

第3回

「あしたの天気」

toru.inoue@kissaki.tv

Agenda

- 1.天気③
- 2.あしたの天気
- 3.壮大なるおまけ

Drivexx/yy

I. 天気③

Drivexx/yy

続。もっとカスタマイズ

FaceWeatherAppDelegate.m に、どんな天気だったらどんな画像、という一覧がある。

時間に余裕があったら、いじってみんべ。の続き。

アプリケーションのフォルダに、

FaceWeatherなんやら > FaceWeather >

のなかに、fair.png とか置いてあると思います。

この画像を適当に他の画像(名前と形式pngとかが一緒！)に変えたと、なんと、、、！！ まあ変わる。

Drivexx/yy

天気によってイメージを変更する
Do it!.

Drivexx/yy

画面の向きを回転させる

<http://www.screenr.com/tbMs>

– (**BOOL**)shouldAutorotateToInterfaceOrientation:
(**UIInterfaceOrientation**)interfaceOrientation

return とある所に、YES。

うん、はい、YES。

shouldAutorotateToInterfaceOrientation

= インターフェースの向きに応じて自動的に回転すべき？ という問いかけ

これにYESと答えを返す(returnする)ことで、自動的に傾きに併せて画面の内容が回転します。

Drivexx/yy

なぜ文字がずれるのか

文字の設定をいじって、ボタンの下にある"まま"にする。

<http://www.screenr.com/bbMs>

文字に対して、"左"、"上"の赤い印を【オン】にしている。

この赤い印が、アンカーとか言う代物。

何でこうなるかというと、

画面が回転する、というときに何が起きているかにしかけがある。

Drivexx/yy

画面が回転する？とはどういう事なのか

手元に紙とかがあったら、回してみよう。
2枚あると良い。

縦にした紙に、文字を書く。
横にした紙を重ねて、下敷きになってる文字をなぞる。

すると、アンカーをつける前と状態が一致しているのが分かるだろうか。

このアンカーは、
画面のとある辺との距離を固定する
ということをしてます。

画面左と上との距離を固定する = 画面のあの辺に「居続ける」という。
ボタンとか他の要素にも設定があるので、いじってみるといいかも。

Drivexx/yy

今日の日付を表示しよう

<http://www.screenr.com/VbMs>

Labelを一つ追加するんだぜ

(える おぼえているか 3しゅうかんまえにも おなじことやった)

で、viewDidLoadって書いてあるかっこの中に、書く。

```
[nowDate setText:[NSString stringWithFormat:@"%今日は%@",  
[NSDate date]]];
```

[NSDate date] これが今日の日付。

Drivexx/yy

文字の表示について説明

[NSString stringWithFormat:@"今日は%@", ひづけ]

で、今日は～となる。こういう仕掛けの物体なんだわコレ。

最後に、一番外側、

[nowDate setText:なにやら];

これで、

さっき作ったLabelに、"今日は～"という文字を表示する。

という具合に、文字が表示できたり

その中身が日付だったりする。

Drivexx/yy

2. あしたの天気

Drivexx/yy

明日の天気

スイッチを一つつけて、明日の天気モードを作ろう。

- ・スイッチを押すと、きょうの天気→明日の天気、へときりかわる
 - ・**その状態でボタンを押すと**、明日の天気がでたりする、、、！！
- という。

Drivexx/yy

明日の天気のために

<http://www.screenr.com/HfMs>

最大、最後の長丁場です。

次のページからちょっとずつ解説があるので、めくりつつ。

Drivexx/yy

スイッチをつけます

デフォルト、offにしときましょう。

押した状態でボタンを押すと、明日の天気、みたいな事にしときます。

Drivexx/yy

ボタンを押したら、のところに、分岐

```
NSString * todayOrTommorrow = @"today";
if ([tomorrowSwitch isOn]) {
    todayOrTommorrow = @"tomorrow";
}
NSDictionary * placeDict = [[NSDictionary
alloc]initWithObjectsAndKeys:
    [NSNumber numberWithInt:46], @"placeNumber",
    todayOrTommorrow, @"todayOrTommorrow",
    nil];
```

もし**tomorrowSwitch**が**on**だったら、
todayOrTommorrow = tomorrow、
そうでなかったら
todayOrTommorrow = today、

となるコードです。

Drivexx/yy

ラスト、Webサービスに贈る言葉

FaceWeatherAppDelegate.m

を開いて、

– (void) tapped:(NSNotification *)notif {

を弄ります。

ま、コピペでOKです。

todayだったりtomorrowだったり、switchによって

内容が変わるのが、ここまで届いて、天気をWebから持ってくるという構造になっています。

で

全員分のボタンにこのコードを書かないといけないのだぜ！！

Drivexx/yy

とりあえず、以上！！！！！！

Drivexx/yy

ここらで一度GitHub Appで保存→DLしてみようぜ！！(まさかのコピペ)

GitHub Appを起動して、
commit (コメント書く作業) →右上のSyncを押して、押したら

toru.inoue@kissaki.tv まで、Pushしましたよメールくれ。

わからなければ過去資料なのだけ
<http://www.screenr.com/hv1s>

Push後、GitHubのサイトのアドレスとか書いてくれると、
メールを受け取る→リンクで直接サイトまで行ける、とかできてうれしい。

Drivexx/yy

今までやってきた事は何だったんだぜ

Objective-Cという、れっきとした**C言語**を使って、
動かしたり回したり天気表示したり今日と明日を切り替えたり
していました。

この資料は、**やりたい事とやってる事が一対一に対応している**ので、
同じような事がしたければやってみると良いのだ。

Drivexx/yy

もっと高度な事がしたい！ ならば

一つ、プログラミング言語を勉強する事をお勧めする。

→正直言って、今までプログラミング言語を勉強した事が無い人が、いきなりObjective-Cを学ぶ事は、

草野球やってる小学生がいきなりメジャーに登板する

レベルに等しい。

それは不可能ではないけれど、
しないほうがいい努力な可能性が高い。
例えばググりまくって同じような学び方した人から学ぶ事は出来る。
でも。
大抵、偏った学び方になる。最悪死ぬ。

Drivexx/yy

もっと高度な事がしたい！ ならば
何語かやってますよ、って人へ

特にLL言語しかやってない場合など、
C言語が動く仕組みを理解していないと、
壮絶に勘違いしたままググってコピペでコードを書くという

俺が殺したい人ナンバーワン圏内に入ってしまう

ので、気をつけよう。
コピペ→できたw うはw おkww→ブログ書く→コピペ、、

ミスっていてもC言語的な前提があっても気にしないクソループの完成。
それだけは赦さない。殺す。

プログラミングの基礎はC言語が一番良いが、
いかんせん全く楽しく無いので、JavaやRubyからCをやるのが良いと思う。

Drivexx/yy

もっと高度な事がしたい！ ならば

おすすめの言語とか

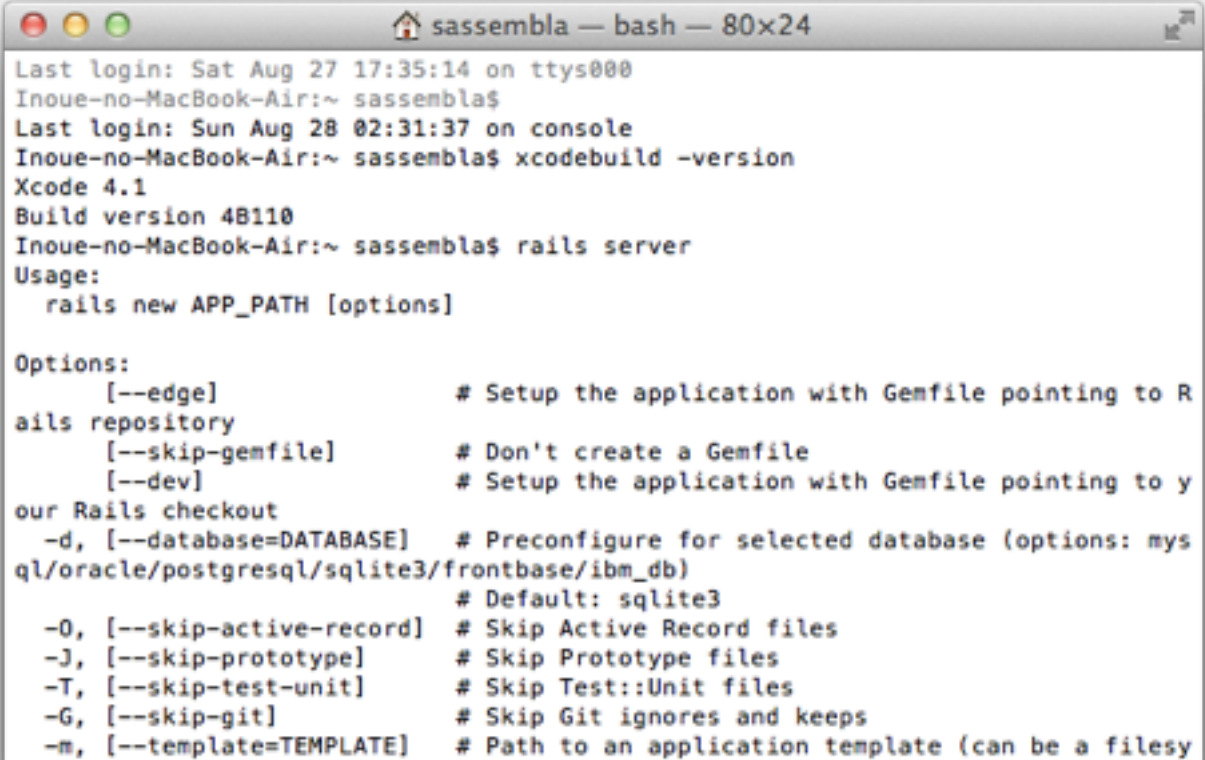
→Java、Ruby、Python。読みやすく書きやすくやってる人が多い言語。
そしてC言語。これは上記が終わってから第二言語としてやるべき。

おすすめの使用シーン、勉強箇所とか

→画面がある環境。

たとえばなにかするたびに「画面に変化が表示される」と、
それはとっても盛り上がる。

こんな画面を
ずーっと見続けて、
はじめっから楽しい
わけが無い。

A terminal window titled 'sassembra — bash — 80x24' showing the output of 'rails server' and the usage of 'rails new'. The output includes the last login time, the user's name, and the version of Xcode and Rails. The usage of 'rails new' is shown with various options and their descriptions.

```
Last login: Sat Aug 27 17:35:14 on ttys000
Inoue-no-MacBook-Air:~ sassembra$
Last login: Sun Aug 28 02:31:37 on console
Inoue-no-MacBook-Air:~ sassembra$ xcodebuild -version
Xcode 4.1
Build version 4B110
Inoue-no-MacBook-Air:~ sassembra$ rails server
Usage:
  rails new APP_PATH [options]

Options:
  [--edge]                # Setup the application with Gemfile pointing to R
ails repository
  [--skip-gemfile]        # Don't create a Gemfile
  [--dev]                 # Setup the application with Gemfile pointing to y
our Rails checkout
  -d, [--database=DATABASE] # Preconfigure for selected database (options: mys
ql/oracle/postgresql/sqlite3/frontbase/ibm_db)
                          # Default: sqlite3
  -O, [--skip-active-record] # Skip Active Record files
  -J, [--skip-prototype]    # Skip Prototype files
  -T, [--skip-test-unit]    # Skip Test::Unit files
  -G, [--skip-git]          # Skip Git ignores and keeps
  -m, [--template=TEMPLATE] # Path to an application template (can be a filesy
```

Drivexx/yy

できる事を増やすには、やりたい事を増やしていく事が必要

ここで、基礎が必要になる。

真似をしながら正しい基礎を学ぶには

→ドキュメントを読む！！

- ・ Appleが作った奴です。ほぼ全くミス無いです。

→言語を勉強する！！

- ・ C言語、Java、Ruby、Pythonなどを勉強するといいいでしょう。
- ・ 逆引き辞典とかおすすめです。

<http://www.amazon.co.jp/最新Java逆引き大全-550の極意-高橋-和也/dp/479800815X>

→徒党を組む！！

- ・ 複数人で勉強した方がはかどります。

どのくらいかというと、

1人→1倍、2人→6~8倍、3人→~12倍 4人→~~モンハン~~

理由は、同じ事を別の人がやるから。

無いようにも因りますが、3人くらいがちょうどいいでしょう。

Drivexx/yy

3. 壮大なるおまけ

Drivexx/yy

壮なおまけ説明

今回の講義の裏側では、複数のシステムが動いています。

GitHub、Git、GitHubApp、TestFlight。

GitHubについての説明

GitHub

みなさんアカウント作りましたが、これは、Gitというアプリケーションを使ったWebサービスです。



Git

[http://en.wikipedia.org/wiki/Git_\(software\)](http://en.wikipedia.org/wiki/Git_(software))

[distributed revision control system](http://en.wikipedia.org/wiki/Git_(software)) (分散バージョン管理システム)

はい、なんのこっちゃ判りませんね。

Drivexx/yy

Gitとは

作り途中のアプリ、テキスト、データなどを、
その状態で保存するアプリケーション(プログラム)です。

プログラムを好きなタイミングで、Gitリポジトリ(データの保存場所)へと
放り込んだり、また古いものを取り出したりできます。

で、リポジトリをWeb上に置いてくれているサービスが、GitHubです。

かつ、Gitには、手元セーブ(commit)と、Gitリポジトリへのセーブ(push)
があります。

まず手元でcommit(+コメント書く)→pushとすることで、
Web上のGitHubにデータを送り込んでいた訳です。



Drivexx/yy

GitHubApp

で、実際にその作業をしていたのが、GitHubApp。
みなさんとはアイコンが違うかもしれません。

な お し ま し た 。

だってかわいく無(ry

このアプリを使って、GitHubへとアップとかしてました。

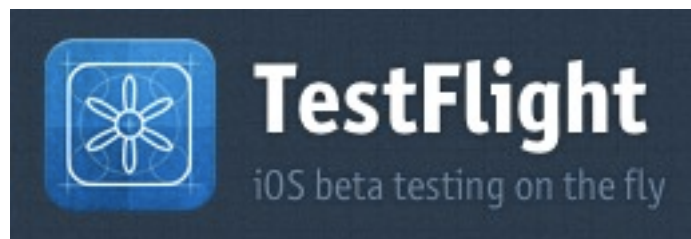


Drivexx/yy

GitHubとTestFlight

で、GitHubからは、皆さんのアプリケーションが自由にDLできます。
それを、皆さんの端末まで届けていたのが、コレ。

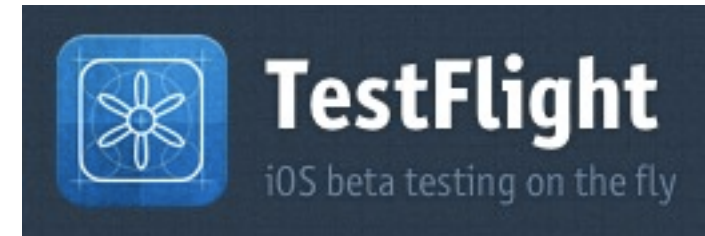
TestFlight というWebサービス。



Drivexx/yy

TestFlight

アプリをここにアップすると、
皆さんのiPhoneとかにメールが届きます。



そのメールを操作してると、アプリがDLできる、という仕組み。

以上、解説まで。

LookBack

- 1.天気③
- 2.あしたの天気
- 3.壮大なるおまけ

LookBack

質問、ヘルプ

メールにて受け付けます。できうるだけフォーマットというか、宛先、自分の名前、とか書いてね。

で、プログラムのな問題であれば(エラーが出る!とか) GitHubへとPushしてもらって、ソレを見てくれ、ってメールしてくれると、凄く楽です。

また、

- ・~をしたらかしくなったでござる
- ・~時以降におかしくなったでござる

など、

原因かどうかわんなくても、「あーこんな事そういえばしたわー」という

情報があると、解決が速かったりします。

See you! someday, somewhere!!