

ようこそ！ Swift Playgroundsへ



まんくまデザイン

2014/10/24 @【#techbuzz】第4回 Swift 勉強会



フロントエンドエンジニア
(2008 年よりフリーランス)

仕事では営業用とか教育用のツール
アプリ制作が多いです

いろいろやっています。
Web: JavaScript, TypeScript
iOS: Objective-C, Swift
Flash/AIR: ActionScript3/2

あと、クリエイティブ系のセミナーを開催したりす
る F-site のスタッフをやってます
<http://f-site.org/>
※来月 11/08 に開催します (ステマ)

ブログ
<http://www.kuma-de.com/>

Twitter: @kinkuma_design

日課管理・記録アプリ

Habit Keeper



AppStore で好評発売中！！

<https://itunes.apple.com/jp/app/id567646997?l=ja&mt=8>

無料版もあります！

<https://itunes.apple.com/jp/app/id602525022?l=ja&mt=8>

おかげさまで有料＋無料合わせて
5 万ダウンロードされました！

ここから本題



Swift Playgrounds って何？

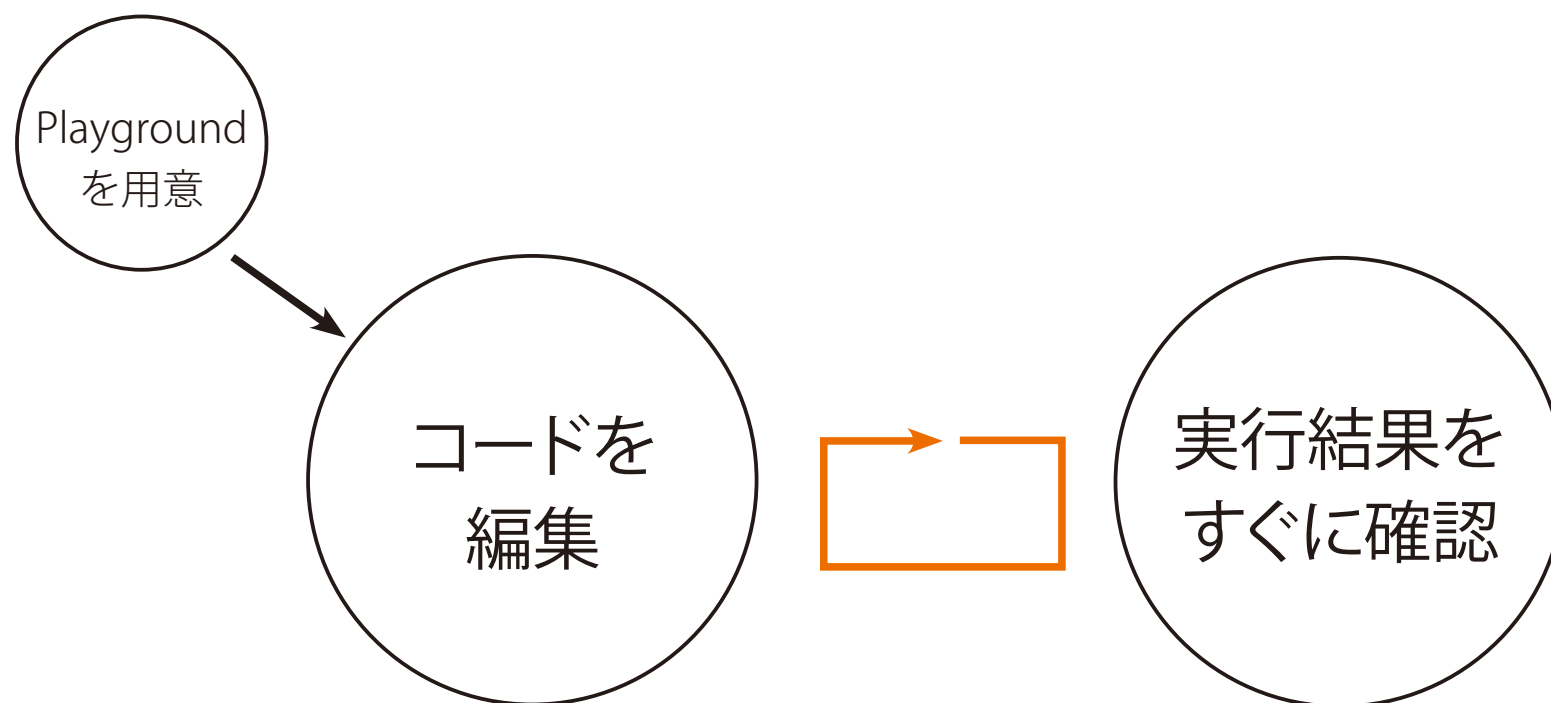
Xcode6 から可能になった swift の実行環境

例えたのなら？

- swift
- Cocoa (OS X)
- Cocoa Touch (iOS)

の遊び場、実験ひろば

Swift Playgrounds って何？



早いサイクルでコードの実験&結果確認ができる
いらなくなったら気軽にゴミ箱へ

Swift Playgrounds って何？

- playground ファイル単体で
 - プロジェクトに追加する形で
- どちらも実行可能

コードを書いているときにちょっと試してみたいことを、
プロジェクトに直接記述せずに
気軽に確認＋実験ができる！！

Swift Playgrounds って何？

こんなときに便利！

文法の書き方

swift で配列の書き方ってどうやるんだっけ？

あれ？ let や var の違いって？

プレビュー

UITableViewCell ってどんなタイプがあるんだっけ？

UIColor のこの色ってどんな色？

インスタンスのプロパティの中身って？



大まかな機能は 3 種類

The Timeline

値をキャプチャしてグラフ化

Quick Looks

プレビュー＋値出力

XCPlayground

Apple 製の Playgrounds 用のモジュール
(追加機能みたいなもの。使わなくても OK)

The Timeline



swiftちゃん

The Timeline

値をキャプチャしてグラフ化

TheTimeline.playground > No Selection

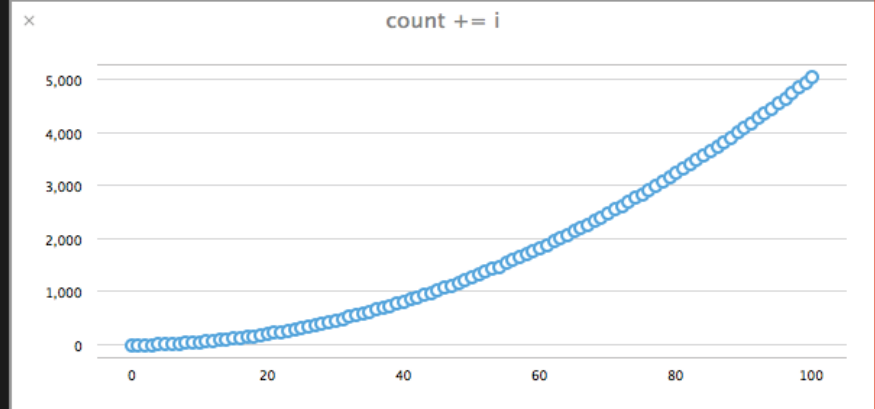
```
1 // Playground - noun: a place  
2   where people can play  
3 import UIKit  
4 var count = 0  
5  
6  
7 for i in 0 ... 100{  
8   count += i  
9 }  
10
```

0

(101 times)

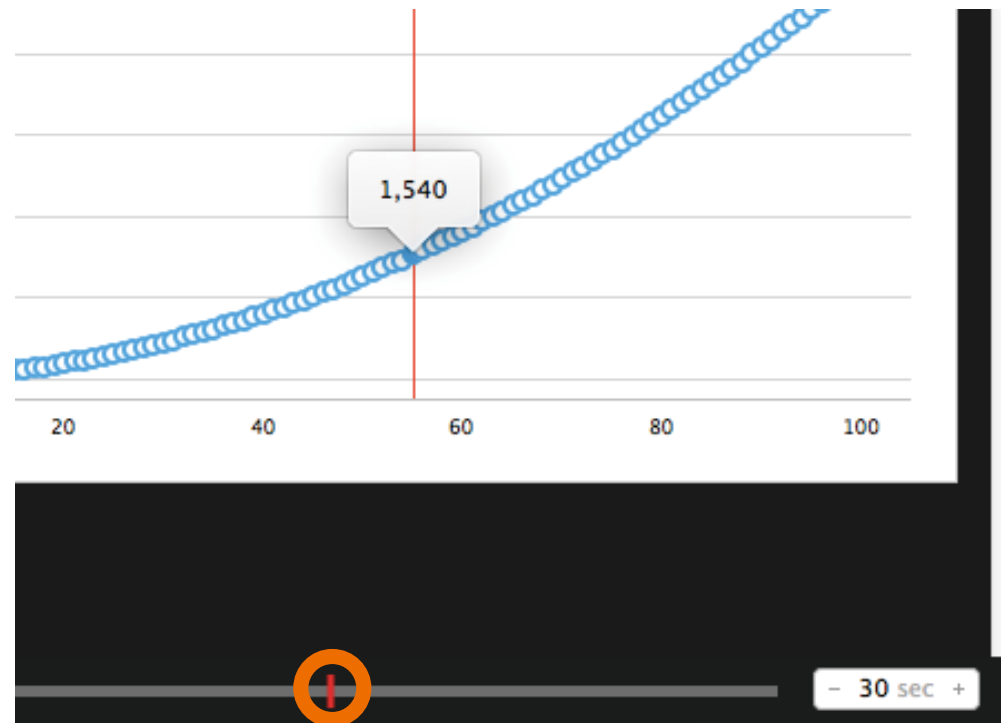


Timeline > TheTimeline.playground (Timeline)



The Timeline

右下のスライダを動かすとグラフ上の値が確認できる



■ The Timeline

デモ

Quick Looks



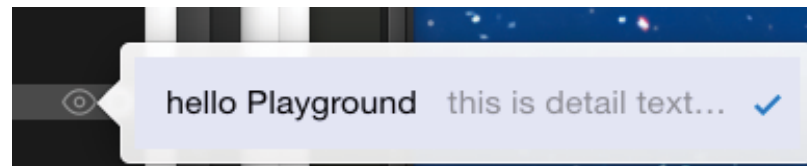
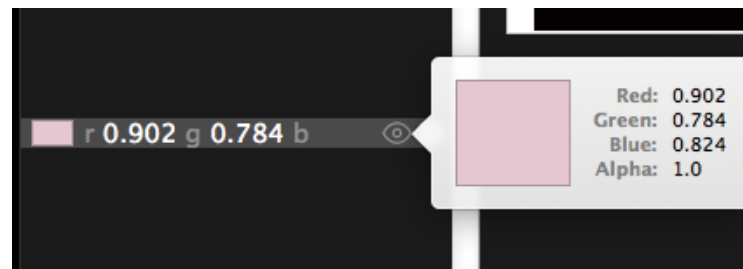
swiftちゃん

Quick Looks

プレビュー+値出力

いろんなものを**その場で確認**できる

- 色
- 文字列
- 画像
- ビュー (UIView)
- 配列+辞書
- Points, rects, sizes
- ベジェパス
- URL
- クラスと構造体 などなど



Quick Looks

表示できるタイプ一覧

Operating System Types Supporting debugQuickLookObject

https://developer.apple.com/library/IOs/documentation/IDEs/Conceptual/CustomClassDisplay_in_QuickLook/CH02-std_objects_support/CH02-std_objects_support.html#//apple_ref/doc/uid/TP40014001-CH3-SW1

また Quick Looks はどんな風に見せるのか カスタマイズも可能

About Variables Quick Look for Custom Types

https://developer.apple.com/library/IOs/documentation/IDEs/Conceptual/CustomClassDisplay_in_QuickLook/Introduction/Introduction.html

Swift: Adding custom quick look to your classes

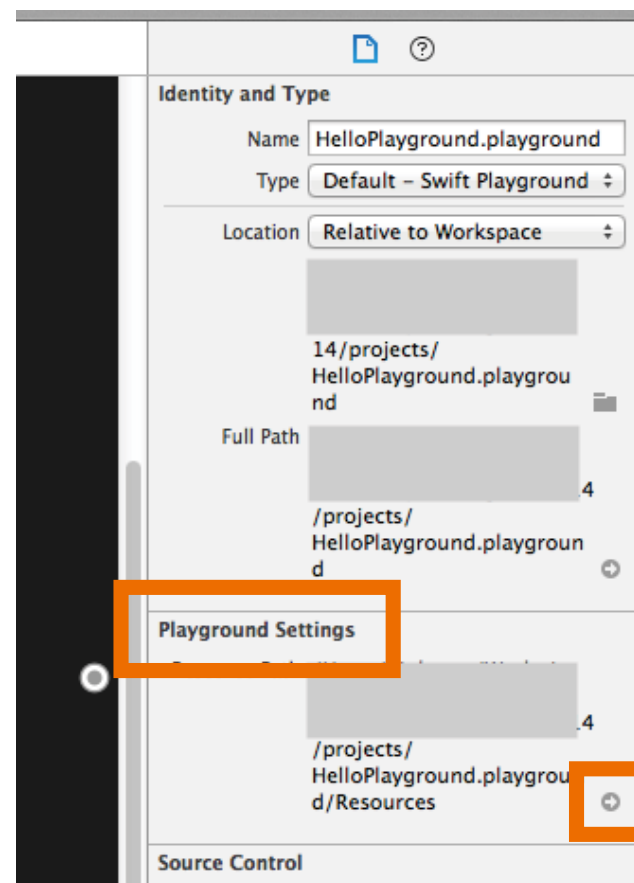
<http://ericasadun.com/2014/06/26/swift-adding-custom-quick-look-to-your-classes/>

Playground への画像の追加方法

View > Utilities > Show File Inspector

Playground Settings の矢印を
クリックして Finder の中を開く

そこに追加すれば OK



■ Quick Looks

デモ

XCP playground



swiftちゃん

XCPlayground

Apple 製の Playgrounds 用のモジュール

- XCPCaptureValue

手動での値キャプチャ

- XCPShowView

ライブビュー

- XCPSetExecutionShouldContinueIndefinitely

非同期実行

XCPlayground

使う前に、上のところに

```
import XCPlayground
```

と書いておく

XCPCaptureValue

手動での値キャプチャ

```
func XCPCaptureValue<T>(identifier: String, value: T)
```

identifier でひとつのグラフにすることができる

自動だと、値の流れがぶつ切りになってしまうところも
つなげることができるようになる。

XCPCaptureValue

```
import UIKit
import XCPlayground

//XCPCaptureValue
// 手動で値をとった場合

let selfCapId = "manual_capture"
var n = 0
n = 800
XCPCaptureValue(selfCapId, n)
n = 30
XCPCaptureValue(selfCapId, n)

for i in 0 ... 30{
    n += i
    if n % 3 == 0 {
        XCPCaptureValue(selfCapId, n)
    }
}
```

XCPShowView

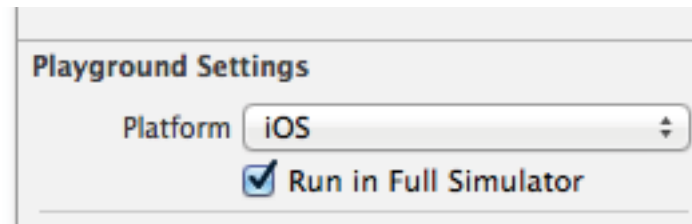
SpriteKit や View のアニメーションを
Playground で確認できる

ただし、Xcode6.0.1 の時点では UIView は未対応。
OSX の NSView は対応している。

~~Xcode6.1 では UIView の対応ができているらしい (未確認)~~

-> Xcode6.1 がリリースされて対応が確認できました。

File Inspector の Run in Full Simulator
を有効にする必要あり



XCPShowView

```
import UIKit
import XCPlayground

// アニメーション
//Run in Full Simulatorを有効にする必要あり
var animContainerView = UIView(frame: CGRectMake(0, 0, 300, 300))
XCPShowView("animationView", animContainerView)

var animView = UIView(frame: CGRectMake(0, 0, 30, 30))
animView.backgroundColor = UIColor.blueColor()
animContainerView.addSubview(animView)

UIView.animateWithDuration(1.0
    , animations: { () -> Void in
        var pos = animView.frame.origin
        pos.x = 230
        pos.y = 230
        animView.frame.origin = pos

    }, completion: { (success:Bool) -> Void in
        println("complete!!")
    })
})
```

XCPSetExecutionShouldContinueIndefinitely

非同期実行

```
func XCPSetExecutionShouldContinueIndefinitely(conti  
nueIndefinitely: Bool = default)
```

(関数名長い!)

普通は上からプログラムを順番に実行してその場で終わってしまうが、時間のかかる処理を待つことができる

XCPSetExecutionShouldContinueIndefinitely

```
import UIKit
import XCPlayground
class SampleTimer{
    ... 省略

    func startTimer(){
        stopTimer()
        timer = NSTimer.scheduledTimerWithTimeInterval(1.0, target: self, selector: "onTimerTick",
userInfo: nil, repeats: true)
    }

    ... 省略
}

XCPSetExecutionShouldContinueIndefinitely(continueIndefinitely: true)
let timer = SampleTimer()
timer.startTimer()
```

■ XCPlayground

デモ

注意事項＋まとめ



swiftちゃん

■ 気をつけること

パフォーマンステストには用いないようにする
やりたいときは XCTest を利用する

(WWDC2014 Swift Playgrounds より)

今はできないこと (制限事項)

- ユーザーインタラクション (マウスとかタッチとか)
- Entitlements (Push Notifications とかのあたり)
- 実機での実行
- 自作の Framework

(WWDC2014 Swift Playgrounds より)

まとめ

- P** 書いたものがすぐに実行できる
Playground は**便利**！
- P** ファイル 1 枚で完結するので**気軽に**作ろう！
- P** 文の書き方やロジックなど、
気になるところは**すぐ試そう**！



参考リンク

WWDC2014 Swift Playgrounds

<https://developer.apple.com/videos/wwdc/2014/>

XCPlayground Module Reference

https://developer.apple.com/library/ios/documentation/Swift/Reference/Playground_Ref/Chapters/XCPlayground.html

[Swift] グラフもレンダリングもイける！

Playgroud が超絶便利！

<http://www.dorapro.co.jp/engineerblog/?p=137>

ありがとう
ございました！



swiftちゃん