress = address.addingPercentEncoding(withAllowedCharacters: CharacterSet.urlQueryAllowed)
let urlpath = String(format: "http://maps.googleapis.com/maps/api/geocode/json?address=%@(escap
```

Use `stringByAddingPercentEncodingWithAllowedCharacters`:

45

```
var escapedAddress = address.stringByAddingPercentEncodingWithAllowedCharacters(NSCharacterSet
```

Use `stringByAddingPercentEscapesUsingEncoding`: Deprecated in iOS 9 and OS X v10.11

45

```
var address = "American Tourister, Abids Road, Bogulkunta, Hyderabad, Andhra Pradesh, India"
var escapedAddress = address.stringByAddingPercentEscapesUsingEncoding(NSUTF8StringEncoding)
let urlpath = NSString(format: "http://maps.googleapis.com/maps/api/geocode/json?address=%@(esc
```



Products Customers Use cases Search ...

Log in Sign up

How to encode a URL in Swift

Asked 5 years, 6 months ago Active 6 months ago Viewed 119k times

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

TAGS

What's best?

Free 30 Day Trial

This is my URL.

127

The problem is, that the `_address` field is not being appended to `_urlpath`.

Does anyone know why that is?

```
var address:String
address = "American Tourister, Abids Road, Bogulkunta, Hyderabad, Andhra Pradesh, India"
let urlpath = NSString(format: "http://maps.googleapis.com/maps/api/geocode/json?address=%@(escap
```

Xcode Swift urlencode

share Improve this question

edited Jul 22 '14 at 9:39

Dmitriy

2,842 4 15 +38

wanted Jul 22 '14 at 9:39

User2541487

1,373 3 9 +6

```
2 let urlpath = "http://maps.googleapis.com/maps/api/geocode/json?address=%@(address)": is
enough! guess - jharroy Jul 22 '14 at 5:45 ✓
2 "Some string" + "(aString)" is overkill... Simply use "Some string (aString)" - n071590 Mar 19 '15 at 10:10 ✓
2 Or just "some string" + aString - Guotatu Sep 17 '15 at 13:55
We can use stringByAddingPercentEscapesUsingEncoding method. You can find more
details in this ANSWER: stackoverflow.com/a/44043893/4061593 - Lal Krishna Aug 8 '16 at 8:03
```

add a comment

13 Answers

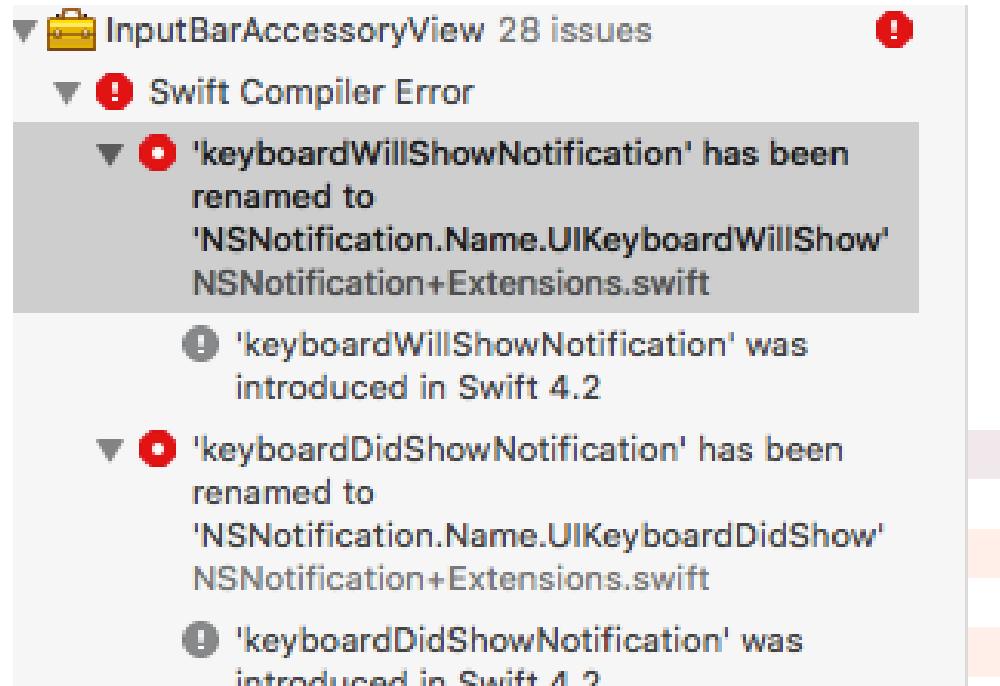
active oldest votes

Swift 4.2

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

그래도 괜찮다.

컴파일러가 도와주기 때문



맥 컴퓨터 사용법



Mac에서 Multi-Touch 제스처 사용하기

Multi-Touch 트랙패드 또는 Magic Mouse를 사용하는 경우 하나 이상의 손가락을 이용한 탭하기, 쓸어넘기기, 오므리기, 펼치기 동작으로 유용한 작업을 수행할 수 있습니다.

트랙패드 제스처

트랙패드 제스처에 대한 자세한 내용을 확인하려면 Apple 메뉴(🍎) > 시스템 환경설정을 선택한 다음 '트랙패드'를 클릭합니다. 제스처를 끄거나 제스처 유형을 변경할 수 있으며 Mac에서 사용할 수 있는 제스처에 대해 알아볼 수 있습니다.

트랙패드 제스처를 사용하려면 [Magic Trackpad](#) 또는 내장 Multi-Touch 트랙패드가 있어야 합니다. 트랙패드에서 Force Touch를 지원하는 경우 [세계 클릭하고 햅틱 피드백을 받을 수도 있습니다.](#)



탭하여 클릭하기

한 손가락으로 탭하여 클릭합니다.



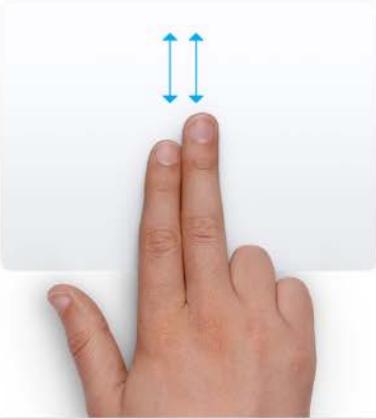
보조 클릭하기(오른쪽 클릭하기)

두 손가락으로 클릭하거나 탭합니다.



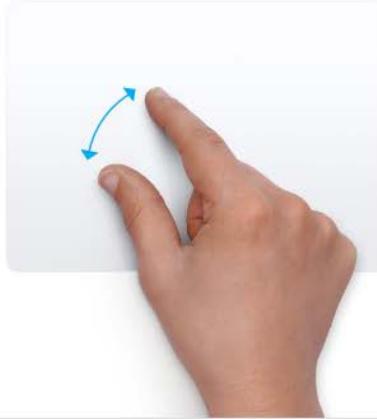
스마트 확대/축소하기

두 손가락으로 이중 탭하여 웹 페이지 또는 PDF를 확대하거나 원래대로 축소합니다.



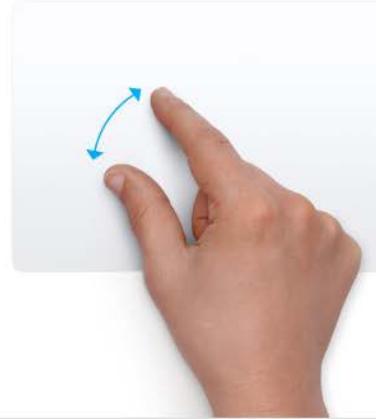
스크롤하기

두 손가락을 위 또는 아래로 밀어
스크롤합니다.¹



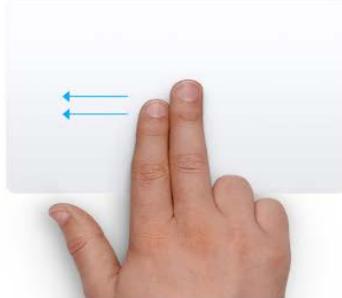
확대 또는 축소하기

두 손가락을 펼치거나 오므려서
확대하거나 축소합니다.



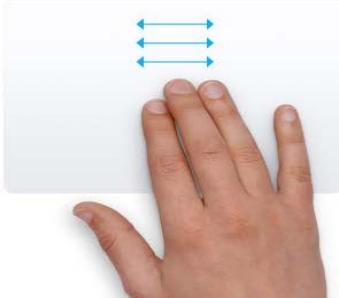
회전하기

두 손가락을 같이 둥글게 움직여 사진이나
기타 항목을 회전합니다.



알림 센터 열기

두 손가락으로 오른쪽 가장자리에서
왼쪽으로 쓸어넘겨 알림 센터를
표시합니다.



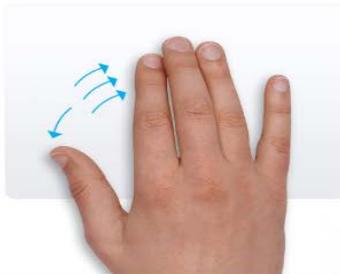
세 손가락으로 드래그하기

세 손가락으로 화면의 항목을 드래그한
다음 클릭하거나 탭하여
드롭합니다. [손쉬운 사용 환경설정](#)에서 이
기능을 켤 수 있습니다.²



찾아보기 및 데이터 탑색기

세 손가락으로 탭하여 단어를 찾아보거나
날짜, 주소, 전화번호 및 기타 데이터에
대한 작업을 수행합니다.



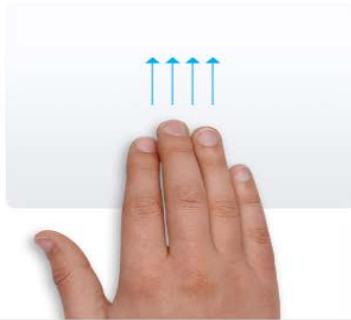
데스크탑 보기

엄지와 세 손가락을 서로 다른 방향으로
펼쳐 데스크탑을 표시합니다.



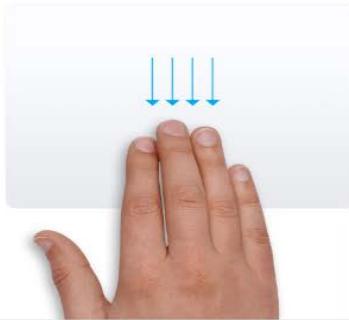
Launchpad

엄지와 세 손가락을 함께 오므려서 Launchpad를 표시합니다.



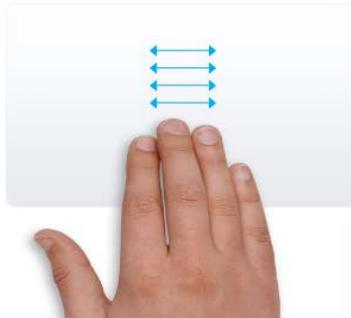
Mission Control

네 손가락으로 위로 쓸어올려³ Mission Control을 엽니다.



Expose

네 손가락으로 아래로 쓸어내려³ 사용 중인 앱의 윈도우를 모두 표시합니다.



전체 화면 앱 쓸어넘기기

네 손가락으로 왼쪽 또는 오른쪽으로 쓸어넘겨³ 데스크탑과 전체 화면 앱 간에 이동합니다.



<https://support.apple.com/ko-kr/HT201236>

Mac 키보드 단축키

보통은 마우스, 트랙패드 또는 기타 입력 기기가 있어야 가능한 작업을 특정 키 조합을 눌러 수행할 수 있습니다.

키보드 단축키를 사용하려면 하나 이상의 조합 키를 누른 상태에서 단축키의 마지막 키를 누릅니다. 예를 들어 command-C(복사)를 사용하려면 command 키를 누른 상태에서 C 키를 누른 다음 두 키를 놓습니다. Mac 메뉴와 키보드에서는 다음과 같이 조합 키를 비롯하여 [특정 키가 기호로 표시되기도 합니다.](#)

- command(또는 cmd) ⌘
- shift ⇧
- option(또는 alt) ⌥
- control(또는 ctrl) ^
- caps lock ⇤
- fn

Windows PC용 키보드에서는 option 대신 Alt 키, command 대신 Windows 로고 키를 사용합니다.

일부 Apple 키보드에는 디스플레이 밝기 ☀, 키보드 밝기 ✎, Mission Control 등 특수 기호 및 기능을 가진 키가 있습니다. 키보드에 이러한 기능 키가 없는 경우 [자신만의 키보드 단축키를 생성](#)하여 일부 기능을 재현할 수 있습니다. 이러한 키를 F1, F2, F3 또는 기타 표준 기능 키로 사용하려면 fn 키와 함께 사용합니다.

오려두기, 복사하기, 붙여 넣기 및 기타 일반적으로 사용하는 단축키

- **command-X**: 선택한 항목을 잘라내어 클립보드에 복사합니다.
- **command-C**: 선택한 항목을 클립보드에 복사합니다. 이 키는 Finder의 파일에도 작동합니다.
- **command-V**: 클립보드의 콘텐츠를 현재 문서 또는 앱에 붙여 넣습니다. 이 키는 Finder의 파일에도 작동합니다.
- **command-Z**: 이전 명령을 실행 취소합니다. 그런 다음 shift-command-Z 키를 눌러 실행 복귀하면 실행 취소 명령을 되돌릴 수 있습니다. 일부 앱에서는 여러 명령을 실행 취소하고 실행 복귀할 수 있습니다.
- **command-A**: 모든 항목을 선택합니다.
- **command-F**: 문서에서 항목을 찾거나 찾기 윈도우를 엽니다.
- **command-G**: 다시 찾기: 이전에 찾은 항목이 다음으로 나타나는 위치를 찾습니다. 이전에 나타난 위치를 찾으려면 shift-command-G 키를 누릅니다.
- **command-H**: 전면에 있는 앱의 윈도우를 가립니다. 전면에 있는 앱을 제외한 나머지 앱을 모두 가리려면 option-command-H 키를 누릅니다.
- **command-M**: 전면에 있는 윈도우를 최소화하여 Dock에 추가합니다. 전면에 있는 앱의 윈도우를 모두 최소화하려면 command-option-M 키를 누릅니다.
- **command-O**: 선택한 항목을 열거나 열려는 파일을 선택할 수 있는 대화상자를 엽니다.
- **command-P**: 현재 문서를 프린트합니다.
- **command-S**: 현재 문서를 저장합니다.
- **command-T**: 새 탭을 엽니다.
- **command-W**: 전면에 있는 윈도우를 닫습니다. 앱의 윈도우를 모두 닫으려면 option-command-W 키를 누릅니다.
- **option-command-esc**: 앱을 강제 종료합니다.

Swift

기본 문법

문장의 끝;

Swift

```
print("Hello")
let str = "Hello World"
func doHelloWorld() -> Int
```

```
NSLog(@"Hello");
NSString * str = @"Hello World";
- (int) doHelloWorld;
```

Objective-C

변수와 상수 선언

Swift

```
var num = 0  
let _ratio2 = 123.45  
let 한글상수 = “유니코드”
```

```
int num = 0;  
const double _ratio2 = 123.45;
```

Objective-C

자료형 & 형식추론

Swift

```
let heightAndWeight = (180.0, 72)
func loadHtml( ) -> (code: Int, msg: String) {
    return (200, "OK")
}
let str = "string"
let str2: String = "string"
```

```
NSInteger number = 5;
int count = 0;
```

```
NSString * str = @“Hello”;
```

Objective-C

문자와 문자열

Swift

```
let str = "Swift String"  
let ch: Character = "a"  
print("value of ch: \(ch)")
```

```
char * cStr = "C String";  
NSString * str = @“Objective-C String”;  
char ch = 'a';  
NSLog(@“value of ch: %c”, ch);
```

Objective-C

예외처리

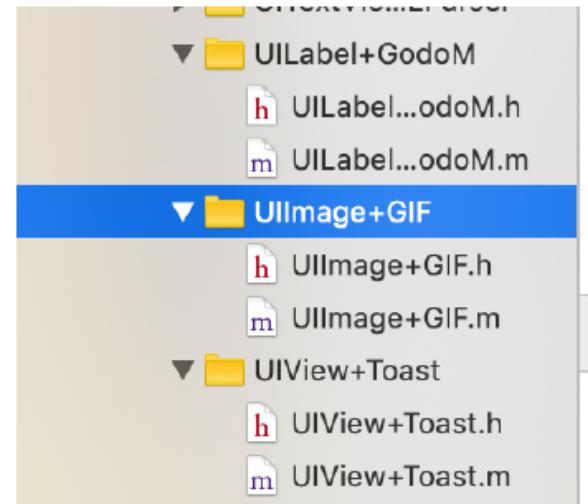
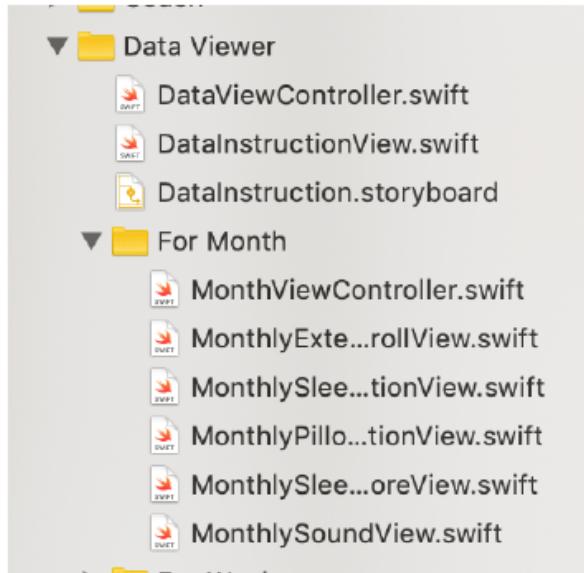
Swift

```
func trySomething() {  
    do {  
        try dSomething()  
    } catch {  
        print(error)  
        return  
    }  
}
```

소스파일

Swift

.swift
(1 file)



Objective-C

.m .h
2 files fair

Welcome to Xcode



Welcome to Xcode

Version 9.4 (9F1027a)

-  **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.
-  **Clone an existing project**
Start working on something from an SCM repository.

Show this window when Xcode launches

[Open another project...](#)

Recently
Opened
Projects
Area

Xcode #1 Playground

Swift Playground



Welcome to Xcode

Version 9.4 (9F1027a)

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.

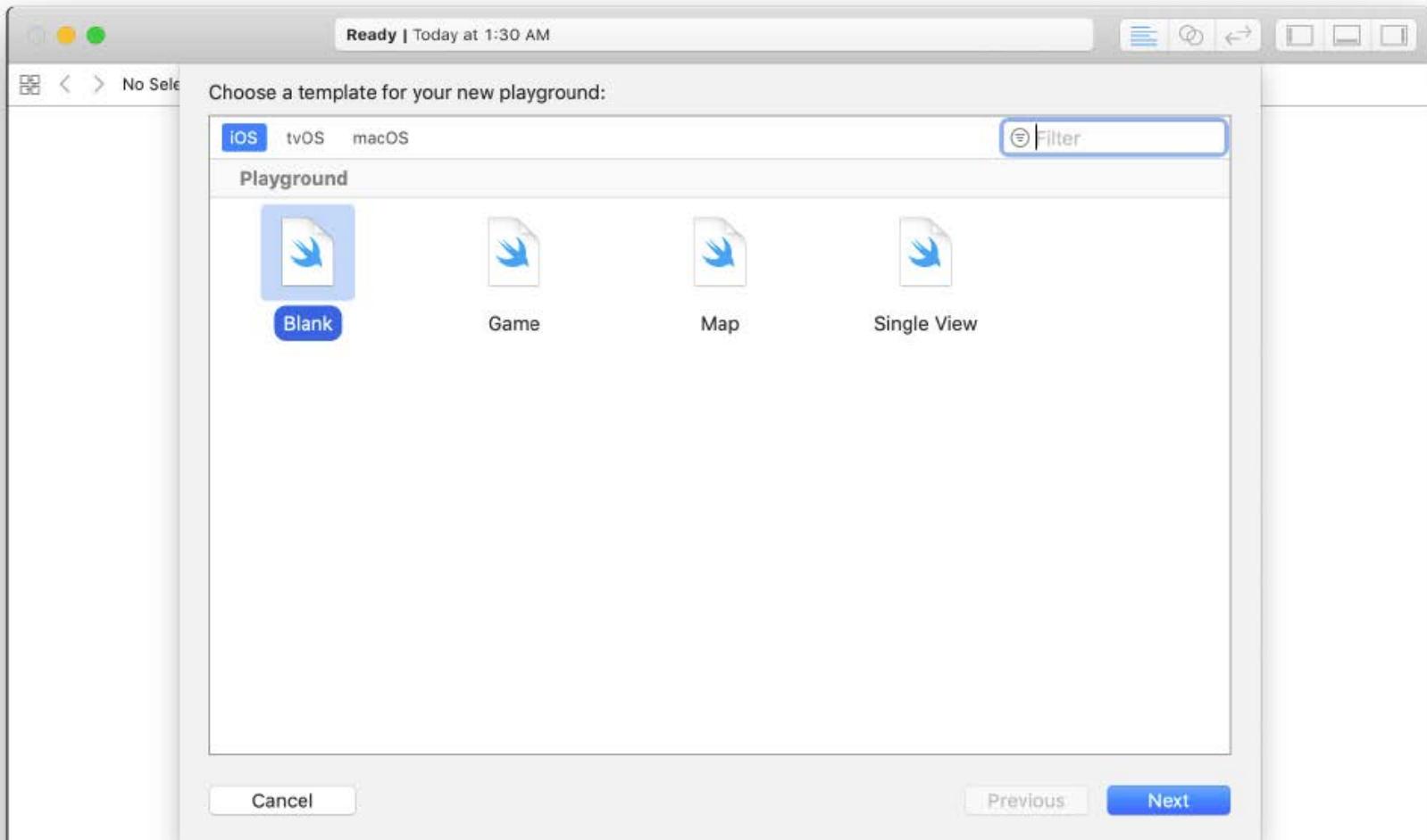
Clone an existing project
Start working on something from an SCM repository.

Show this window when Xcode launches

Open another project...

Recently Opened Projects Area

Swift Playground



The image shows a screenshot of an Xcode playground window titled "Playground#1". The main area is labeled "Source Editor" and contains the following Swift code:

```
1 //: Playground - noun: a place where people can play
2
3 import UIKit
4
5 var str = "Hello, playground"
6
7 print("Test is \(str)")
8
```

To the right, the "Results Sidebar" displays two output snippets:

- "Hello, playground" (with a small screenshot icon)
- "Test is Hello, playground" (with a small screenshot icon)

At the bottom left, the "Debug Area" shows the output "Test is Hello, playground". A callout arrow points from the text "show/hide the Debug Area Button" to the disclosure triangle icon in the toolbar. Another callout arrow points from the text "Run Button" to the play button icon in the toolbar.

Source Editor

Results Sidebar

show/hide the Debug Area Button

Run Button

Test is Hello, playground

Debug Area

With Some Errors..

The screenshot shows an Xcode playground window titled "Playground#1". The code in the editor is:

```
//: Playground - noun: a place where people can play
import UIKit
var str = "Hello, playground"
for i in 0..<3 {
    print("loop count \(i)")
}
print("Test is \(str)")
```

A red error highlight covers the line "print("loop count \(i)")". A tooltip above the highlight says "Expected ')' in expression list". An arrow points from the Korean text "에러 위치 표시" (Error location indicator) to the opening parenthesis "(" in the highlighted line.

"Hello, playground
"Hello, playground
(3 times)
"Test is Hello, playground

에러 위치 표시

print("loop count \(i)")
^

E.g. #1 - for Loop

The screenshot shows a Mac OS X window titled "Playground#1" with the status bar indicating "Ready | Today at 1:40 AM". The playground code is as follows:

```
1 //: Playground - noun: a place where people can play
2
3 import UIKit
4
5 var str = "Hello, playground"
6
7 for i in 0..<3 {
8     print("loop count \(i)")
9 }
10
11 print("Test is \(str)")
12
```

The output pane displays the results of the code execution:

- "Hello, playground..." (3 times)
- "Test is Hello, playground"

The bottom pane shows the raw command-line output:

```
loop count 0
loop count 1
loop count 2
Test is Hello, playground
```

E.g. #2 - Visualized

The screenshot shows an Xcode playground window titled "Playground#1". The code in the editor is:

```
//: Playground - noun: a place where people can play
import UIKit
var str = "Hello, playground"
for i in 0..<100 {
    //print("loop count \(i)")
    i
}
print("Test is \(str)")
```

The line `i` in the loop is highlighted with a blue box. To the right of the editor, a visual representation of the loop's execution is shown: a series of blue dots forming a curve that starts low and rises steeply towards the top right. An annotation with an arrow points to this curve, labeled "(100 times)". Below this, the Korean text "시각화 버튼" (Visualization button) is written, with another arrow pointing to a small blue square icon.

A screenshot of an Xcode playground window titled "Playground#1". The code editor contains the following Swift code:

```
//: Playground
import UIKit
var str = "Hello, playground"
for i in 0..<5 {
    //print(i)
    i
}
print("Test is here")
```

The line `i` at the end of the first `for` loop iteration is highlighted with a blue selection rectangle. A dotted blue line extends from this selection rectangle towards the bottom right corner of the editor area, indicating a copy operation.



Test is Hello, playground

Ready | Today at 1:44 AM

Playground#1

```
1 //: Playground - noun: a place where people can play
2
3 import UIKit
4
5 var str = "Hello, playground"
6
7 for i in 0..<100 {
8     sin(Double(i) / 2)
9 }
10
```

"Hello, playground..."
(100 times)

Test is Hello, playground

Ready | Today at 1:55 AM

```
3 import UIKit
4
5 var str = "Hello, playground"
6
7 let triple:(Int) -> Int = {
8     (number:Int) in
9     let result = 3 * number
10    number
11    return result
12 }
13 triple(3)
14
15 let listOfNumbers = 1...5
16 var sum = 0
17 for n in listOfNumbers {
18     sum += n
19 }
20
21 var j = 2
22 for i in 0..<5 {
23     j * i
24 }
25
```

"Hello, playgro..." (Int) -> Int 9 3 9 9 {lowerBound...} 0 (5 times) 2 (5 times)



E.g. #3- LiveView

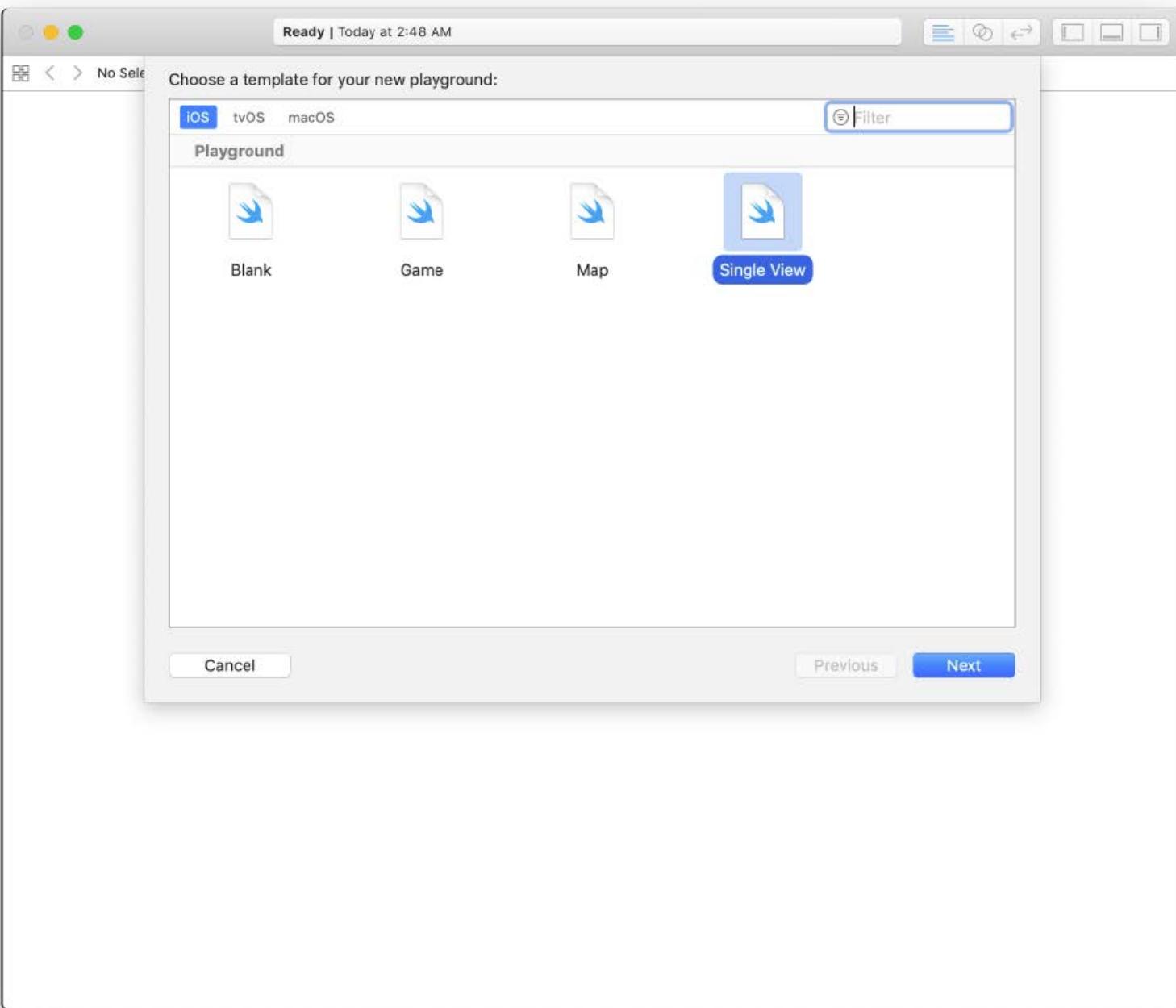
The image shows a screenshot of an Xcode playground window titled "MyPlayground". The title bar also indicates "Running MyPlayground" and "Live View". The main area is divided into two sections: the "Source Editor" on the left and the "Single View Area" on the right.

Source Editor:

```
1 //: A UIKit based Playground for presenting user interface
2
3 import UIKit
4 import PlaygroundSupport
5
6 class MyViewController : UIViewController {
7     override func loadView() {
8         let view = UIView()
9         view.backgroundColor = .white
10
11        let label = UILabel()
12        label.frame = CGRect(x: 150, y: 200,
13            width: 200, height: 20)
14        label.text = "Hello World!"
15        label.textColor = .black
16
17        view.addSubview(label)
18        self.view = view
19    }
20
21 // Present the view controller in the Live View window
22 PlaygroundPage.current.liveView =
23     MyViewController()
24
```

Single View Area:

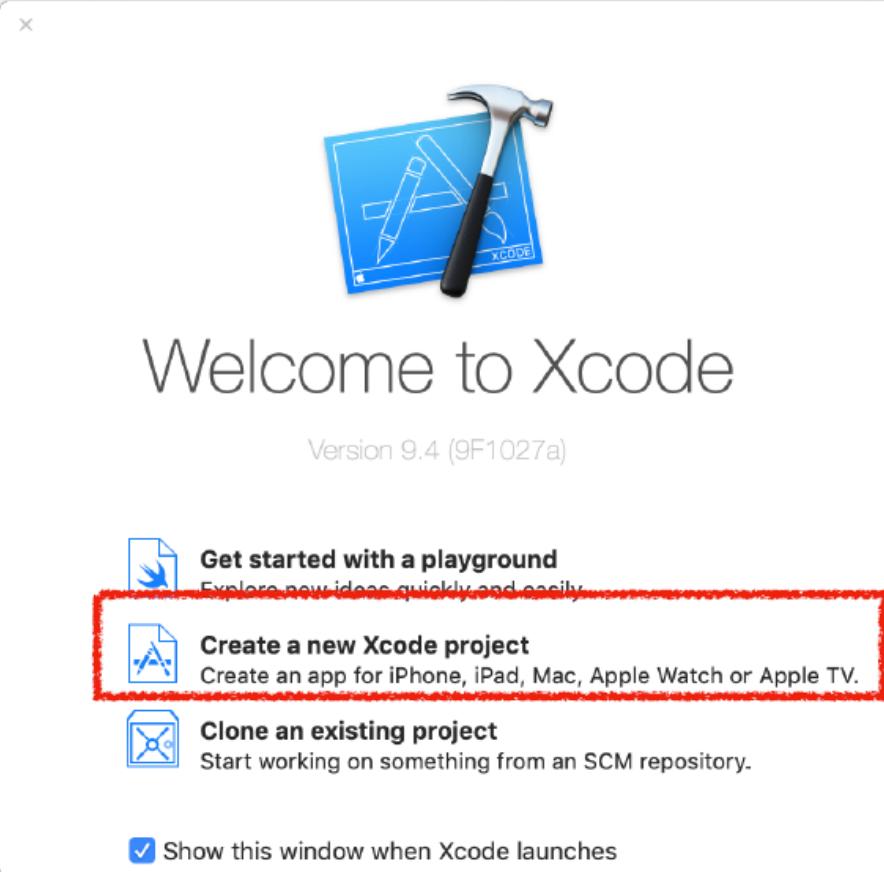
The "Single View Area" displays a large "Hello World!" text centered on the screen. The text is rendered in a large, bold black font. Below the text, there is a smaller, faint "Hello World!" watermark.



xcode #2

Simple Project

Create a New Project



Welcome to Xcode

Version 9.4 (9F1027a)

 **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.

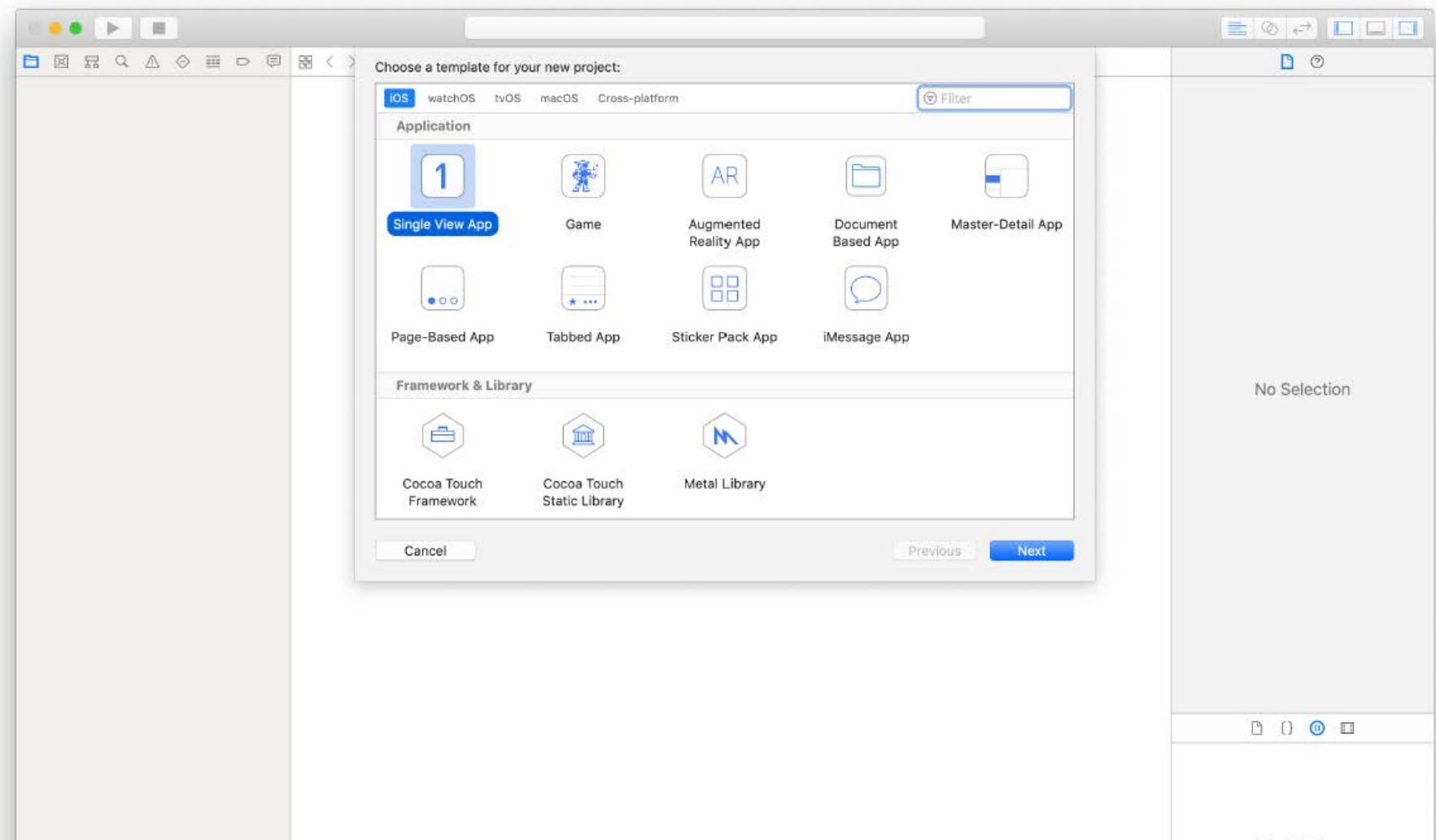
 **Clone an existing project**
Start working on something from an SCM repository.

Show this window when Xcode launches

Open another project...

**Recently
Opened
Projects
Area**

New Project



Choose options for your new project:

Product Name:

Team: None

Organization Name: Forscher Labs.

Organization Identifier: com.forscher

Bundle Identifier: com.forscher.ProductName

Language Swift

Objective-C

Use Core Data

Include Unit Tests

Include UI Tests

Cancel

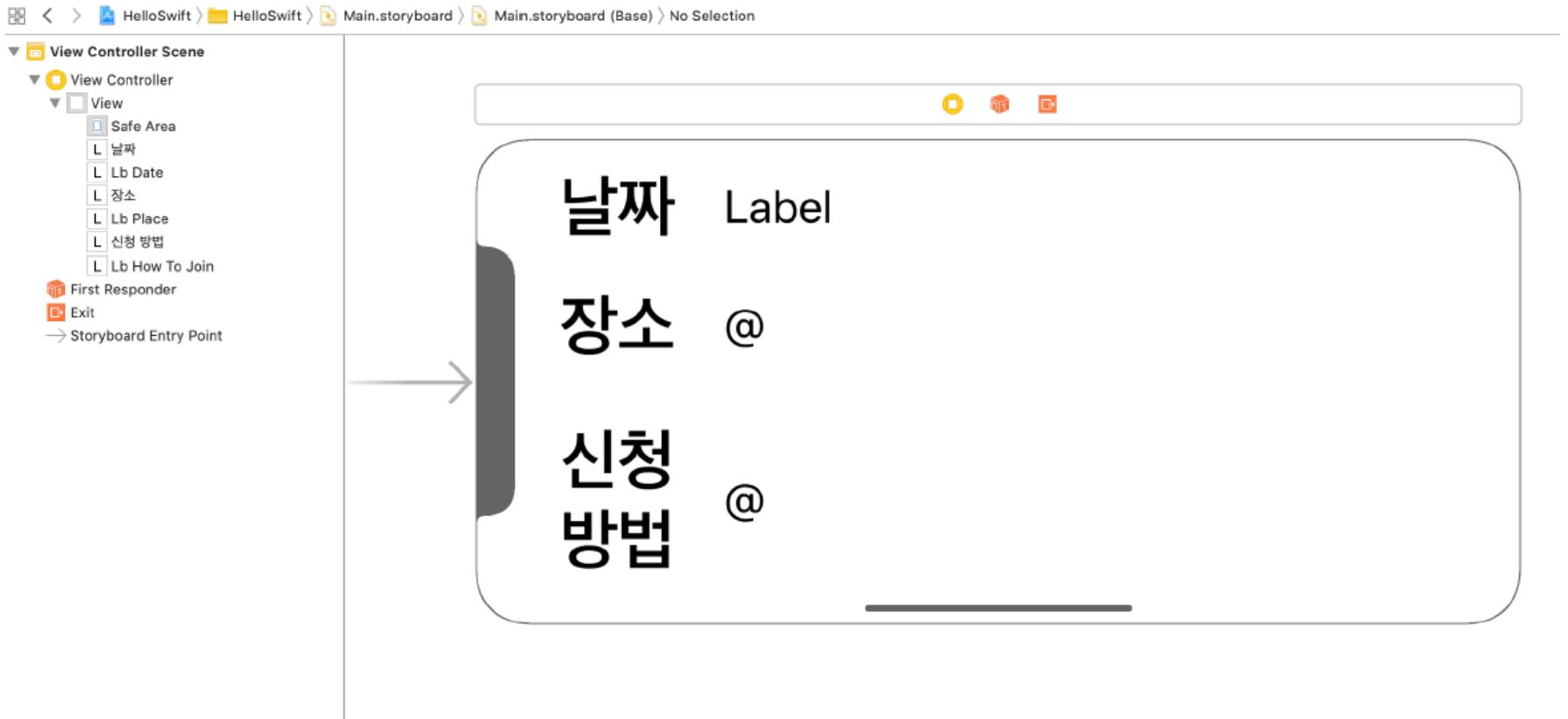
Previous

Next

The screenshot shows the Xcode interface with the following details:

- Title Bar:** HelloSwift > Generic iOS Device
- Status Bar:** HelloSwift | Build HelloSwift: **Succeeded** | 17/01/2019 at 10:56 AM
- Project Navigator:** Shows the project structure for "HelloSwift".
- Editor:** Displays the content of "ViewController.swift".

```
1 // ViewController.swift
2 // HelloSwift
3 //
4 //
5 // Created by SangGil Lee on 17/01/2019.
6 // Copyright © 2019 Forscher Labs. All rights reserved.
7 //
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     @IBOutlet weak var lbDate: UILabel!
14     @IBOutlet weak var lbPlace: UILabel!
15     @IBOutlet weak var lbHowToJoin: UILabel!
16     override func viewDidLoad() {
17
18         super.viewDidLoad() // iOS 앱 개발 시작!! + Swift
19
20         let 기한 = "2019년 1월 28일~30일(월~수) 오후 1시~4시"
21
22         let 장소 = "@Mac실습실 \n(공과대학 5호관3층 315호)"
23
24         let 신청방법 = "1/26(토)까지 아래 URL에서 설문 링크 이용"
25             + "http://bitly.kr/dMJI7 또는"
26             + "https://sites.google.com/cs-cnu.org/swift2019/"
27
28
29         self.lbDate.text = 기한
30         self.lbPlace.text = 장소
31         self.lbHowToJoin.text = 신청방법
32     }
33
34
35
36 }
37
38
```



날짜	2019년 1월 28일-30일(월-수) 오후...
장소	@Mac실습실 (공과대학 5호관3층 3...
신청 방법	1/26(토)까지 홈페이지 내의 참가 신청 설문 링크 이용이름, 직전학기 학년, 학 번, 휴대전화번호 입력