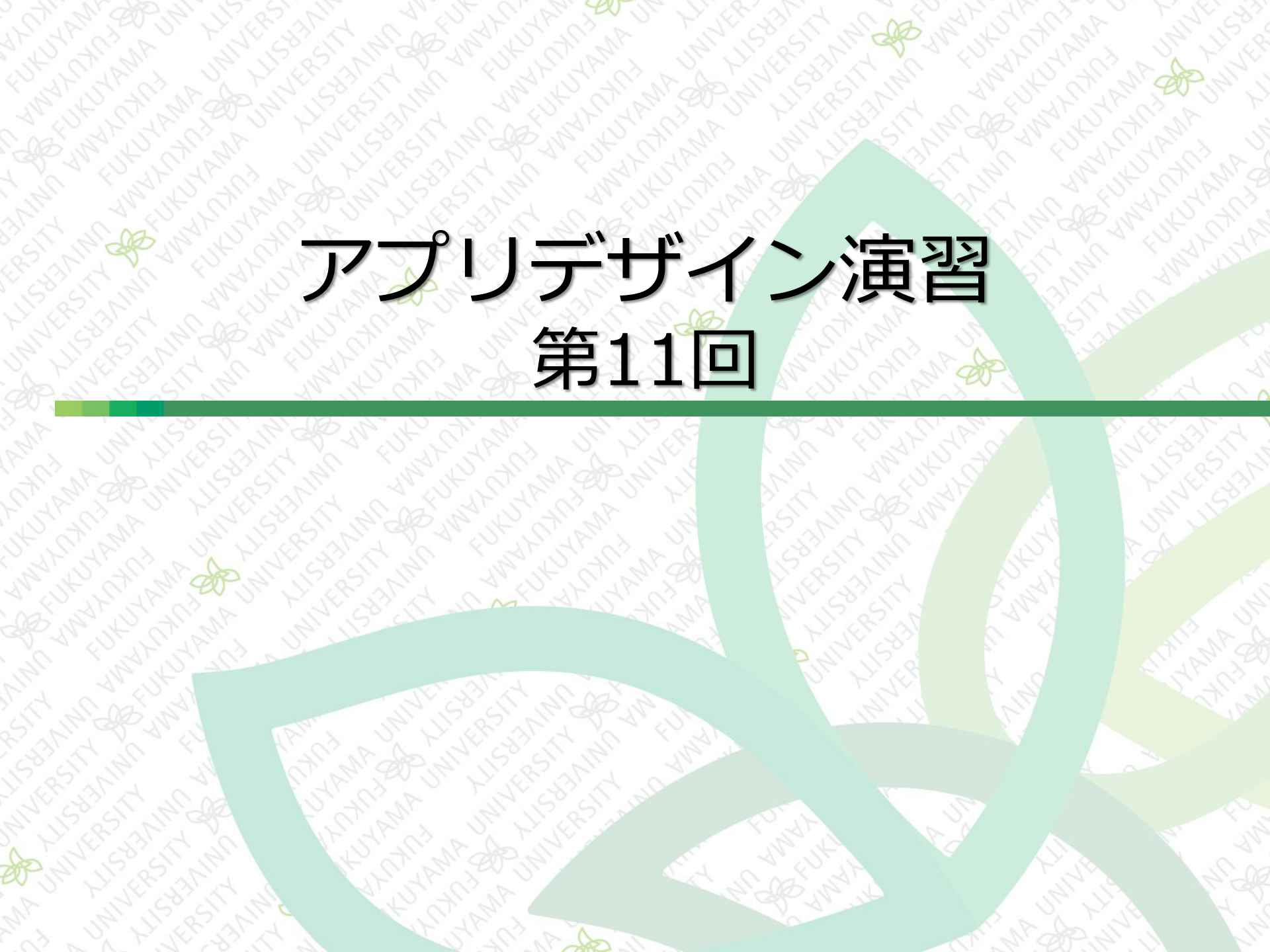


アプリデザイン演習

第11回



今日の予定

■ 準備

■ アプリを作る部品(Uikit)

■ アプリ作成例 (1) - ラベルとボタン

■ アプリ作成例 (2) - いろいろな部品

■ アプリ作成例 (3) - 複数の画面

準備

- グループごとに着席する
- 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！

アプリを作る部品(Uikit)

■ 様々な部品を利用する

Label **Label** - A variably sized amount of static text.

・・・文字列を表示する

Button

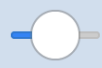
Button - Intercepts touch events and sends an action message to a target object when it's tapped.

・・・タップして何かを処理する



Switch - Displays an element showing the boolean state of a value. Allows tapping the control to toggle t...

・・・オン・オフの切り替え



Slider - Displays a continuous range of values and allows the selection of a single value.

・・・つまみをスライドして値を設定する

Text

Text Field - Displays editable text and sends an action message to a target object when Return is tapped.

・・・1行のテキスト(文字列)を入力する



Image View - Displays a single image, or an animation described by an array of images.

・・・画像を表示する

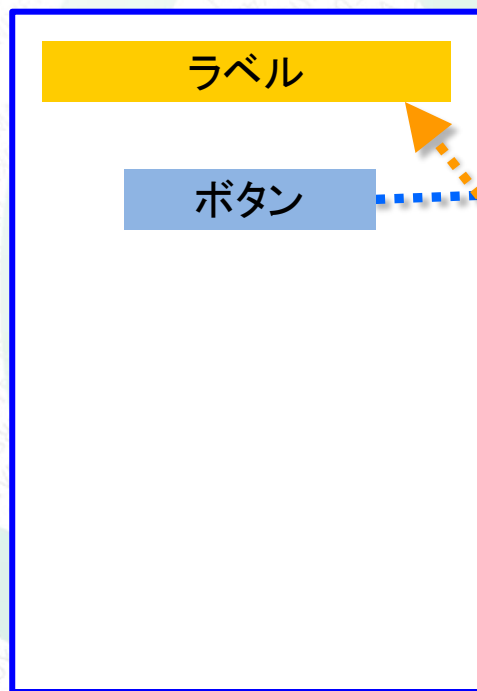
アプリ作成例（１）

- ラベルとボタン

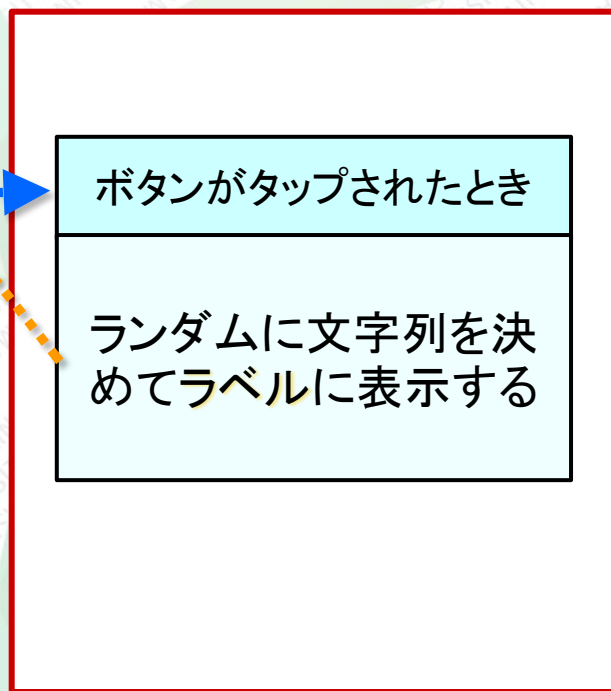
アプリ作成例（１）

- ・ ボタンが押されたら、文字列をランダムに表示するアプリ

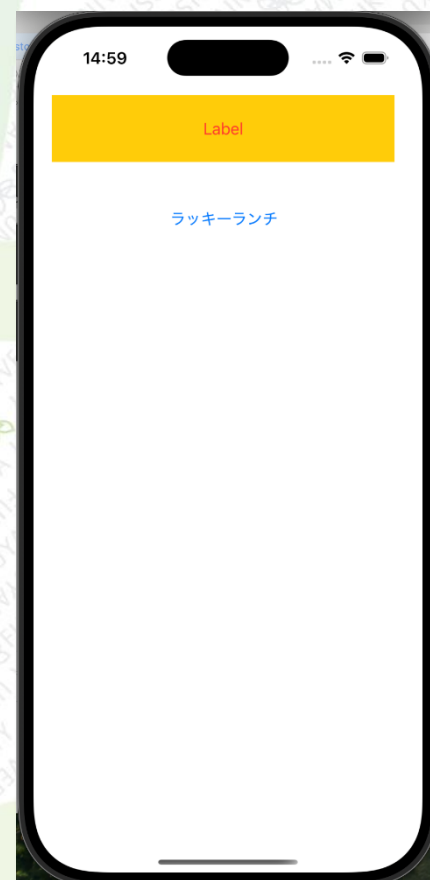
<アプリの画面>



<プログラム>



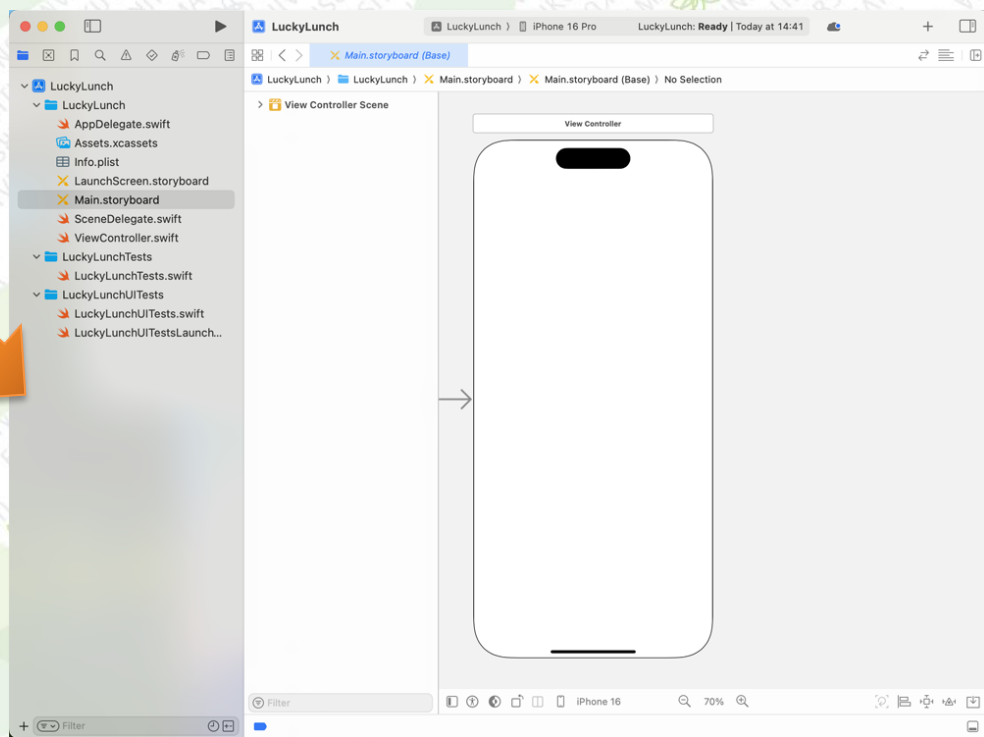
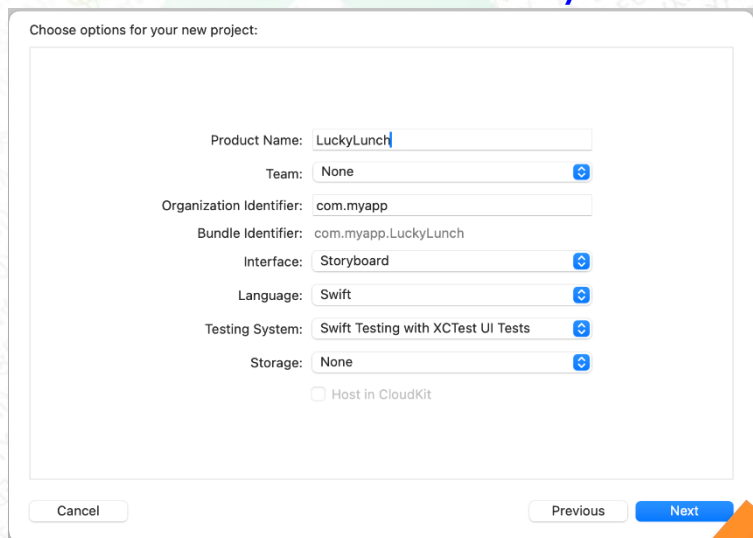
完成例



アプリ作成例（１）

- ・プロジェクトを新規作成する

プロジェクト名は「**LuckyLunch**」



アプリ作成例 (1)

- 「Label」を画面に設置する

(1) クリック

(2) 「Label」をドラッグ&ドロップ

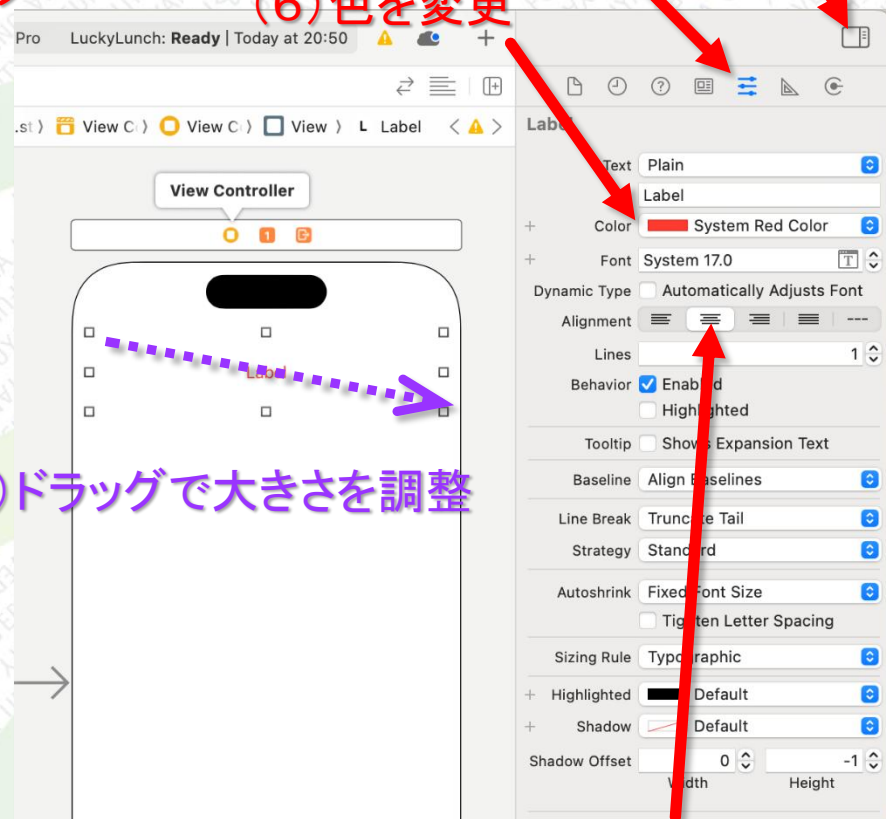
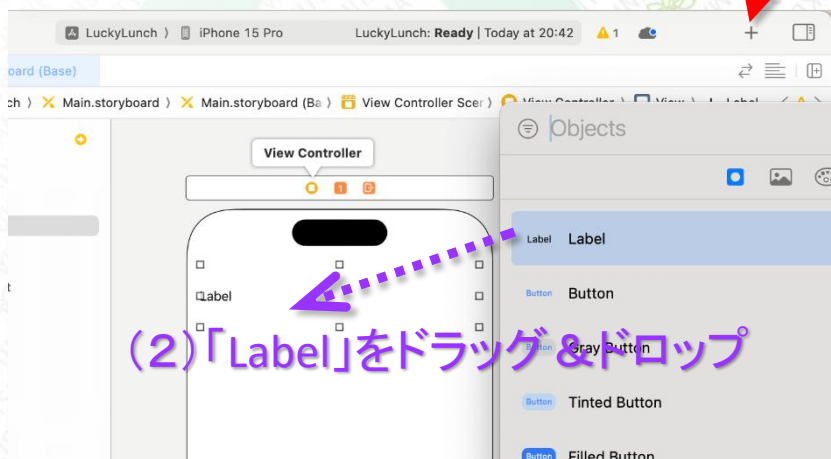
(5) ドラッグで大きさを調整

(4) クリック

(6) 色を変更

(3) クリック

(7) センタリング



アプリ作成例 (1)

- 「Label」を画面に設置する

アシスタントエディタ

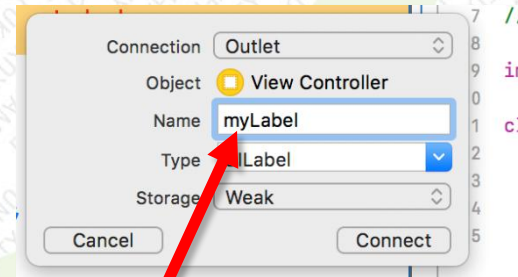
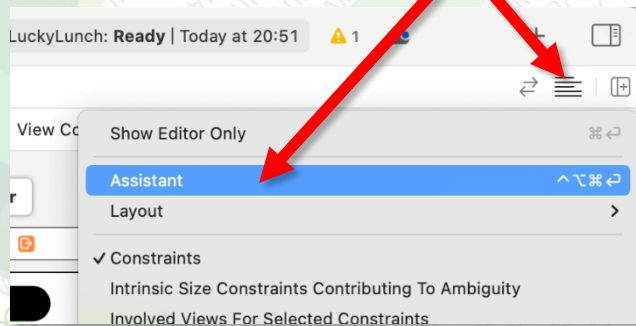
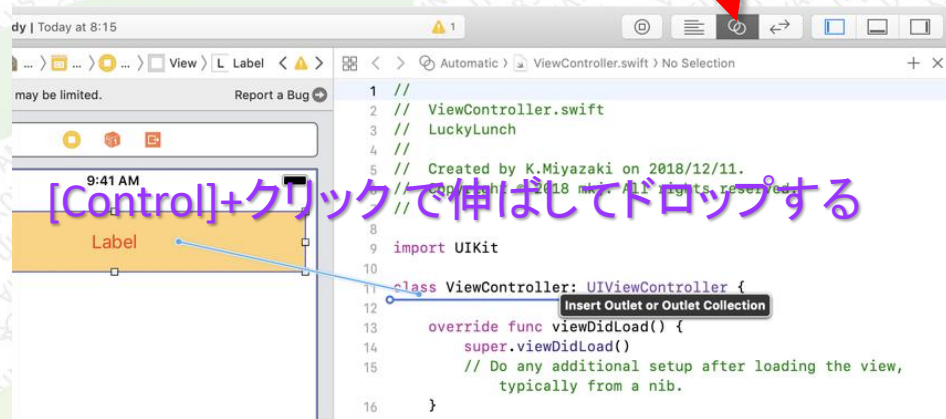
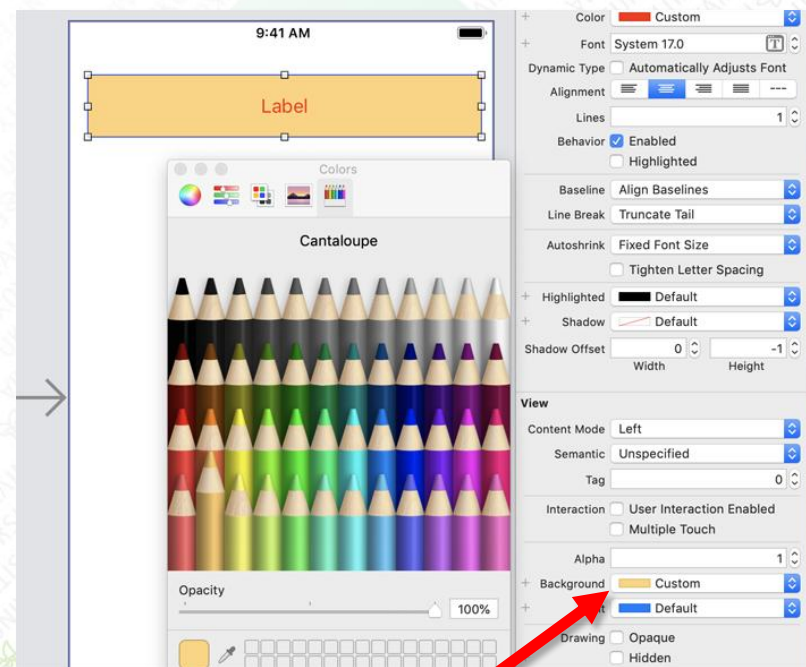
ラベルをプログラムに接続する

[Control]+クリックで伸ばしてドロップする

アシスタントエディタ

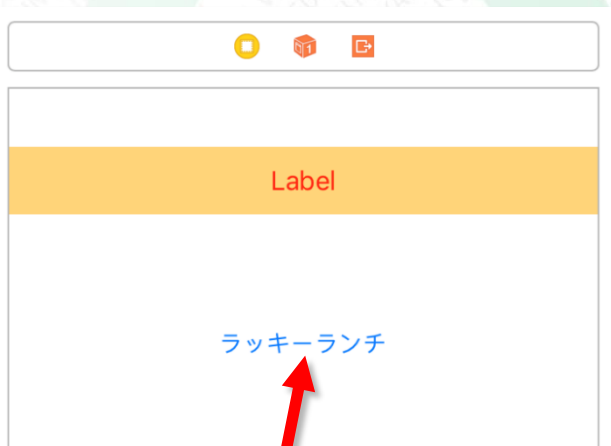
背景色を変更

「myLabel」と入力



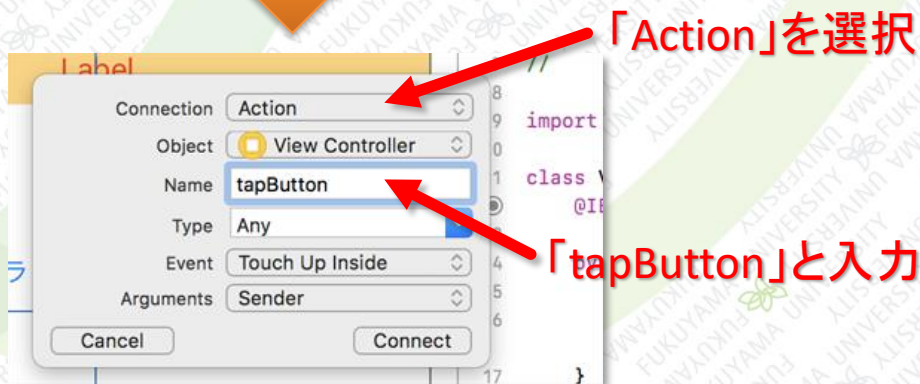
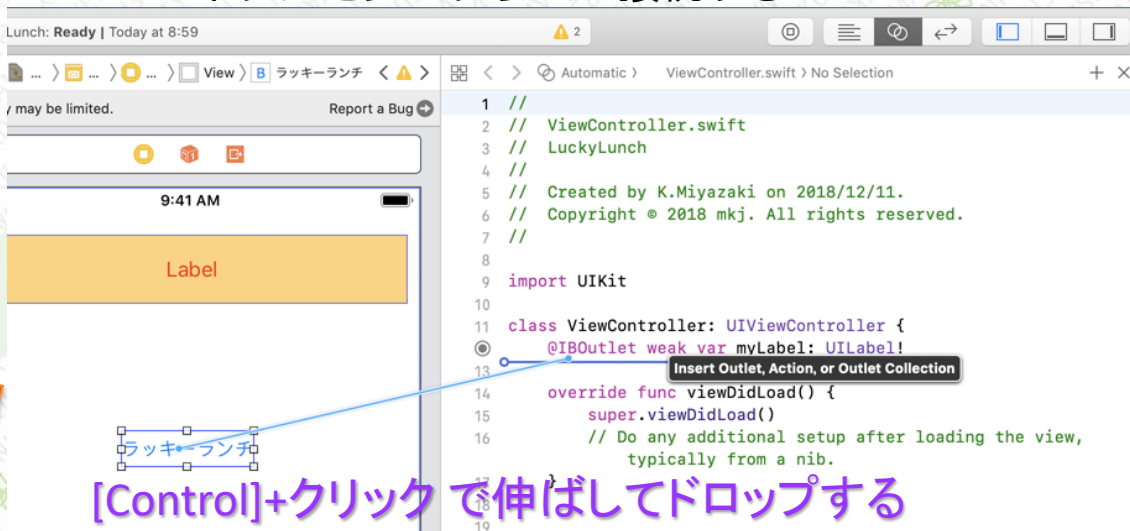
アプリ作成例（１）

・ ボタンを配置する



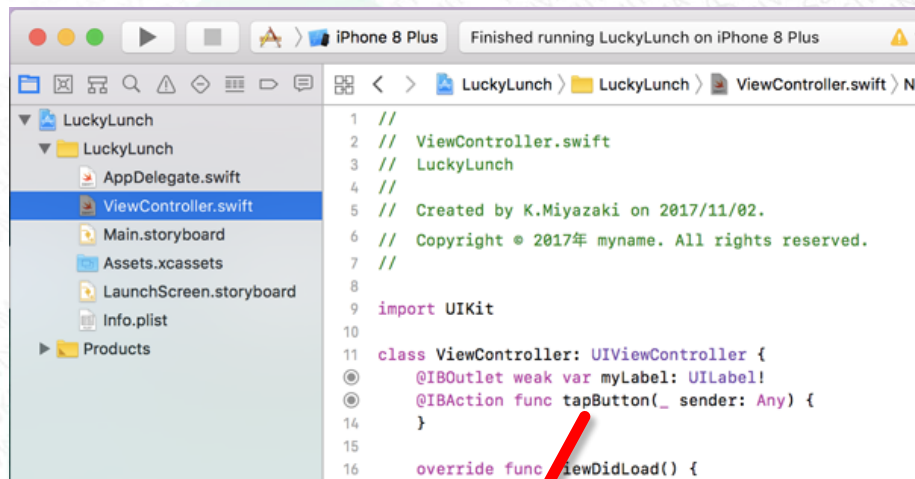
ボタンを設置、
「ラッキーランチ」に変更する

ボタンをプログラムに接続する

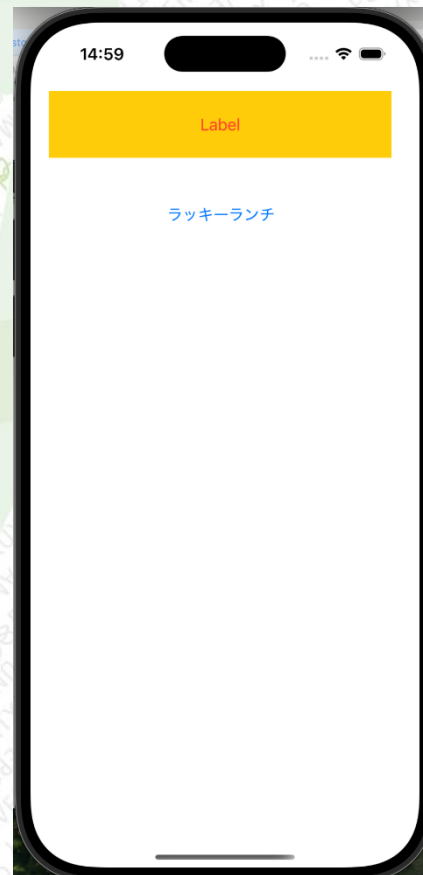


アプリ作成例（１）

・プログラムを入力する



実行してみよう



```
11 class ViewController: UIViewController {
12     @IBOutlet weak var myLabel: UILabel!
13     @IBAction func tapButton(_ sender: Any) {
14         let items = ["弁当", "パン", "カレー", "パスタ", "うどん"]
15         let r = Int(arc4random()) % Int(items.count)
16         myLabel.text = items[r]
17     }
18
19     override func viewDidLoad() {
20         super.viewDidLoad()
```

「tapButton」関数の中に、上記のように入力する

アプリ作成例（２）

- いろいろな部品

アプリ作成例（２）

- いくつかの部品を利用するアプリを作ってみる



Label Label - A variably sized amount of static text.

・・・文字列を表示する

Button Button - Intercepts touch events and sends an action message to a target object when it's tapped.

・・・タップすると、iPhoneの画像が切り替わる

Switch Switch - Displays an element showing the boolean state of a value. Allows tapping the control to toggle t...

・・・オン・オフを切り替える、ラベルに状態を表示する

Slider Slider - Displays a continuous range of values and allows the selection of a single value.

・・・つまみで値を変更する、値をラベルに状態を表示する

Text Text Field - Displays editable text and sends an action message to a target object when Return is tapped.

・・・文字列を入力すると、そのままラベルに表示する

Image View Image View - Displays a single image, or an animation described by an array of images.

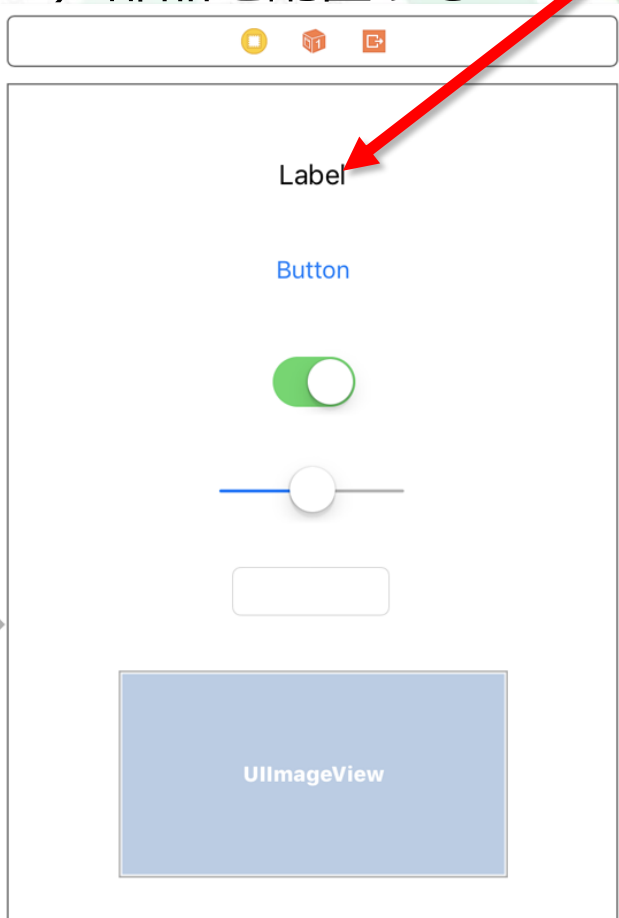
・・・画像の表示、上部「button」をタップすると切り替わる

アプリ作成例（2）

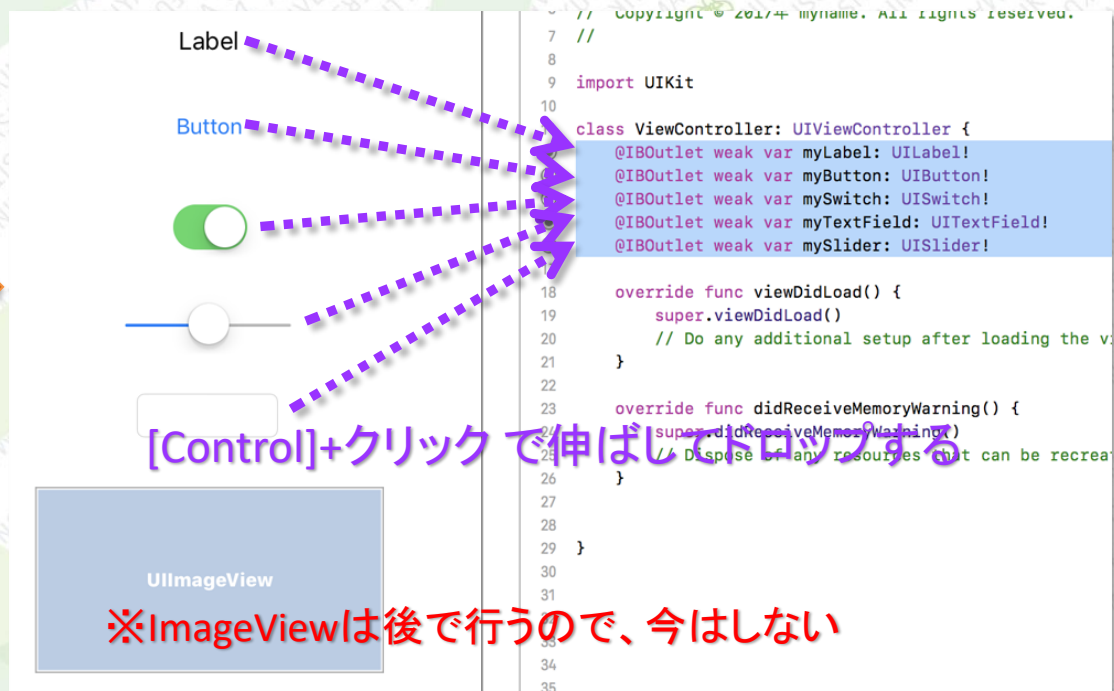
- 1) プロジェクトを新規作成する
プロジェクト名は「Demo」にする

- 2) 部品を配置する

Labelは横幅を少し広げておく



- 3) 部品とプログラムを接続する



アプリ作成例（2）

- ボタンをタップしたときの処理を追加

[Control]+クリック で伸ばしてドロップ



「Action」を選択

「myButton」と入力



入力する

```
19 @IBAction func myButton(_ sender: Any) {
20     myLabel.text = "ボタンが押された"
21 }
```

アプリ作成例（2）

- ・スイッチを切り替えたときの処理を追加

[Control]+クリックで伸ばしてドロップ

「Action」を選択

「mySwitch」と入力

入力する

```
@IBOutlet weak var mySwitch: UISwitch!
@IBOutlet weak var myText: UILabel!
@IBOutlet weak var mySlider: UISlider!

@IBAction func myButton(_ sender: Any) {
    myLabel.text = "ボタン"
}

override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view
}
```

```
@IBOutlet weak var myImageview: UIImageView!
@IBAction func myButton(_ sender: Any) {
    myLabel.text = "ボタンが押された"
}

@IBAction func mySwitch(_ sender: Any) {
    if mySwitch.isOn==true {
        myLabel.text = "SWオン💡"
    } else {
        myLabel.text = "SWオフ"
    }
}

override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view
}
```

アプリ作成例（2）

- ・ スライダーの処理を追加

[Control]+クリックで伸ばしてドロップ

「Action」を選択

「mySlider」と入力

入力する

```
17  @IBAction func myButton(_ sender: UIButton) {
18      myLabel.text = "ボタンが押された"
19  }
20
21  @IBAction func mySwitch(_ sender: UISwitch) {
22      if mySwitch.isOn == true {
23          myLabel.text = "スイッチがオン"
24      } else {
25          myLabel.text = "スイッチがオフ"
26      }
27  }
28
29  override func viewDidLoad() {
30      super.viewDidLoad()
31      // Do any additional setup after loading the view.
32  }
33
34  override func didReceiveMemoryWarning() {
35      super.didReceiveMemoryWarning()
36  }
```

```
29  @IBAction func mySlider(_ sender: Any) {
30      myLabel.text = String(mySlider.value)
31  }
```

アプリ作成例（2）

- ・テキストフィールドの処理を追加

[Control]+クリックで伸ばしてドロップ

UIImageView

```
22     if mySwitch.isOn == true {
23         myLabel.text = "ON"
24     } else {
25         myLabel.text = "OFF"
26     }
27 }
28 @IBAction func mySlider(_ sender: UISlider) {
29     myLabel.text = String(format: "%d", Int(sender.value))
30 }
31 override func viewDidLoad() {
32     super.viewDidLoad()
33     // Do any additional setup after loading the view.
34 }
```

Insert Outlet, Action, or Outlet Collection

Connection: Action
Object: View Controller
Name: myTextField
Type: Any
Event: Did End On Exit
Arguments: Sender

Cancel Connect

Did End On Exit
Editing Changed
Editing Did Begin
Editing Did End
Primary Action Triggered
Touch Cancel
Touch Down
Touch Down Repeat
Touch Drag Enter
Touch Drag Exit
Touch Drag Inside
Touch Drag Outside
Touch Up Inside
Touch Up Outside
Value Changed

```
32 myLabel.text = myTextField.text
33 }
```

入力する

「Action」を選択

「myTextField」と入力

「Event」は「Did End On Exit」を選択する

アプリ作成例（２）

・ ここで、動作確認をしてみよう

- ・ ボタンをタップする
- ・ スイッチを切り替える
- ・ スライドのつまみを移動してみる
- ・ テキストフィールドに文字列を入力してみる

アプリ作成例（2）

- ・ 画像ファイルをプロジェクトに追加する

- 以下 4 個の画像ファイルを使う

ダウンロードしてMacに保存し、
右図の操作を行う



iPhone_b.png



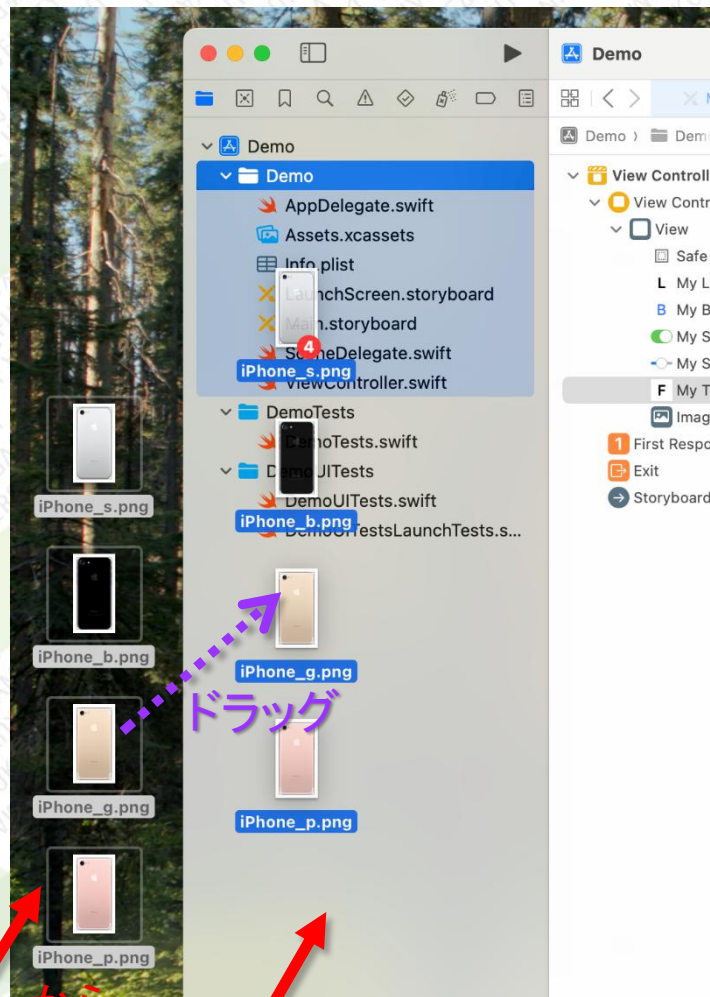
iPhone_s.png



iPhone_p.png



iPhone_g.png



Mac Finderから

Xcode ナビゲータエリア へ

アプリ作成例 (2)

チュエック項目に注意

(2) クリック

(3) 「iPhone_b.png」を設定

(4) 「Aspect Fit」を選択

(5) ドラッグして適当な大きさに調整する

追加された

(1) 「UIImageView」を選択

Choose options for adding these files:

Action: Copy files to destination

Targets:

- ☒ Demo
- ☐ DemoTests
- ☐ DemoUITests

Con...ler Scene > View Controller > View > iPhone_b.png

Image View

Image iPhone_b.png

Highlighted Highlighted Image

State ☐ Highlighted

Accessibility ☐ Adjusts Image Size

View

Content Mode Aspect Fit

Semantic Unspecified

Tag 0

Interaction ☐ User Interaction Enabled

☐ Multiple Touch

Alpha 1

Background

Tint Default

Drawing ☒ Opaque

☐ Hidden

☒ Clears Graphics Context

☐ Clip to Bounds

☒ Autorelease Subviews

Stretching 0 0

Width Height

アプリ作成例（2）



[Button]がタップされたら、画像が切り替わるようにプログラムを追加する



```
8
9  import UIKit
10
11 class ViewController: UIViewController {
12     let fname = ["iPhone_b", "iPhone_g", "iPhone_p", "iPhone_s"]
13     var pic_no: Int = 1
14     @IBOutlet weak var myLabel: UILabel!
15     @IBOutlet weak var myButton: UIButton!
16     @IBOutlet weak var mySwitch: UISwitch!
17     @IBOutlet weak var myTextField: UITextField!
18     @IBOutlet weak var mySlider: UISlider!
19     @IBOutlet weak var myImageView: UIImageView!
20
21     @IBAction func myButton(_ sender: Any) {
22         // myLabel.text = "ボタンが押された"
23         myLabel.text = fname[pic_no]
24         let fn = "\(fname[pic_no]).png"
25         myImageView.image = UIImage(named: fn)
26         pic_no += 1
27         if pic_no == 4 {
28             pic_no = 0
29         }
30     }
31
32     @IBAction func mySwitch(_ sender: Any) {
33         if mySwitch.isOn == true {
```

入力する

「myButton()」
関数に入力する



アプリ作成例（２）

- ・ 動作確認をしてみよう！

- ・ ボタンをタップすると、iPhoneの画像が切り替わる・・・



アプリ作成例（3）

- 複数の画面

アプリ作成例（3）

[別画面で表示]をタップしたら、別画面でiPhoneが表示されるようにする



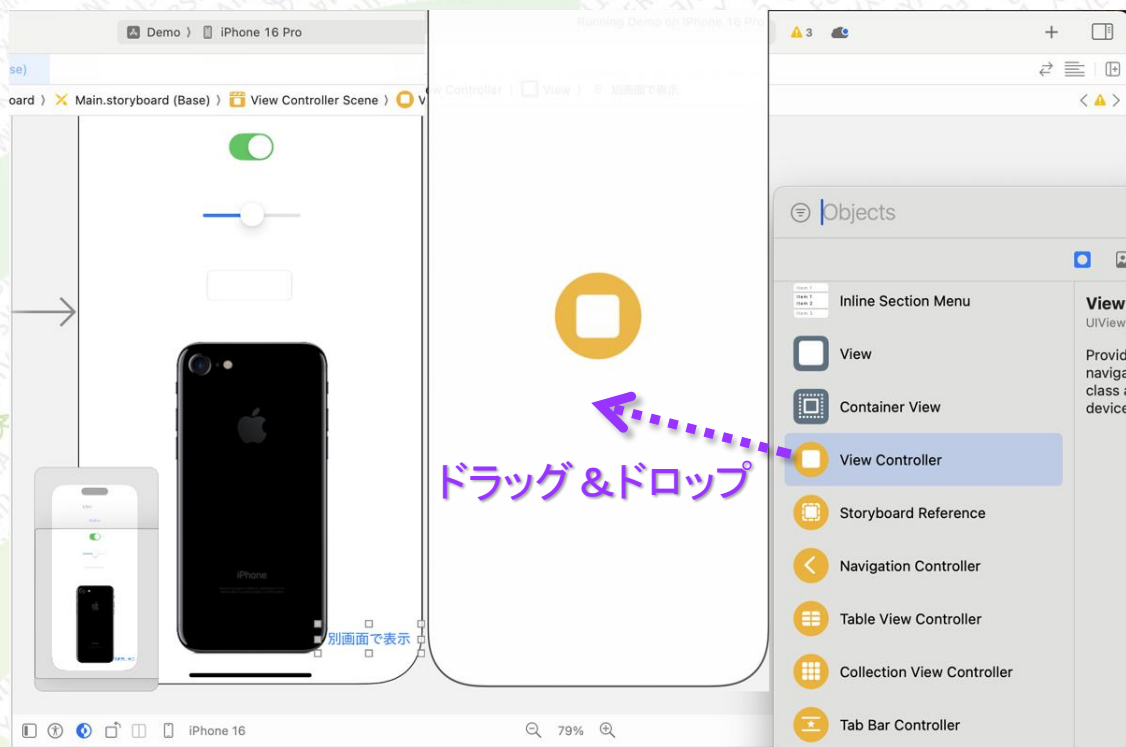
アプリ作成例（3）

- ・ ボタンを追加する

- ・ ViewControllerを追加する

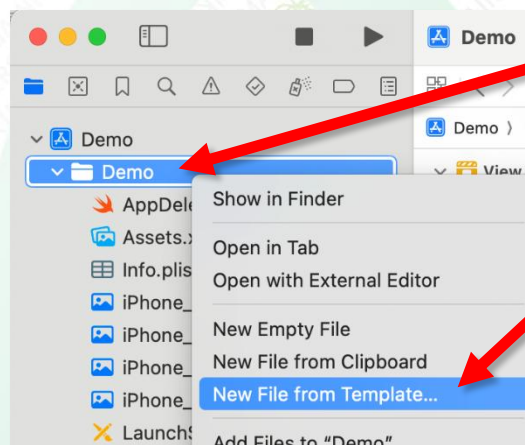


Buttonを追加、
「別画面で表示」に修正



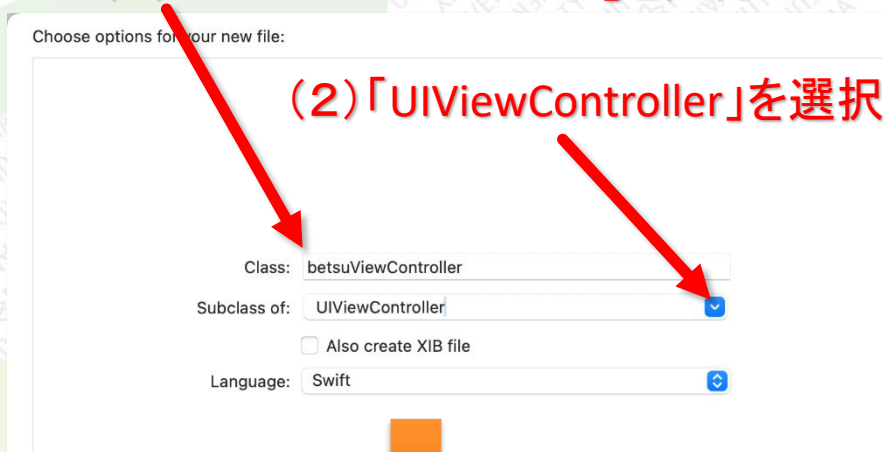
アプリ作成例（3）

- 追加したViewController用のプログラムファイル「betsuViewController.swift」を作る

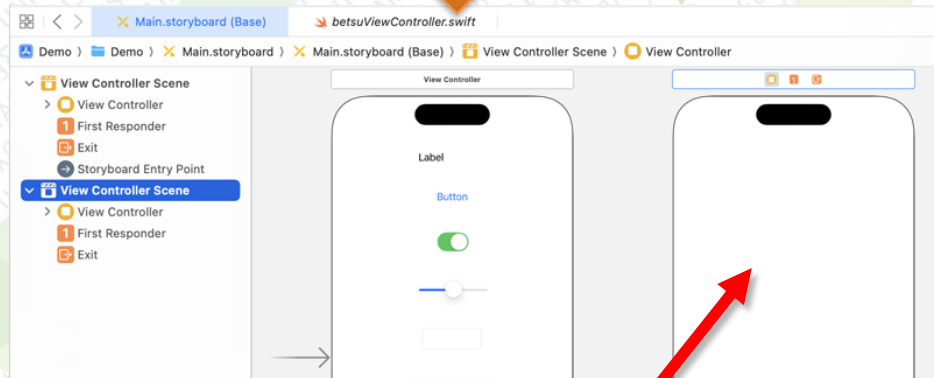
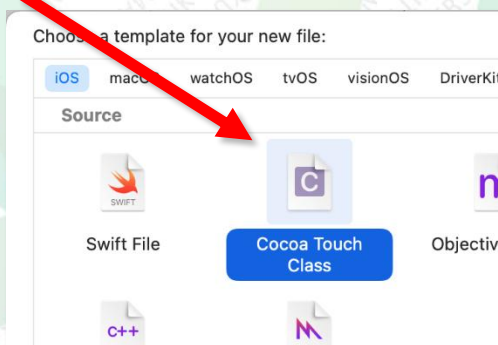


(1)「betsuViewController」と入力

(2)「UIViewController」を選択



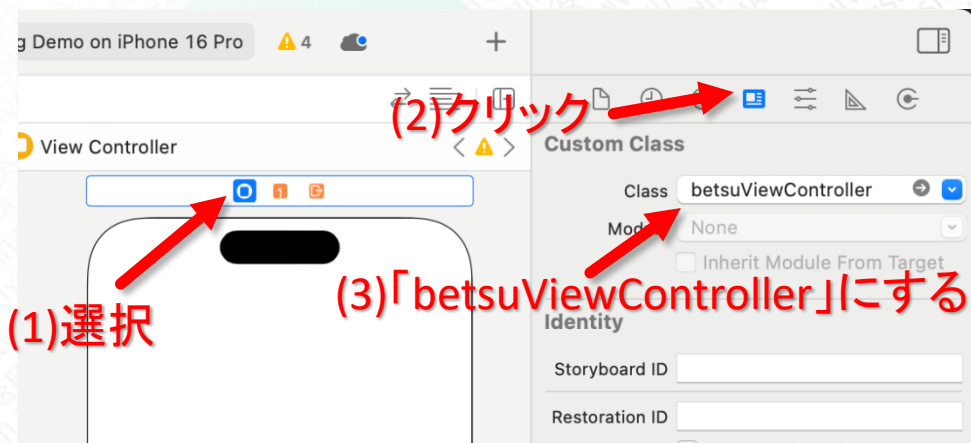
選択



追加された

アプリ作成例 (3)

- betsuViewControllerを2つ目の画面に設定する



- [別画面...]ボタンから2つ目の画面へつなぐ

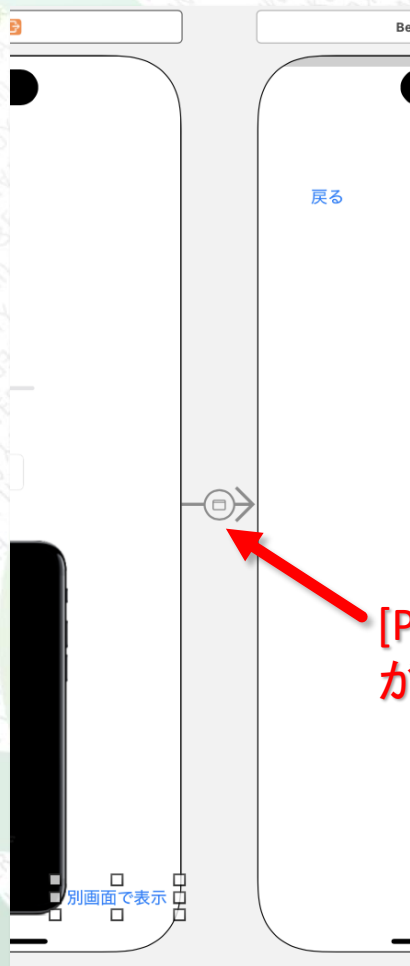


- [戻る]のボタンを追加する

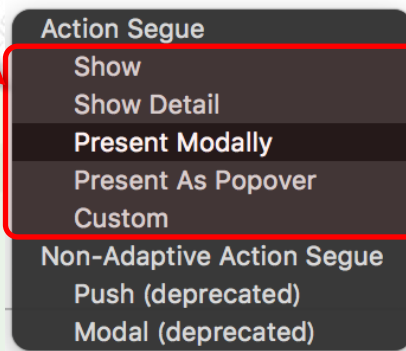


アプリ作成例（3）

- 画面の切り替えは「**segue(セグエ)**」を利用
 - segueは音楽用語で「ある楽章から次の楽章へ切れ目なく移行すること」



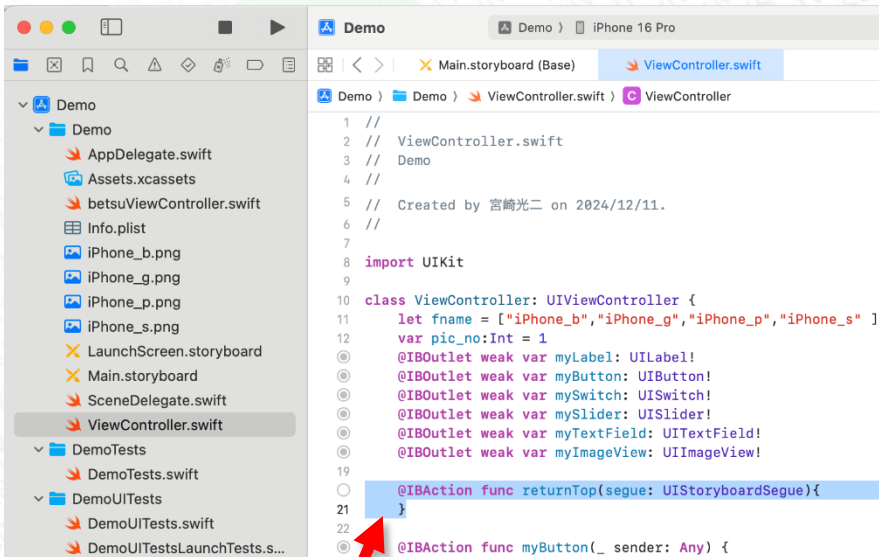
segueの種類はいくつかある



[Present Modally]のsegue
が設定されている

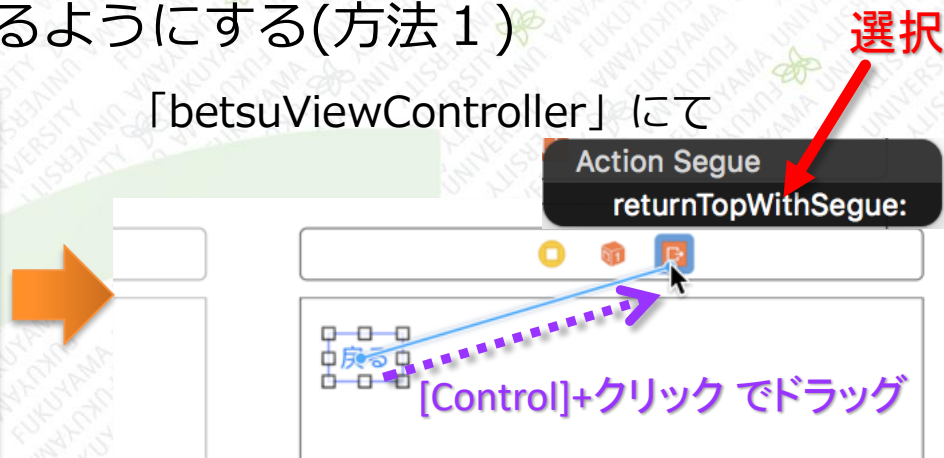
アプリ作成例 (3)

- 2つ目の画面から、元の画面に戻るようになる(方法1)

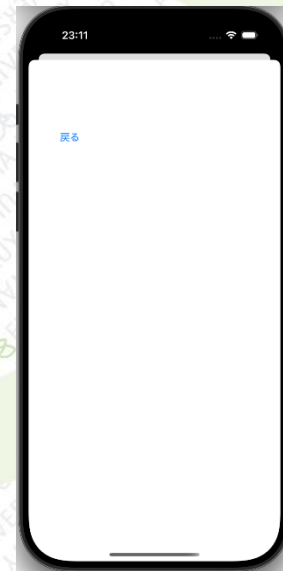
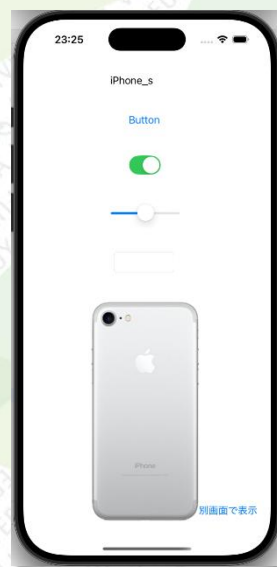


「ViewController.swift」に手入力で打ち込む
これが戻る窓口となる

「betsuViewController」にて



実行して確認してみよう

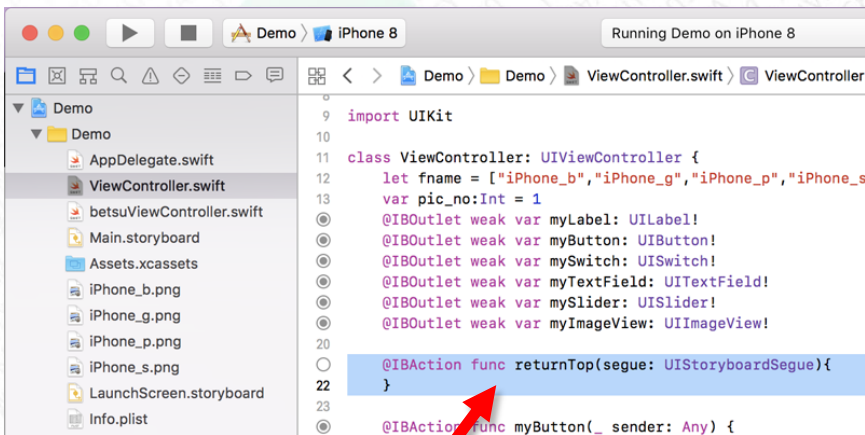
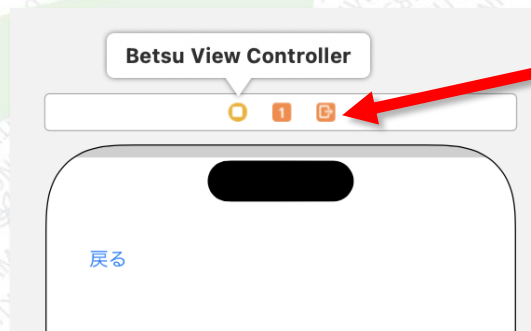


アプリ作成例 (3)

- 2つ目の画面から、元の画面に戻るようにする(方法2)

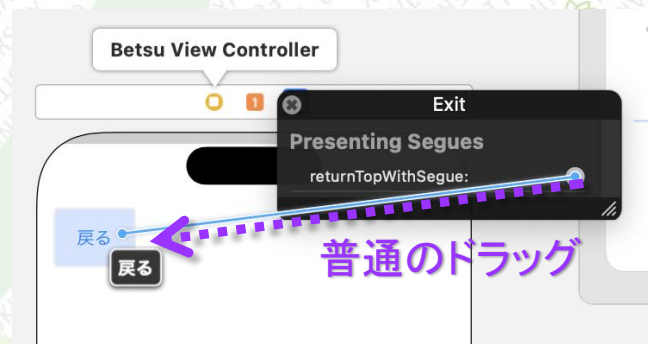
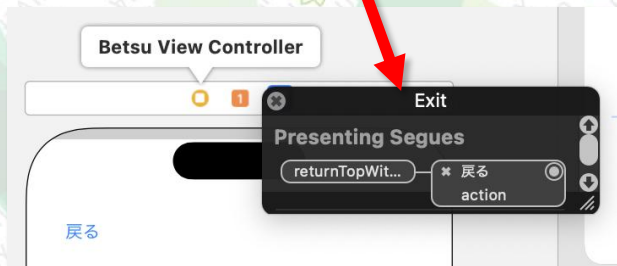
「betsuViewController」にて

右クリック



「ViewController.swift」に手入力で打ち込む
これが戻る窓口となる

こうなればOK



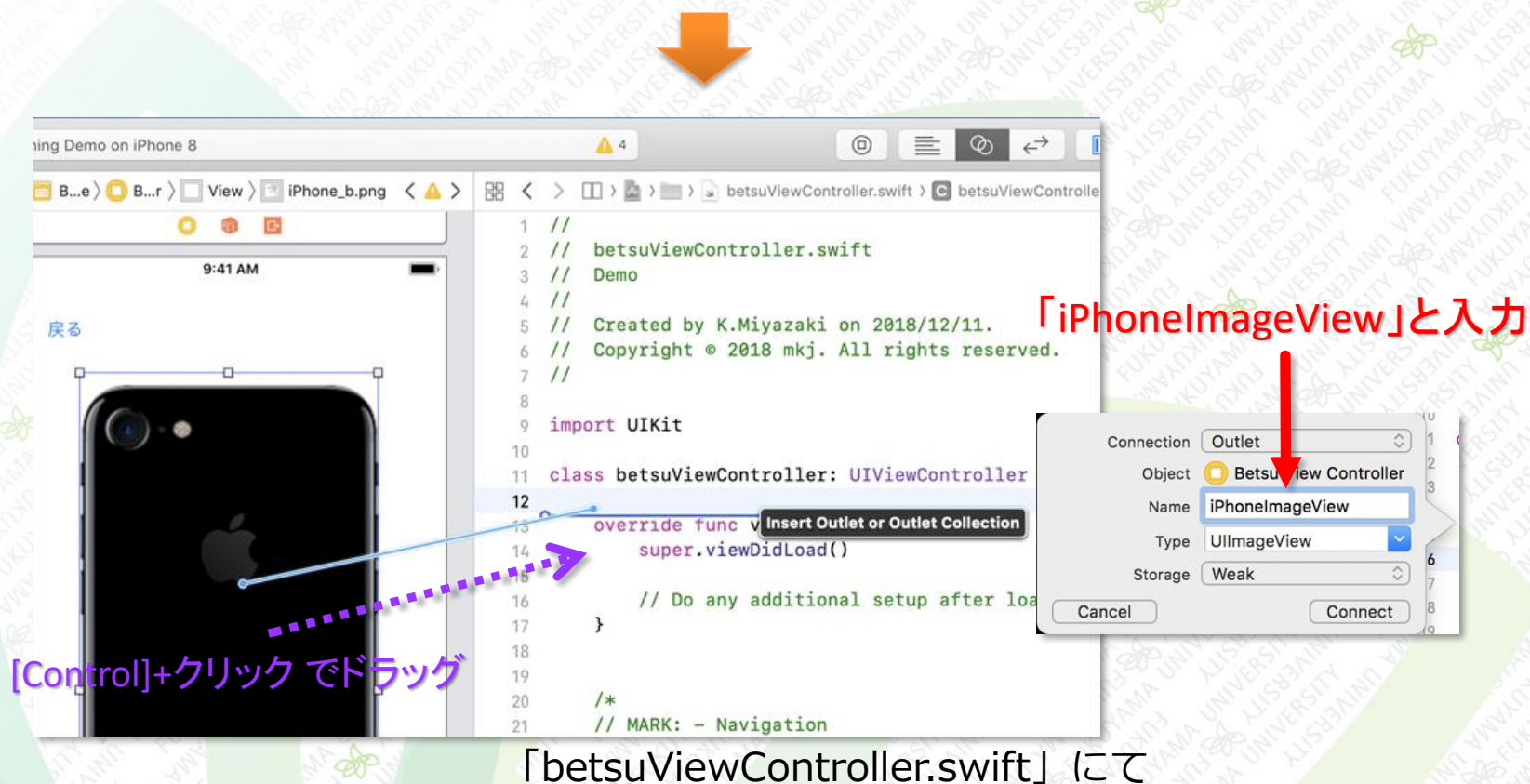
アプリ作成例 (3)

- 2 つ目の画面に画像(iPhone)を大きく表示する



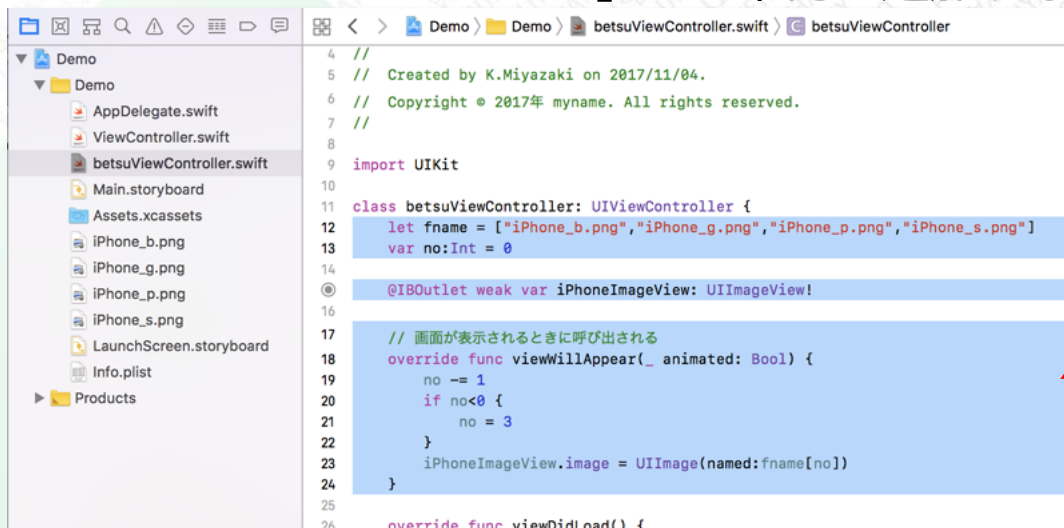
「betsuViewController」にて

アプリ作成例（3）



アプリ作成例（3）

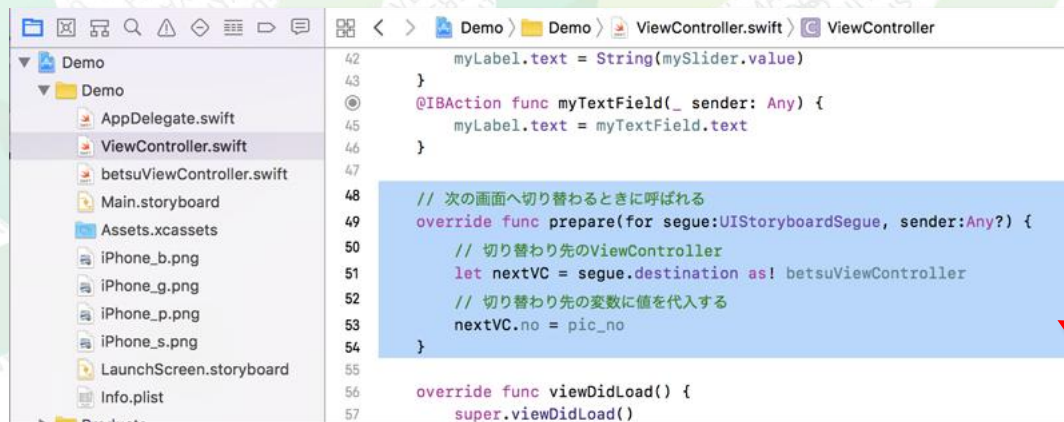
- 「betsuViewController.swift」に、下記を追加する



※viewWillAppear()メソッド
は画面が表示されるときに
呼ばれる

入力する

- 「ViewController.swift」に、下記を追加する



※prepare()メソッドは画面が
切り替わるときに呼ばれる、
2つ目の画面への値の受け
渡しもここで行う

入力する

アプリ作成例（3）

- 1つ目の画面から2つ目の画面へ値を渡すには

< 1つ目の画面では prepare() メソッドを使う >

```
48 // 次の画面へ切り替わるときに呼ばれる
49 override fun prepare(for segue: UIStoryboardSegue, sender: Any?) {
50     // 切り替わり先のViewController
51     let nextVC = segue.destination as! BetsuViewController
52     // 切り替わり先の変数に値を代入する
53     nextVC.no = pic_no
54 }
```

nextVC が 2つ目の ViewController
のオブジェクトとなる

2つ目の画面オブジェクトの
変数 no に pic_no の値を代入する

変数 no に値をコピーしてから
2つ目の画面が表示される

< 2つ目の画面では viewWillAppear() メソッドにて処理する >

```
12 let fname = ["iPhone_b.png", "iPhone_g.png", "iPhone_p.png", "iPhone_s.png"]
13 var no: Int = 0
```

```
17 // 画面が表示されるときに呼び出される
18 override fun viewWillAppear(_ animated: Bool) {
19     no -= 1
20     if no < 0 {
21         no = 3
22     }
23     imageView.image = UIImage(named: fname[no])
24 }
```

アプリ作成例（3）

- ・これで完成！

