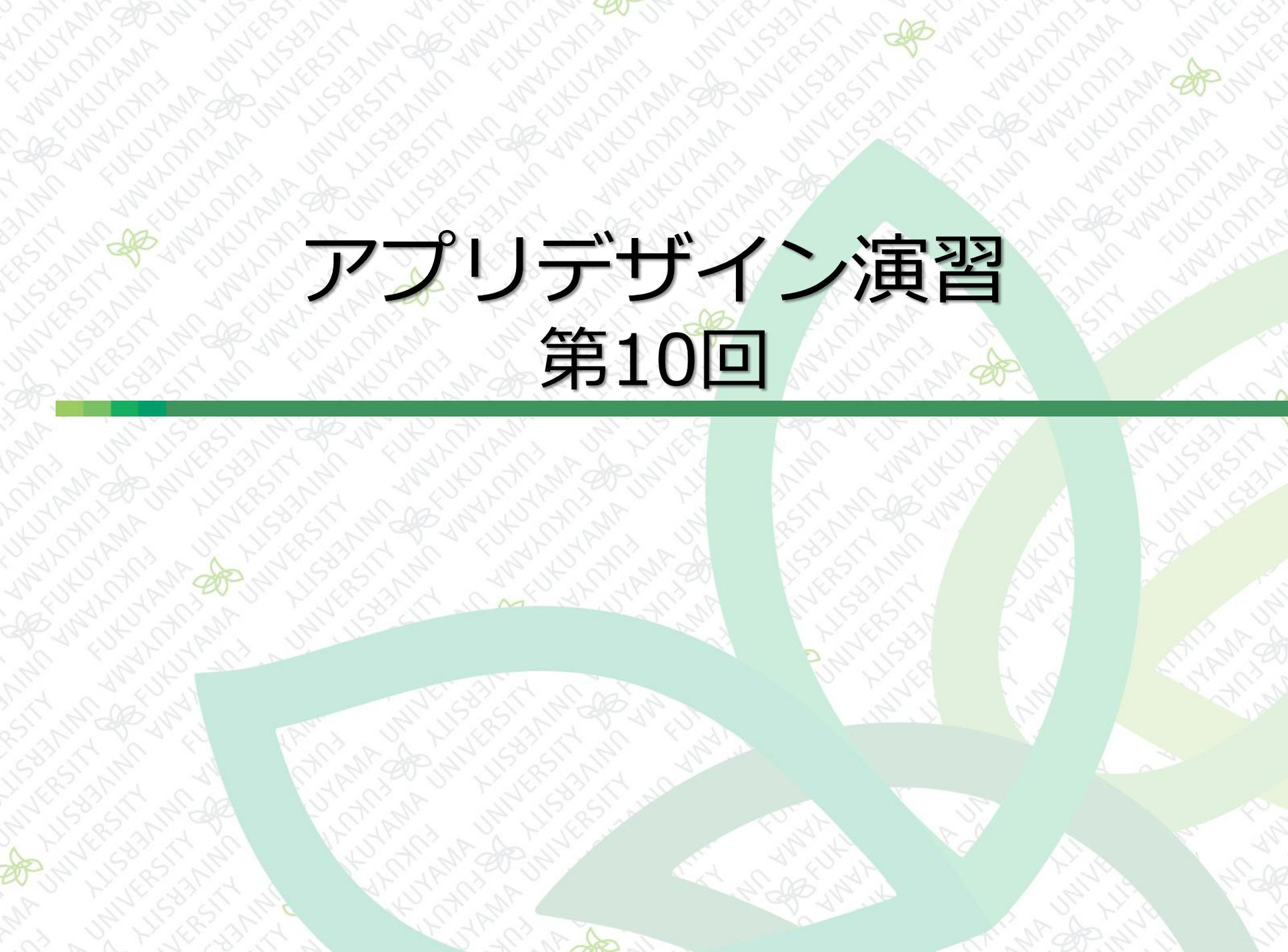


# アプリデザイン演習 第10回



# 今日の予定

## ■ 準備

## ■ iOSアプリの開発について

## ■ Xcodeの概要

## ■ アプリ作成例（1） - Helloを表示

## ■ アプリ作成例（2） - Webを表示

# 準備

- 指定されたグループの所に着席する
- 1グループに1台MacBookを使用
- 棚から出して各グループに設置する
- 使い終わったら、元に戻しておくのを忘れずに！

# iOSアプリの開発について

- iPhoneやiPadで動作するアプリ
- MacのPCが必要
  - クロスプラットフォームもあるが未成熟
- Xcode(統合開発環境)を使用する
- シミュレータと実機で実行テスト
- 開発言語は Swift がメイン
  - Objective-C も一部使う…

# iOSアプリの開発について

## ■ 開発の手順

プロジェクトを作る

- ・ テンプレートからアプリの種類を選ぶ
- ・ プロジェクトの名前を決める

画面をデザインする

- ・ **Interface Builder** を使う
- ・ 画面が複数ある場合は画面同士のつながりも考える

部品とプログラムをつなぐ

- ・ **Assistant Editor** を使う
- ・ 部品の処理内容を考える、名前を付ける

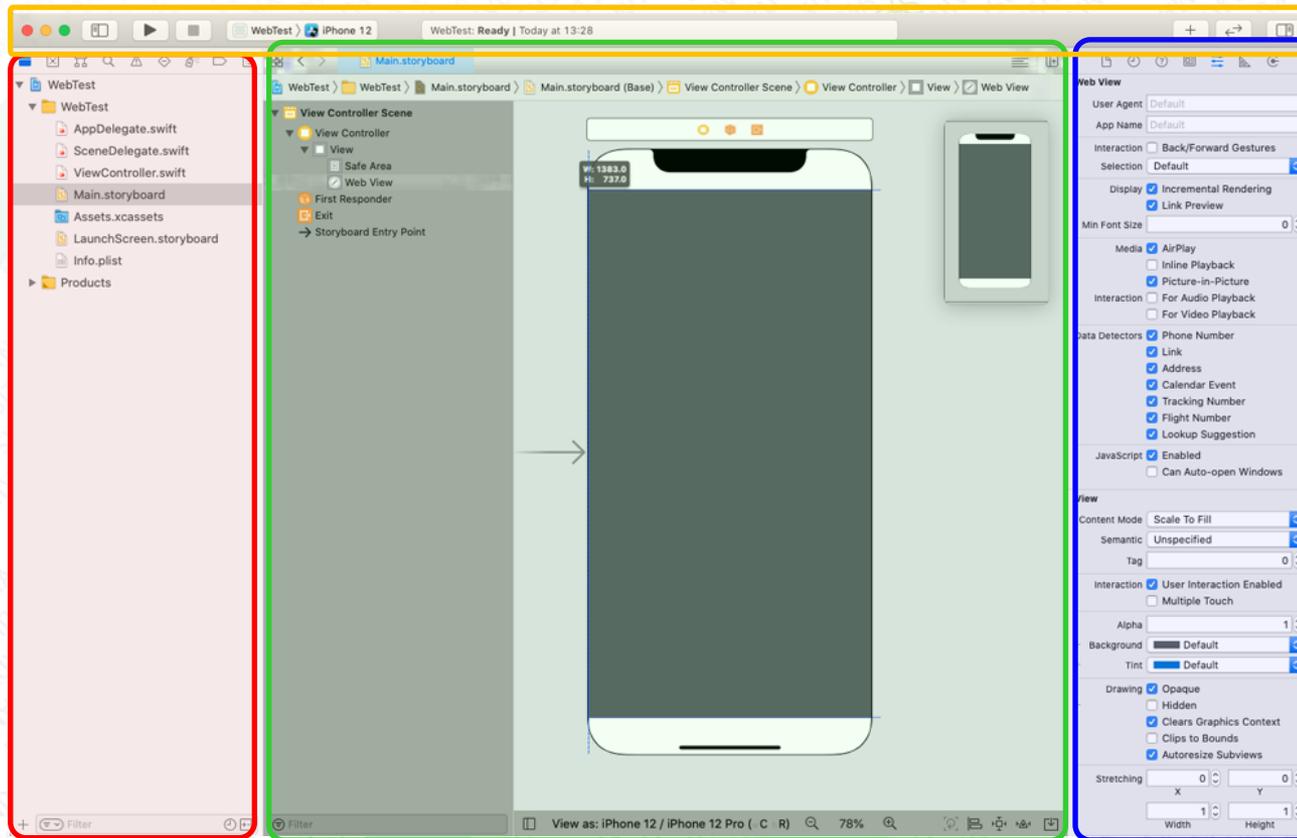
プログラムを書く

- ・ **Source Editor** を使う
- ・ アプリ内で行う処理を記述する

シミュレータで実行する

- ・ アプリの動作確認

# Xcodeの概要



- **ナビゲータエリア**：作業ファイルを選択する
- **ツールバー**：画面の切り替え、シミュレーターの実行など
- **エディターエリア**：画面の作成、プログラムの編集
- **ユーティリティエリア**：画面を作る部品を選択、設定

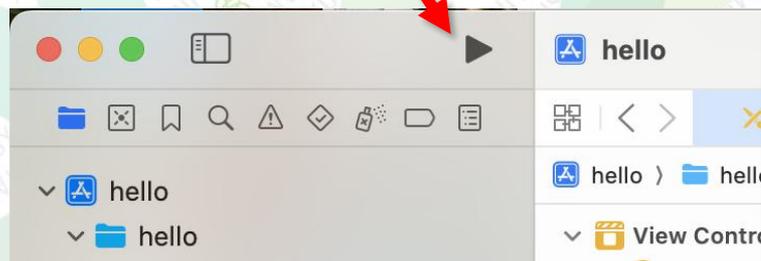
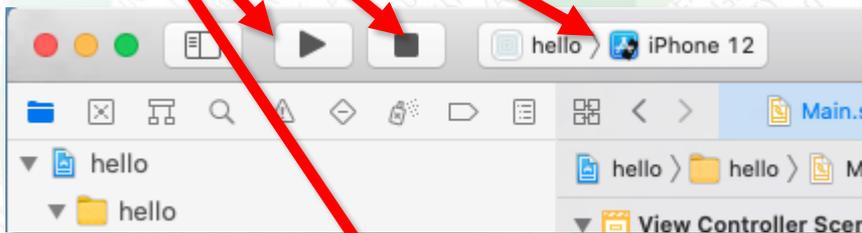
# Xcodeの概要

- ・ **ツールバー**：画面の切り替え、シミュレーターの実行など

[Run(実行)]ボタン：プロジェクトをビルドして、アプリを実行する

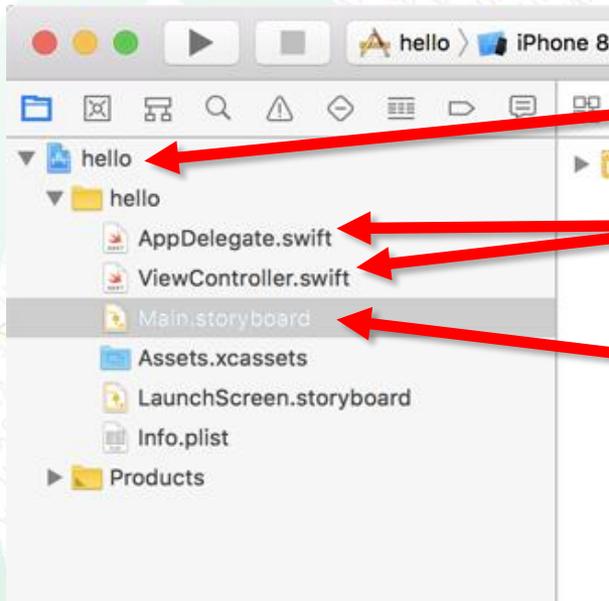
[Stop(停止)]ボタン：実行中のアプリを停止する

[Scheme(スキーム)]ボタン：テストするシミュレータの種類を選択する



# Xcodeの概要

- ・ **ナビゲータエリア**：作業ファイルを選択する



プロジェクトの設定

「\*.swift」プログラムソースのファイル

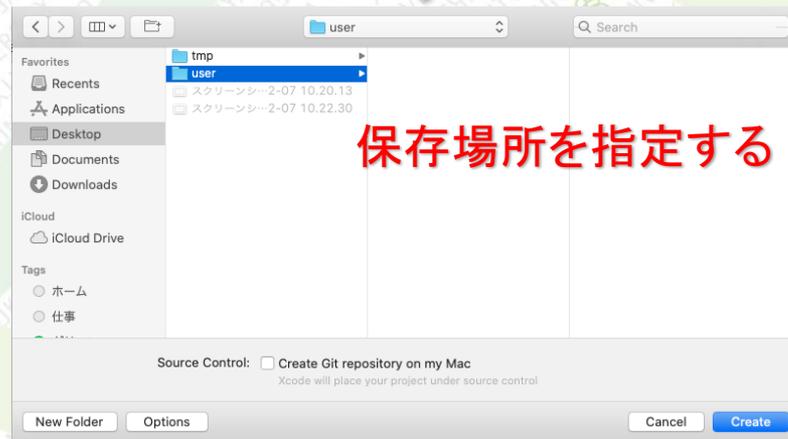
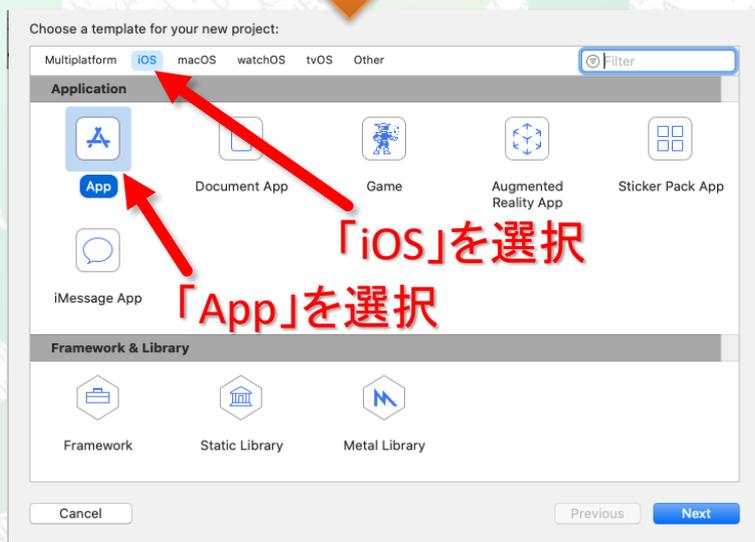
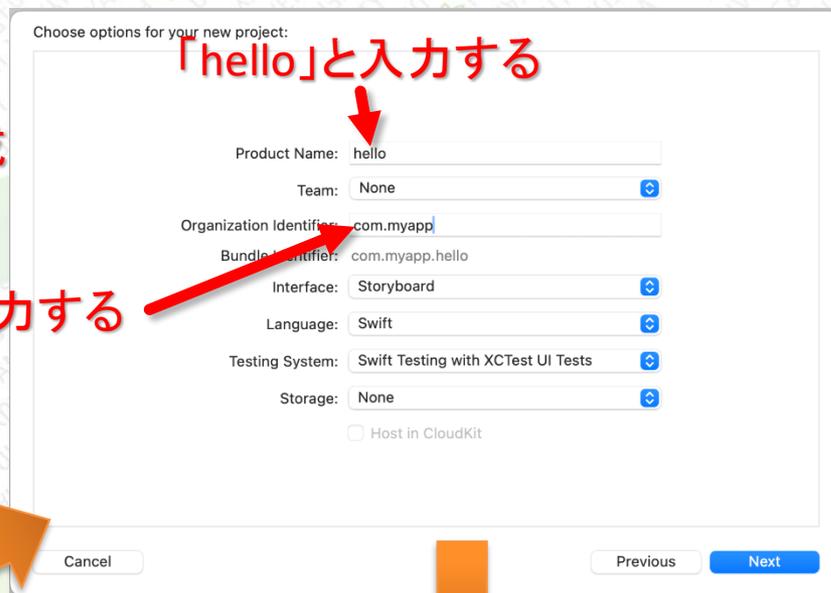
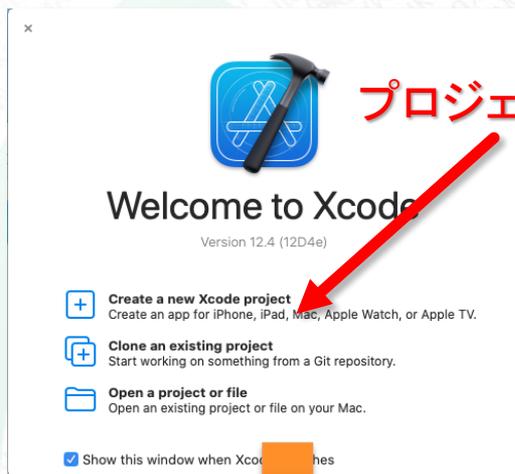
「\*.storyboard」画面の作成、  
InterfaceBuilder

- ・ **ナビゲータエリア**でファイルを選択すると、**エディターエリア**の表示が切り替わる

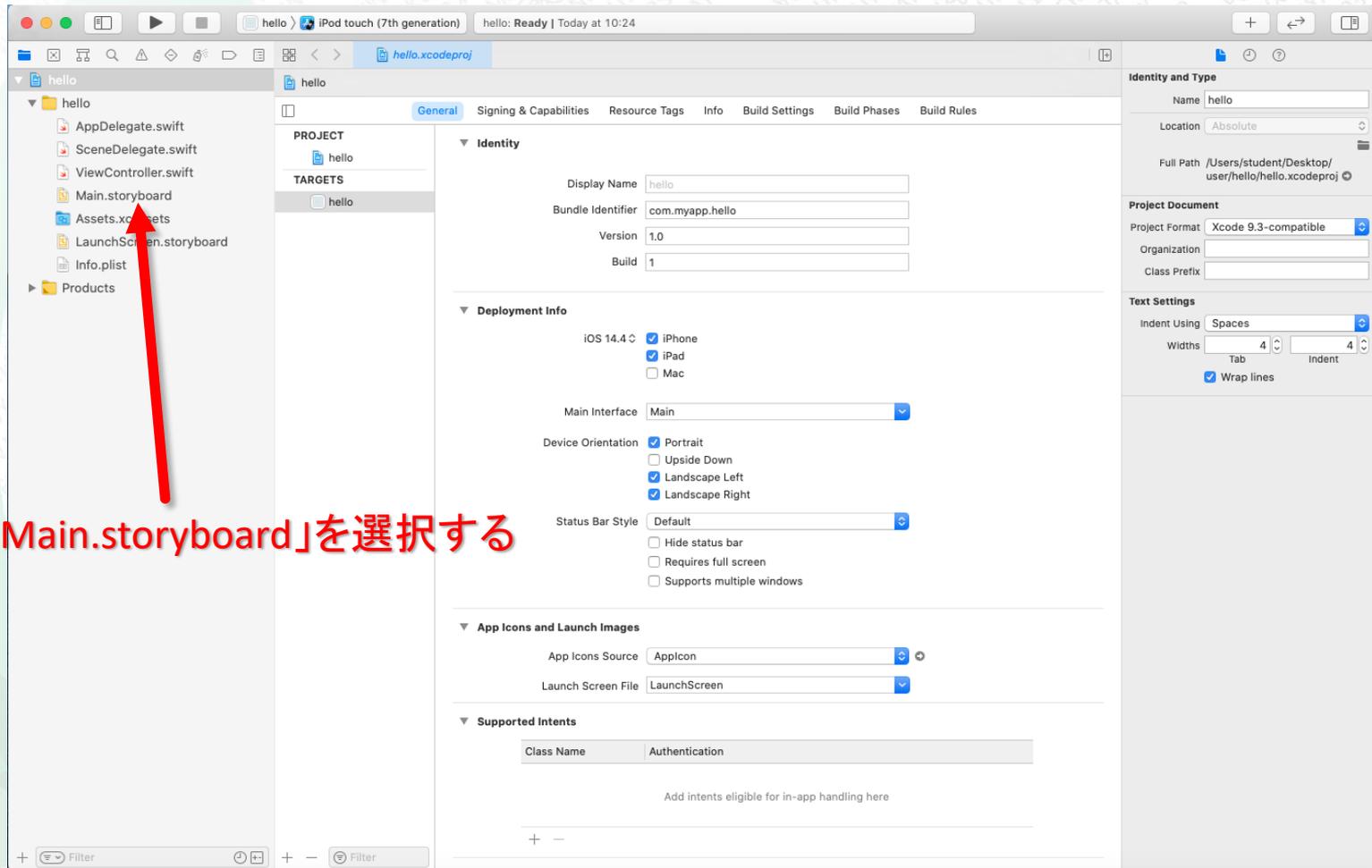
# アプリ作成例（1） - Helloを表示

# アプリ作成例（1） - Helloを表示

- プロジェクトを新規作成する

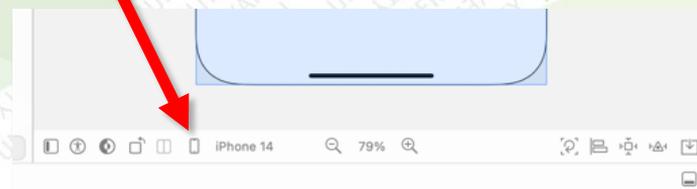


# アプリ作成例（1） - Helloを表示



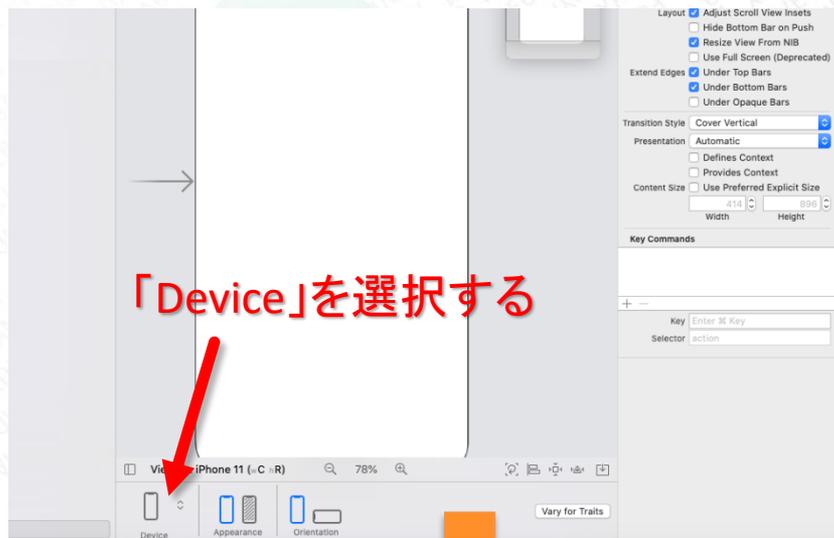
「Main.storyboard」を選択する

# アプリ作成例（1） - Helloを表示

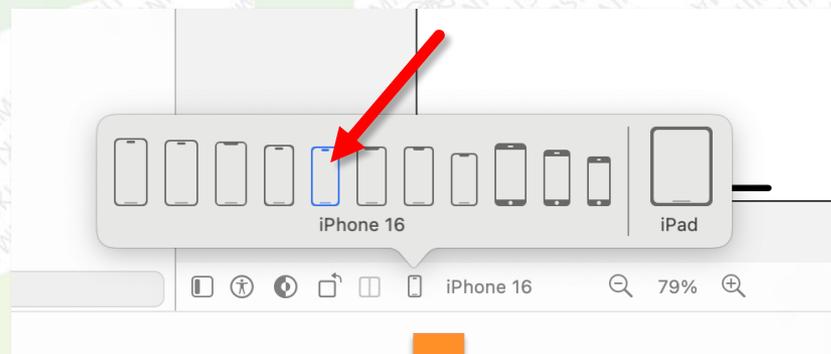


# アプリ作成例（1） - Helloを表示

- Storyboardの機種を変更する



または



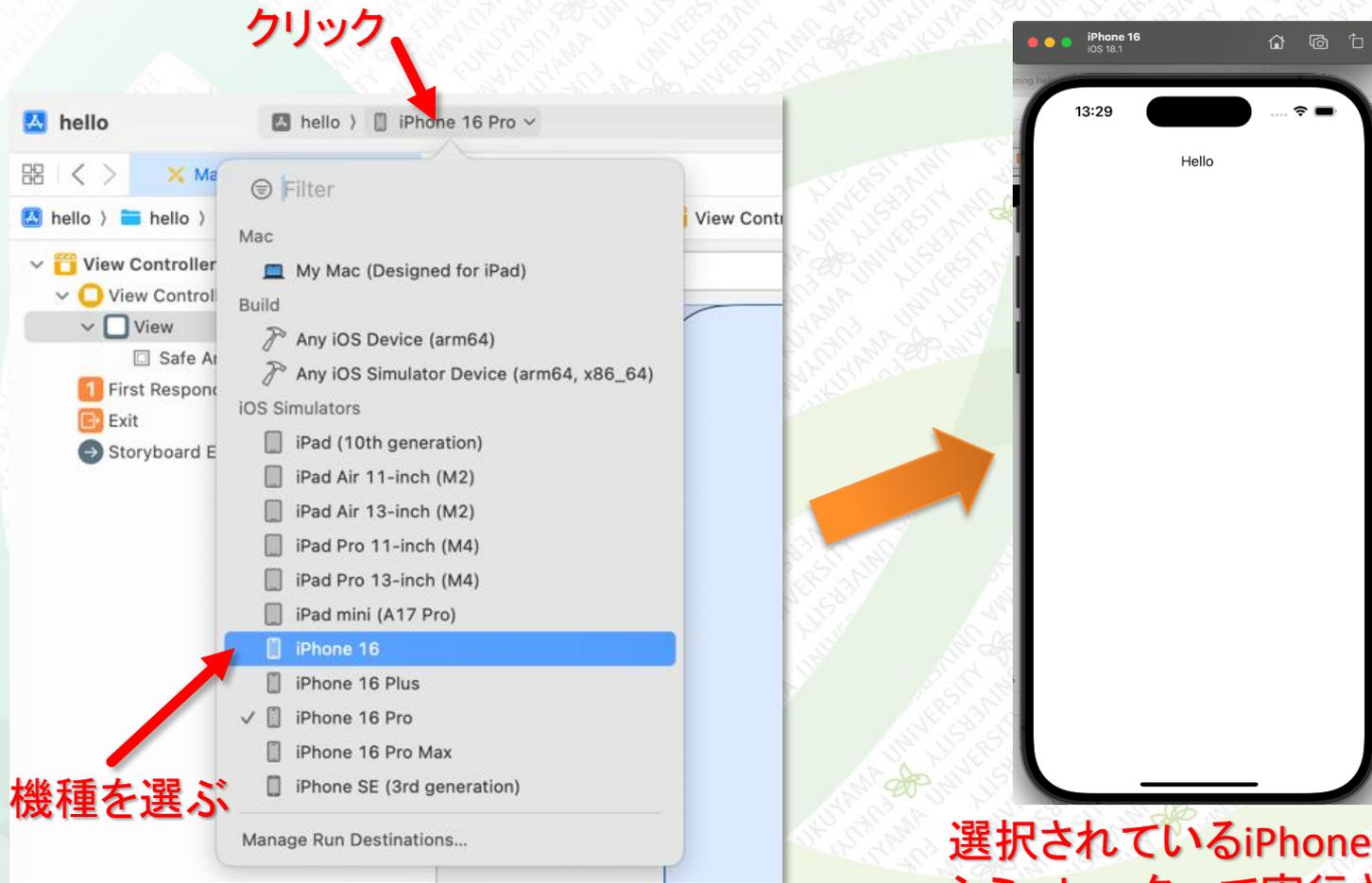
選択して変更



機種を選ぶ  
(iPhone12 or iPhone14)

# アプリ作成例（1） - Helloを表示

- ・ シミュレーターの機種を変更する

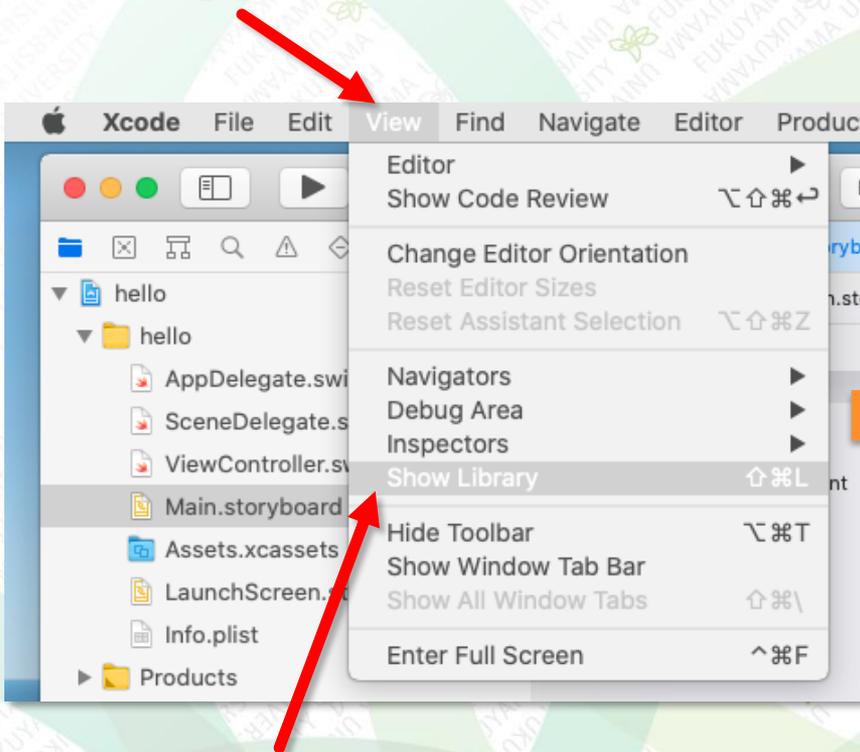


選択されているiPhoneの  
シミュレーターで実行される

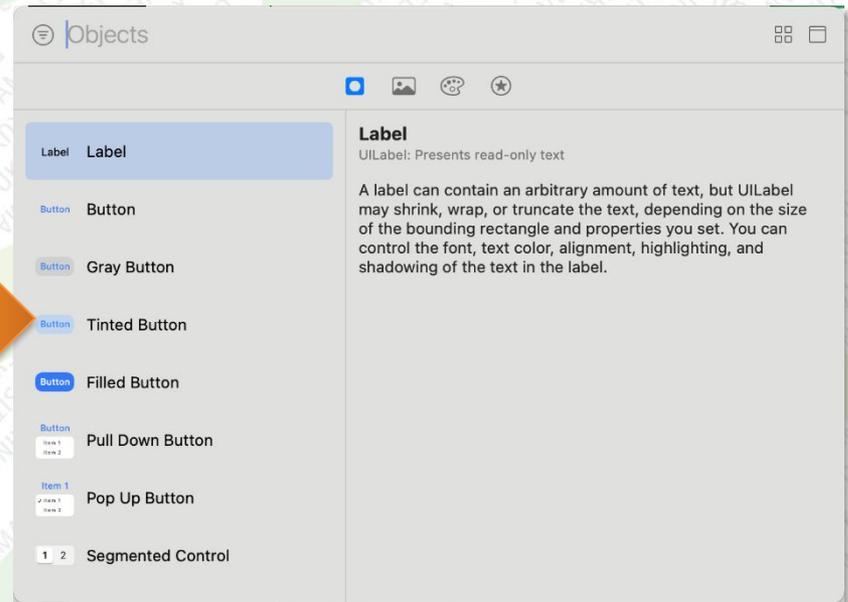
# アプリ作成例（1） - Helloを表示

- ・ライブラリを表示する

「View」をクリック

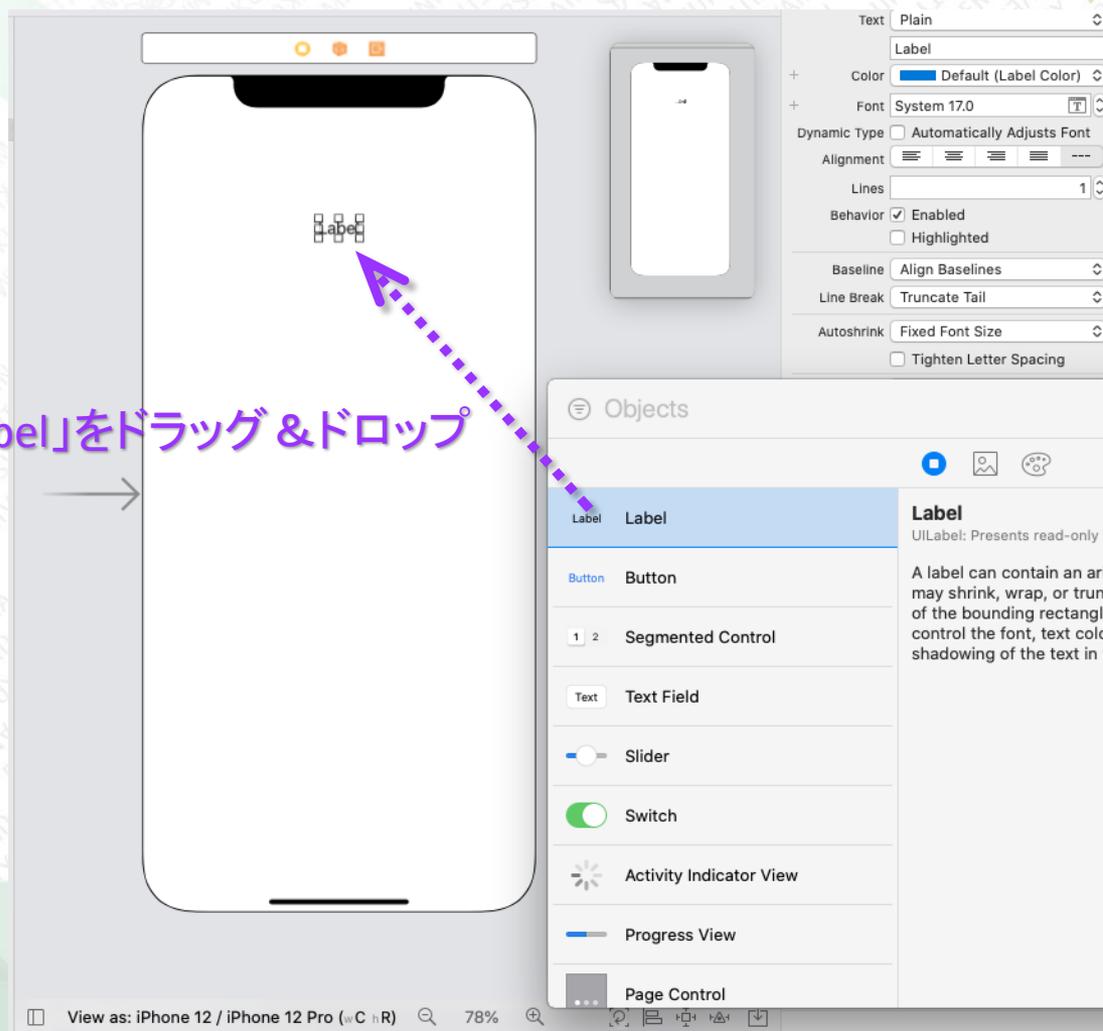


「Show Library」を選択



「Library」が表示される

# アプリ作成例（1） - Helloを表示

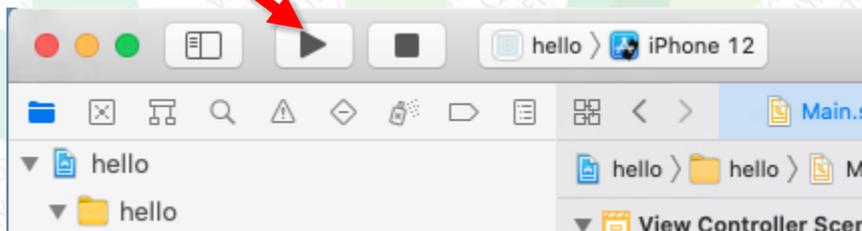


# アプリ作成例（1） - Helloを表示



このように表示されたらOK!

クリックして、ビルド&実行する

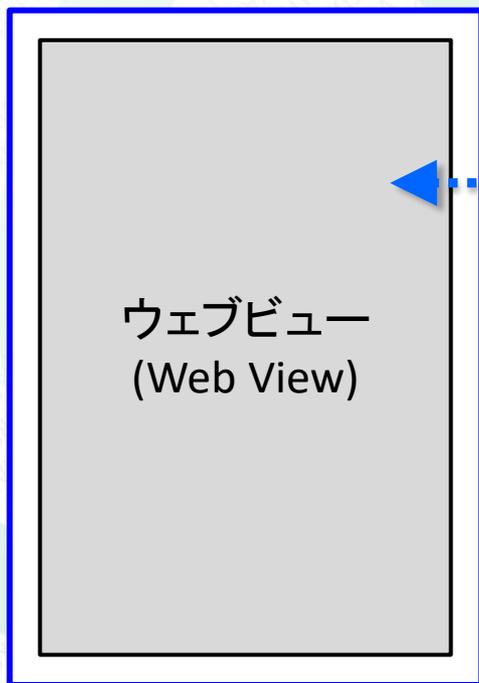


# アプリ作成例（2） - Webを表示

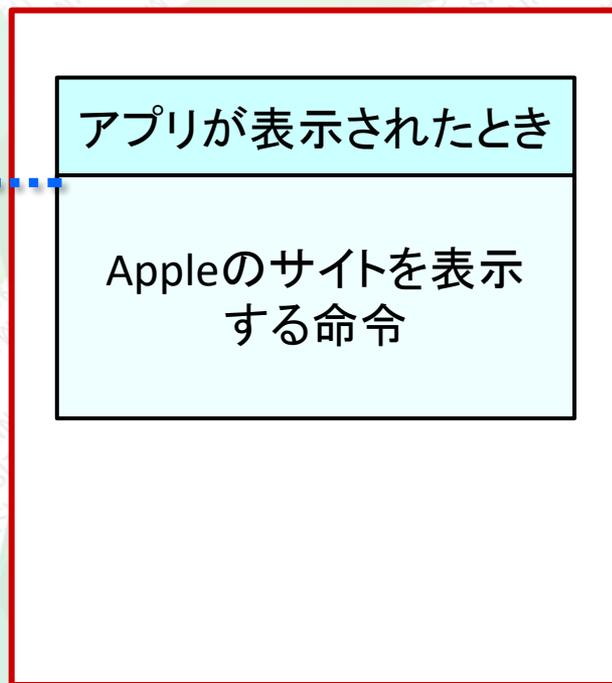
# アプリ作成例（2） - Webを表示

- AppleのWebページを表示するアプリを作成する

<アプリの画面>



<プログラム>

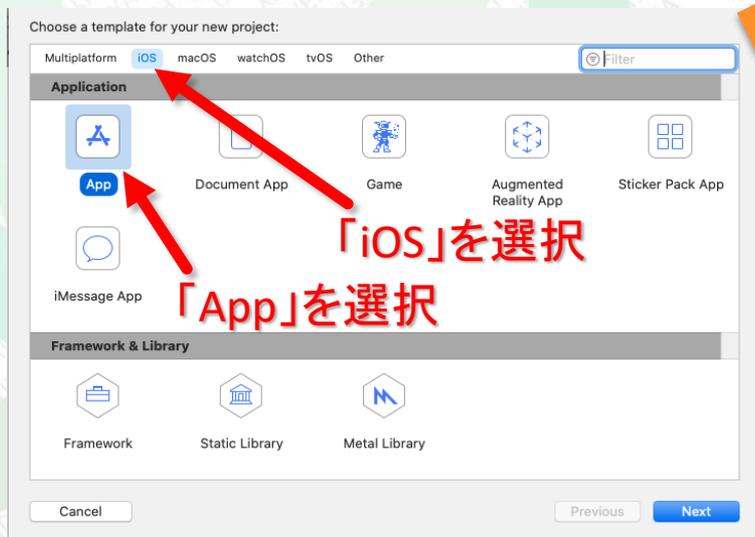
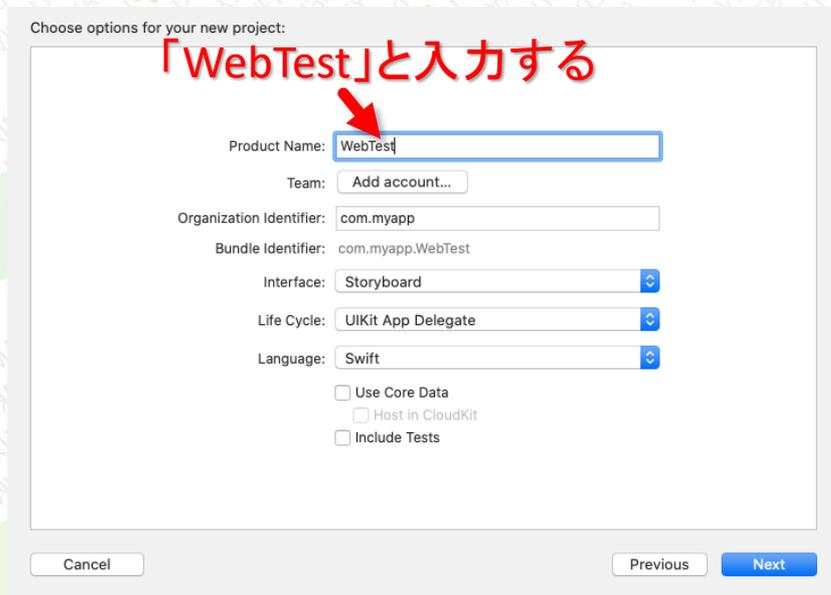
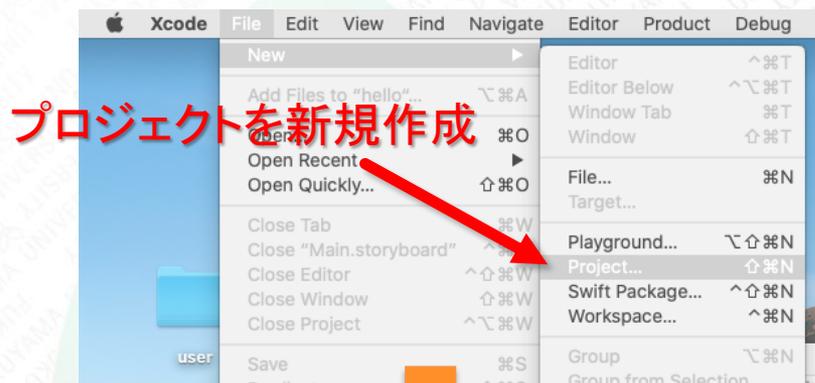


完成例



# アプリ作成例（2） - Webを表示

- プロジェクトを新規作成する

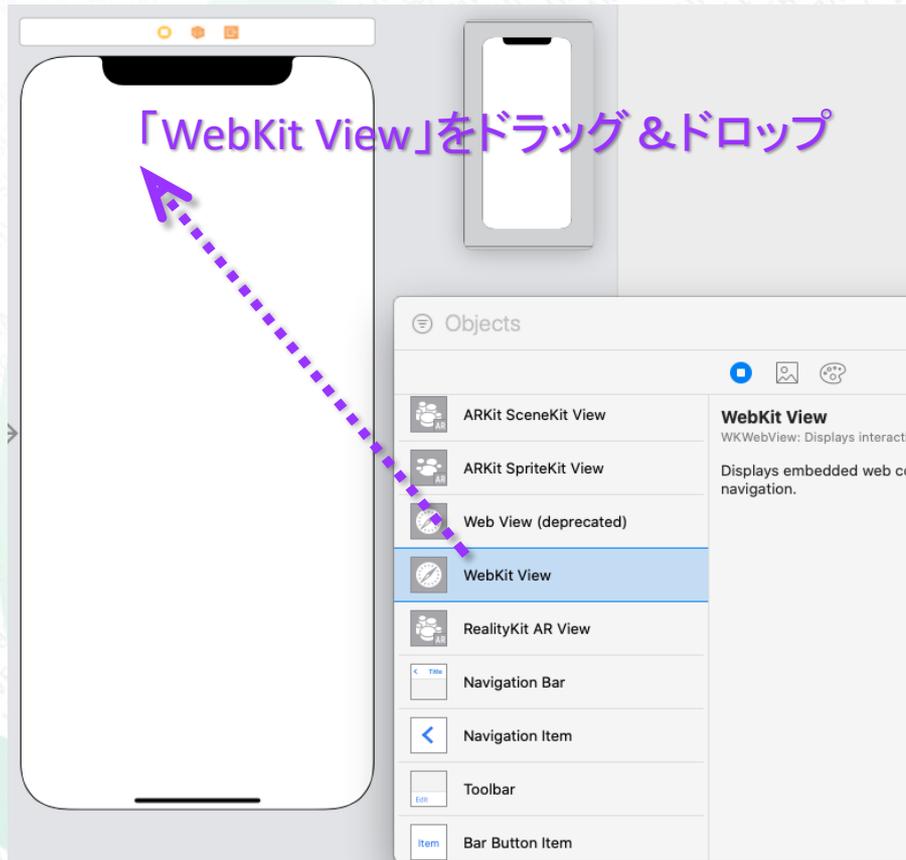


保存場所を指定する

# アプリ作成例（2） - Webを表示

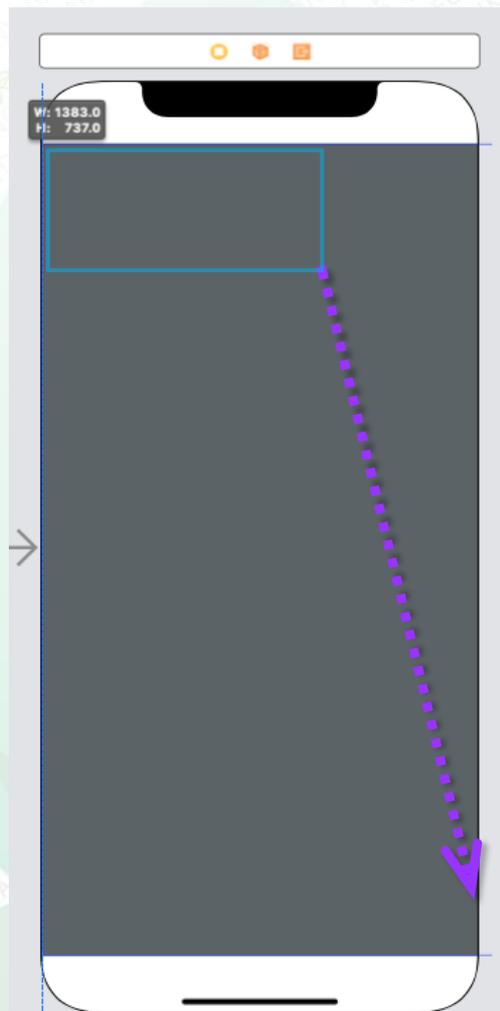


# アプリ作成例（2） - Webを表示



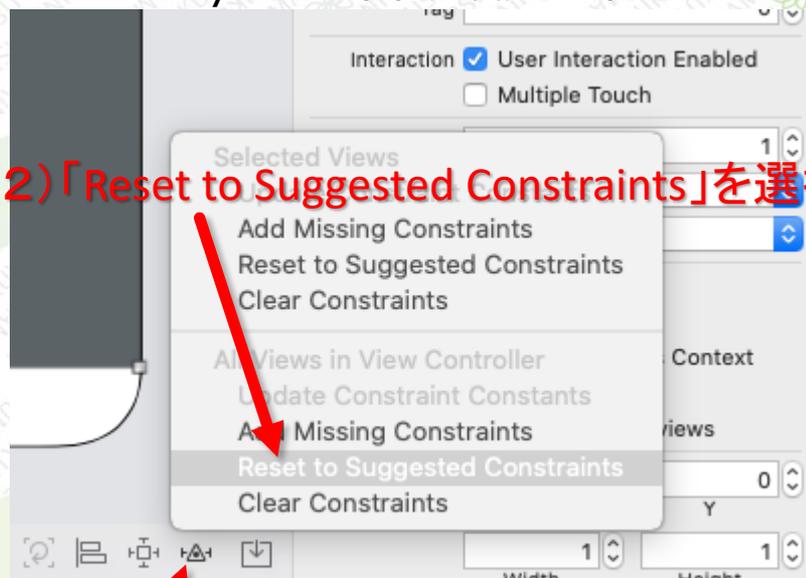
# アプリ作成例（2） - Webを表示

- Auto Layoutの自動設定を行う

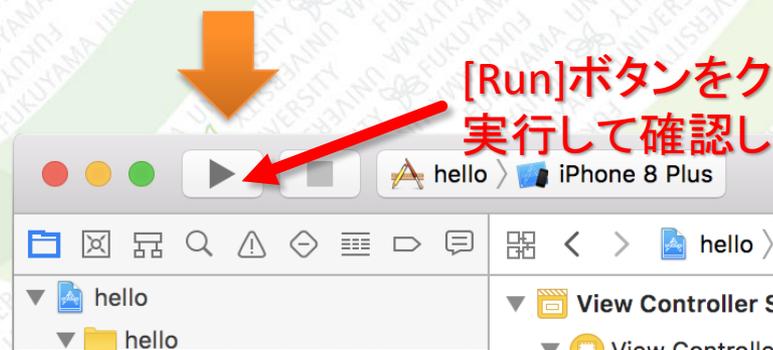


ドラッグして、画面いっぱい広げる

(2)「Reset to Suggested Constraints」を選択



(1)「Resolve Auto Layout Issues」をクリック



[Run]ボタンをクリック、  
実行して確認してみる

# アプリ作成例（２） - Webを表示

- ・画面上の部品とプログラムをつなげる

- [画面上の部品]と[プログラム]がつながると、プログラムから部品をコントロールすることができるようになる

「IBOutlet接続」：部品に名前をつけてつなぐ

- プログラムから部品の値を調べる、部品の値を更新するなど

「IBAction接続」：部品に仕事の名前をつけてつなぐ

- ボタンを押す、スイッチを切り替えるなど部品を操作したときに実行する仕事（メソッド）を設定する

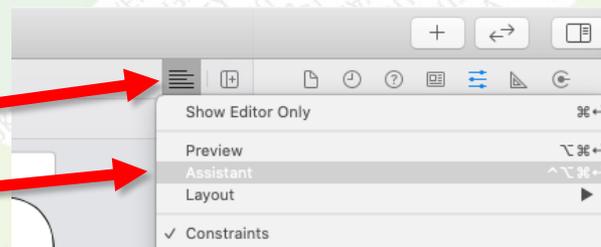
IB : Interface Builder の頭文字

- ・ [画面上の部品]と[プログラム]をつなげるには

「アシスタント・エディター」を使う

クリック

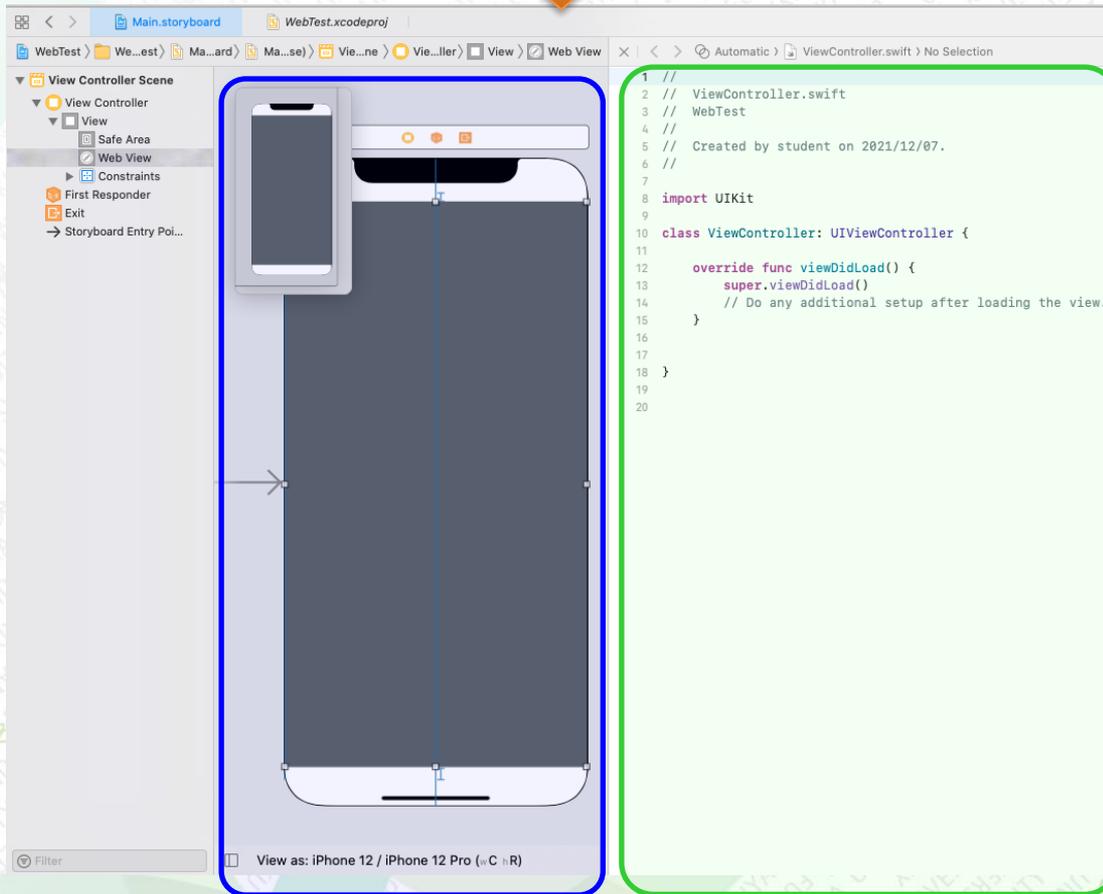
「アシスタント・エディター」



# アプリ作成例（2） - Webを表示



「ソース・エディター」が追加表示



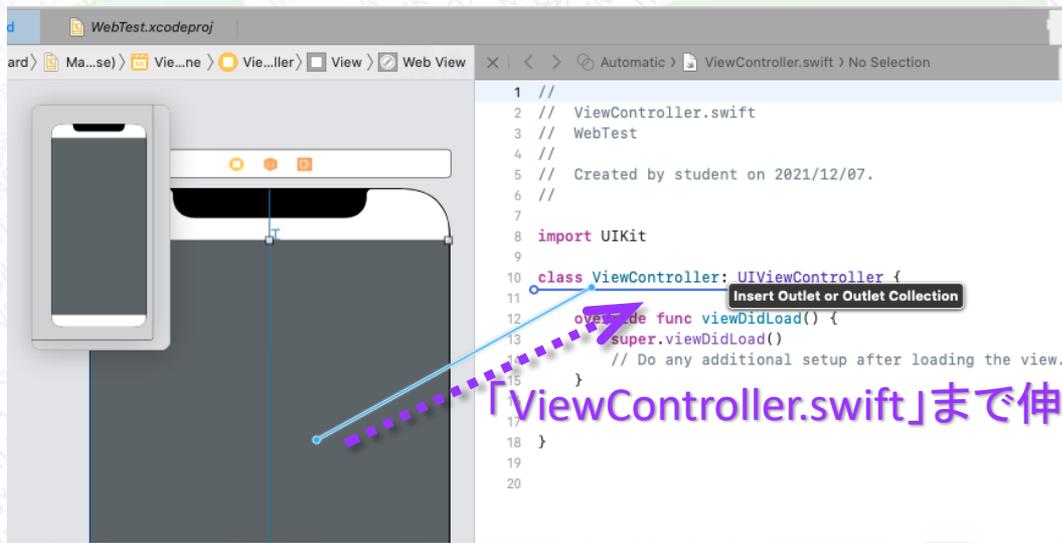
インタフェース・ビルダー

ソース・エディター

# アプリ作成例（2） - Webを表示

- Web Viewをプログラムに接続する

> [Web View]を右クリック(control+クリック)してドラッグすると線が伸びる

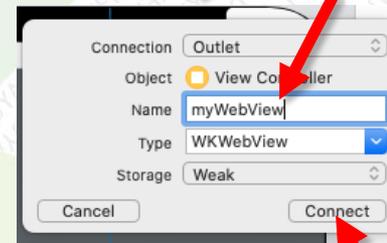


「ViewController.swift」まで伸ばしてドロップする

```
1 //
2 // ViewController.swift
3 // WebTest
4 //
5 // Created by student on 2021/12/07.
6 //
7
8 import UIKit
9
10 class ViewController: UIViewController {
11     @IBOutlet weak var myWebView: WKWebView!
12 }
13
14 override func viewDidLoad() {
15     super.viewDidLoad()
16     // Do any additional setup after loading the view.
17 }
18 }
19
20
21
```

IBOutletが追加されていることを確認

エラーが出るのは...



(1)「myWebView」と入力する

(2)クリック

# アプリ作成例（2） - Webを表示

- エラーが出るのは、必要なライブラリをimportしていないから

```
1 //
2 // ViewController.swift
3 // WebTest
4 //
5 // Created by student on 2021/12/07.
6 //
7
8 import UIKit
9
10 class ViewController: UIViewController {
11     @IBOutlet weak var myWebView: WKWebView!
12
13     override func viewDidLoad() {
14         super.viewDidLoad()
15         // Do any additional setup after loading the view.
16     }
17
18 }
19
20
21
```

エラーが表示されている

Cannot find type 'WKWebView' in scope

この1行を追加する

```
8 import UIKit
9 import WebKit
10
11 class ViewController: UIViewController {
12     @IBOutlet weak var myWebView: WKWebView!
13
14     override func viewDidLoad() {
15         super.viewDidLoad()
16     }
17
18 }
19
20
21
```

```
1 //
2 // ViewController.swift
3 // WebTest
4 //
5 // Created by student on 2021/12/07.
6 //
7
8 import UIKit
9 import WebKit
10
11 class ViewController: UIViewController {
12     @IBOutlet weak var myWebView: WKWebView!
13
14     override func viewDidLoad() {
15         super.viewDidLoad()
16         // Do any additional setup after loading the view.
17         // HPを開く
18         let url = URL(string:"https://www.apple.com/jp/")
19         let request = URLRequest(url: url!)
20         myWebView.load(request as URLRequest)
21     }
22
23
24
25 }
```

エラーが消える

# アプリ作成例（2） - Webを表示

- Webページを表示するプログラムを書く

```
1 //
2 // ViewController.swift
3 // WebTest
4 //
5 // Created by student on 2021/12/07.
6 //
7
8 import UIKit
9 import WebKit
10
11 class ViewController: UIViewController {
12     @IBOutlet weak var myWebView: WKWebView!
13
14     override func viewDidLoad() {
15         super.viewDidLoad()
16         // Do any additional setup after loading the view.
17         // HPを開く
18         let url = URL(string:"https://www.apple.com/jp/")
19         let request = URLRequest(url:url!)
20         myWebView.load(request as URLRequest)
21
22     }
23
24
25 }
```

コードを追加する



# アプリ作成例（2） - Webを表示

- ・シミュレーターで実行してみる

