

【iOS開発】Swiftで簡易PDFビューワを作成
(PDFを読み込み、表示)

2017/06/02 | category:[iOS](#), [技術](#)



突然ですが、仕事で急遽必要になったため、Swiftで簡易pdfビューワを作りました。

実際はもう少し色々機能があったのですが、「PDFを表示する」という目的に絞ってスリム化しました。

ネットでpdfを表示するような記事はヒットするんですが、古いものが多かったので一応Swift3に対応しているものとして、参考となれば幸いです。

機能としては以下のようなものです。

- ①初期画面に「PDF表示」のボタンを配置し、クリックすると「PDF閲覧画面」に遷移。
- ②PDF閲覧画面にてPDFを表示。ページ切り替えは上下のスワイプにて行う。

一応作り方を簡単にご紹介しておきます。

目次 [\[非表示\]](#) [\[非表示\]](#)

- 1 プロジェクトの作成
- 2 PDFの表示
- 3 storyboard
- 4 動作確認

プロジェクトの作成

まずは新規で「プロジェクト」を作成します。

Xcodeを立ち上げ、「Create a new Xcode project」をクリックします。



Profile



名前：Kerubito

大阪でフリーランスエンジニアをメインに活動していますが、自由な生き方をするため、収入の多角化を目指しています。



詳しいプロフィール

お問い合わせ

お問い合わせはこちら

最近の投稿

香港旅行をお勧めする6つの理由 2017年5月31日

香港を何倍も楽しむためにタクシー、地下

次はプロジェクトのタイプを選択します。

「Single View Application」 をクリックします。

あとはアプリやデベロッパの名称、言語、対象デバイスを設定します。今回はiPad向けです。

PDFの表示

下準備が終わったら、次は実際にPDFを表示するロジックを実装していきます。

以下のような構成です。

ViewController.swift

```
import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {

        super.viewDidLoad()

        // Do any additional setup after loading the view, typically from

    }

    override func didReceiveMemoryWarning() {

        super.didReceiveMemoryWarning()

        // Dispose of any resources that can be recreated.

    }

}
```

ViewController.swiftはプロジェクトを作成すると、自動的に生成されます。

今回はViewController.swiftには一切手を加えていません。

PDFShowViewController.swift

```
import UIKit
```

鉄、バス、トラムを使いこなそう 2017年5月29日

何度も訪れたい香港（香港島エリア）の安くて美味しい厳選グルメ5選 2017年5月27日

USBメモリ（USB3.0 Express ST1-2 ST3U32ES12 [32GB]）が価格の割に便利で手放せない 2017年5月25日

フリーランスのエンジニアが奄美大島に仕事をもち込んでみた結果 2017年5月22日

よく読まれている記事



奄美大島で金作原原生林のツアーが凄かった件

旅行記 の下



レンタカーで周りたい独断と偏見で選んだ奄美大島のおすすめ観光スポット

旅行記 の下



奄美大島の名瀬周辺を街歩きしてみた

旅行記 の下



奄美大島からフェリーで加計呂麻島に渡り、レンタカーで島内を周ってみた

旅行記 の下



奄美大島からフェリーで加計呂麻島に渡り、レンタルバイク（原付）で島内を周ってみた

旅行記 の下



奄美大島の漁港で釣りをしていたら肉をもらって野球までした話

旅行記 の下



フリーランスエンジニアが奄美大島に住めそうか現地調査してみた

考察 の下



疑問や質問がない人はエンジニアには向いていない

考察 の下



香港旅行をお勧めする6つの理由

旅行記 の下

フリーランスのエンジニアが奄美大島に仕事をもち込んでみた結果

```
class PDFShowViewController: UIViewController {

    var pages = 1
    var nowPage = 1
    var pdfView = PDFView(frame: CGRect(x: 0, y: 0, width: 0, height: 0))
    let pdfName = ConstStruct.pdf_file_name

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib

        let url = Bundle.main.bundleURL.appendingPathComponent(pdfName)
        let doc = CGPDFDocument(url as CFURL)

        pages = doc!.numberOfPages

        let page = doc!.page(at: nowPage)
        let myBoundSize: CGSize = UIScreen.main.bounds.size
        pdfView = PDFView(frame: CGRect(x: 0, y: 0, width: myBoundSize.width, height: myBoundSize.height))

        pdfView.backgroundColor = UIColor.white
        pdfView.page = page
        self.view.addSubview(pdfView)
    }
    //次ページへ
    @IBAction func downSwiped() {
        if nowPage < pages {
            nowPage = nowPage + 1
            let url = Bundle.main.bundleURL.appendingPathComponent(pdfName)
            let doc = CGPDFDocument(url as CFURL)
            let page = doc!.page(at: nowPage)
            pdfView.page = page
            for subview in self.view.subviews {
                subview.setNeedsDisplay()
            }
        }
    }
    //前ページへ
    @IBAction func upSwiped() {
        if nowPage > 1 {
            nowPage = nowPage - 1
            let url = Bundle.main.bundleURL.appendingPathComponent(pdfName)
            let doc = CGPDFDocument(url as CFURL)
            let page = doc!.page(at: nowPage)
            pdfView.page = page
            for subview in self.view.subviews {
                subview.setNeedsDisplay()
            }
        }
    }

    override func didReceiveMemoryWarning() {
```



アーカイブ

- 2017年5月
- 2017年4月
- 2017年3月

カテゴリー

- Apple
- iOS
- おすすめ
- グルメ
- サービス
- フリーランス
- ライフハック
- 主張
- 仕事術
- 働き方
- 技術
- 旅行記
- 考察
- 雑感

検索

```
super.didReceiveMemoryWarning()

// Dispose of any resources that can be recreated.
}
}
```

PDFShowViewController.swiftを追加します。 このクラスでPDFの表示を行います。

```
let url = Bundle.main.bundleURL.appendingPathComponent(pdfName)
let doc = CGPDFDocument(url as CFURL)

pages = doc!.numberOfPages

let page = doc!.page(at: nowPage)
let myBoundSize: CGSize = UIScreen.main.bounds.size
pdfView = PDFView(frame: CGRect(x: 0, y: 0, width: myBoundSize.wid

pdfView.backgroundColor = UIColor.white
pdfView.page = page
self.view.addSubview(pdfView)
```

この辺りがメインの処理ですが、上から順に

- ①PDFファイルの読み込み
- ②ページ数を読み込んで設定
- ③PDFViewのインスタンス化
- ④PDFViewを画面に表示

といった事をやっています。

PDFViewはUIViewを継承したPDFの表示に特化したクラスです。

PDFView.swift

```
import UIKit

import CoreGraphics

class PDFView: UIView {
    var page: CGPDFPage?

    override func draw(_ rect: CGRect) {
        guard let page = page else { return }
    }
}
```



```

let context = UIGraphicsGetCurrentContext()

context!.translateBy(x: 0, y: rect.size.height)
context!.scaleBy(x: 1.0, y: -1.0)

let box = page.getBoxRect(.artBox)
let xScale = rect.size.width / box.size.width
let yScale = rect.size.height / box.size.height
let scale = min(xScale, yScale)
let tx = (rect.size.width - box.size.width * scale) / 2
let ty = (rect.size.height - box.size.height * scale) / 2

context!.translateBy(x: 0, y: ty)
context!.scaleBy(x: scale, y: scale)

context!.drawPDFPage(page)
}
}

```

viewに何か描画したい場合はdrawメソッドを呼び出します。 コンテキストを取得して、サイズやスケールなどの設定、最後にdrawPDFPageメソッドにてPDFを表示します。

Define.swift

```

import UIKit

struct ConstStruct {
    static let main_color_red: CGFloat = 0.00
    static let main_color_green: CGFloat = 0.40
    static let main_color_blue: CGFloat = 1.00
    static let main_title_color_red: CGFloat = 1.00
    static let main_title_color_green: CGFloat = 1.00
    static let main_title_color_blue: CGFloat = 1.00

    static let pdf_file_name: String = "setsumei.pdf"

}

class Define{
}

```

おまけです。

storyboard

クラスを実装し終わったら、次はstoryboardで画面遷移やアクションなどを定義

していきます。

といっても、今回は非常にシンプルなアプリなのでやる事はそれほどないです。

まずは「View Controller」のオブジェクトをstoryboardに配置します。

必要なのはさきほど追加したPDFShowViewController.swiftに対応するオブジェクトです。

配置する対象はMain.storyboardです。

これをドラッグでstoryboard内の適当な場所に置いてやります。

次はそのオブジェクトとPDFShowViewController.swiftを関連付けます。

classにPDFShowViewControllerを指定します。

次はViewControllerからPDFShowViewControllerへの遷移を定義します。

まずはViewControllerにボタンを配置します。さっきの「View Controller」のオブジェクトを配置したのと同じやり方で、今度は「Button」を置いてやります。仮に「PDF表示」としています。

そして、PDFShowViewControllerの上部の一番左のシンボルを右クリックします。

するとメニューが出てきますので、その中の「show」の右端のマルをドラッグして、さきほど追加したView Controllerのボタンに紐づけます。

スワイプのアクションも同じように、スワイプのオブジェクトを「PDFShowViewController」に追加し、アクションを紐づけます。

これでstoryboard上での各種定義は完了です。

動作確認

それでは早速動かしてみましょう。

PDF表示をタップすると、PDFが表示されます。

Carrier

12:00 PM

61%

Back

SHIKISENSHA Book Guide

繊維製品の事典

定価 8,400 円 (本体) 8,000 円

本書は、繊維の種類、糸の形態、構成、組織、加工、色彩、光沢、模様などによってその性能や外観および風合い特性が著しく変わる繊維製品を第1章から第7章にまとめたものです。特に第4章では数値で表しにくい複雑な構造をもつ多くの繊維製品について、超低倍率の走査電子顕微鏡 (SEM) 下で表面、裏面および断面などを走査し、新しい「ものづくり」の原点につながることを考え作成を試みています。
■執筆 者 廣川 基

染色ノート 改訂 第24版

定価 10,290 円 (本体) 9,800 円

染色ノートは、現用されている「染料・ビグメントレジンカラーおよび繊維加工用薬剤」の実用便覧として、初版以来、繊維・染色加工関係その他多方面でお役に立ち、好評をたまたわっています。今般発行の改訂第24版は、前版以降の業界の推移に対応して、精密な調査と整理を行い、より正確かつ豊富な内容と、より便利な索引を期しております。
日常の業務用に、ぜひ活用ください。

学振版 染色機能加工要論

定価 2,856 円 (本体) 2,720 円

本書は、今後の日本における染色技術の高度化・進展に役立てようと、学振120委員会の手で編集されました。染色機能加工の基本原理が広い範囲でわかりやすく丁寧にまとめられています。学術的にもレベルが高く、また現場の技術者の皆さんにはもちろんのこと、これから染色技術を学ぼうとする学生の皆さんにとっても必見の書です。

綿繊維の構造機能と加工技術

定価 2,856 円 (本体) 2,720 円

綿繊維の一般知識、基本物性、並びにマーセル化の歴史と特徴、最近の樹脂加工技術が網羅されており、繊維の研究者や技術者には是非一読されることをおすすめしたい本です。また、自然に学ぶナノテクノロジーの参考書としても好書です。研究展開のヒントを求めている異業種の方々にも示唆に富む成書でもあります。
■執筆 者 安部 俊三

繊維染色用語辞典 第二版

定価 2,856 円 (本体) 2,720 円

繊維染色用語辞典は染色加工中心の技術・業務関連専門用語を英和対称12,000語、和英索引約12,000語、付録として経常用語など英和関連用語13部門それぞれ約200語づつを選んであります。
■編集 者 安部田貞治 益田 泰
米長 榮 船田登美司

複合素材の染色

定価 5,250 円 (本体) 5,000 円

当社はイギリスの名門 The Society of Dyers and Colourists から染色技術書シリーズの一作として1998年に発行され好評を得た“Blends Dyeing”の翻訳出版権を取得しました。翻訳は学識経験の豊かな安部田貞治氏であります。ぜひ座右において活用してください。
■著 者 John Shore
■訳 者 安部田 貞治

染色加工品の欠点解説 四訂版

定価 7,136 円 (本体) 6,796 円

本書はメーカーの立場からの見解を示したのが大きな特長であります。不上りをなくすることは古くて新しい問題です。本書の利用が染色加工技術の向上に実務的な目標を与え、さらにコスト低減に役立つことでありましょう。
■監 修 福西 興至
■執筆 者 伊藤 信也 池永 静義
立岡 良夫 早川 邦明
三橋 肇夫 吉澤 靖夫

実用織編物の基礎知識

定価 12,747 円 (本体) 12,140 円

織編物の組織を系統的に解説した実務書が少なく待望久しいものがありました。本書はこの要望に応えたもので、実物サンプルも付した有用なデータブックであります。
■別 冊 布地サンプル付
■執筆 者 竹松 茂

株式名染社

〒531-0041 大阪市北区天神橋 7-5-8 HATAビル 4F
TEL 06-6351.6915 FAX 06-6351.6920

ソースコードをGitHubにあげていますので、試してみたい方は以下からどうぞ。

[サンプル](#)

非公開、案件多数

あなたも、フリーランス

詳しくはコチラ ▶

香港

36,000円~

今すぐチェック

*空港送迎、燃費サーチャージ別途

数々の受賞歴を誇るエアラインで

GATHAY PACIFIC

Life Well Travelled

前の記事:
[香港旅行をお勧めする6つの理由](#)

iOS開発という視点でMacbook ProとMacbookを比較してみた

2017/03/27
category:[技術](#)

iPhone (iPod、iPad) の純正充電ケーブル (Lightningケーブル) を無料で交換してみた

2017/04/10
category:[iOS](#), [ライフハック](#)

iPad対応のSwift Playgroundsにプログラミング学習の未来を見た

2017/04/03
category:[iOS](#), [技術](#)

Macでウィンドウが画面外に出て戻せない場合の6つの対処法

2017/03/09
category:[技術](#)



Xcode8で実機インストールができなくなる

2017/03/06
category:[技術](#)

iOS10.3へのアップデートを実際に検証してみた

2017/03/30
category:[iOS](#), [技術](#)

SourceTreeからBitbucketのリポジトリがクローンできない

2017/03/20
category:[技術](#)