

iOS Apps

An Introduction



Tools



“optional”

+



(a sexy, yet not-so-bug-free beast)

Objective-C

- C with a “hint” of SmallTalk.
- Made popular by NeXT and Apple
- Compiled with Clang or GCC
- Thus... friendly with plain C and C++

You'll learn to love it.

What's Different?

C++

- 1) `Type *object = new Type;`
- 2) `object->function(argument);`
- 3) `delete object;`

Objective-C

- 1) `... = [[Type alloc] init];`
- 2) `[object function:argument];`
- 3) `[object release];`

Messaging (Calling Methods)

```
[receiver method];
```

```
[receiver method:argument];
```

```
[receiver method:arg1 withMore:arg2];
```

Method names get **long**.

This increases *readability* and decreases
the need for *documentation*.

Things are more or less written in the same order as C++

Camel Case

```
[receivingObject switchBooleanTo:YES  
withTiming:1.0f  
andOhGodIsThisGettingTooLong:YES  
maybeIWillPassInAnotherObject:otherObject  
andJustToFinishHereIsTheLastObject:last  
Object];
```

Ewww....

Style > Short Names

Line up your colons.



```
[receivingObject switchBooleanTo:YES  
                        withTiming:1.0f  
andOhGodIsThisGettingTooLong:YES  
maybeIWillPassInAnotherObject:otherObject  
andHereIsTheLastObject:lastObject];
```

A little better...

#xcodedoesthisforyou

MEH-MUHR-REE!

- Reference Counting \neq Garbage Collection
- Every object has the member: *retainCount*
- And four important methods:

```
graph LR; A["[object alloc]  
(new)"] --> B["[object dealloc]  
(delete)"]
```

[object retain] [object release]
(uhh... what are these?)

But No Worries!



(that's Automatic Reference Counting)

Structure

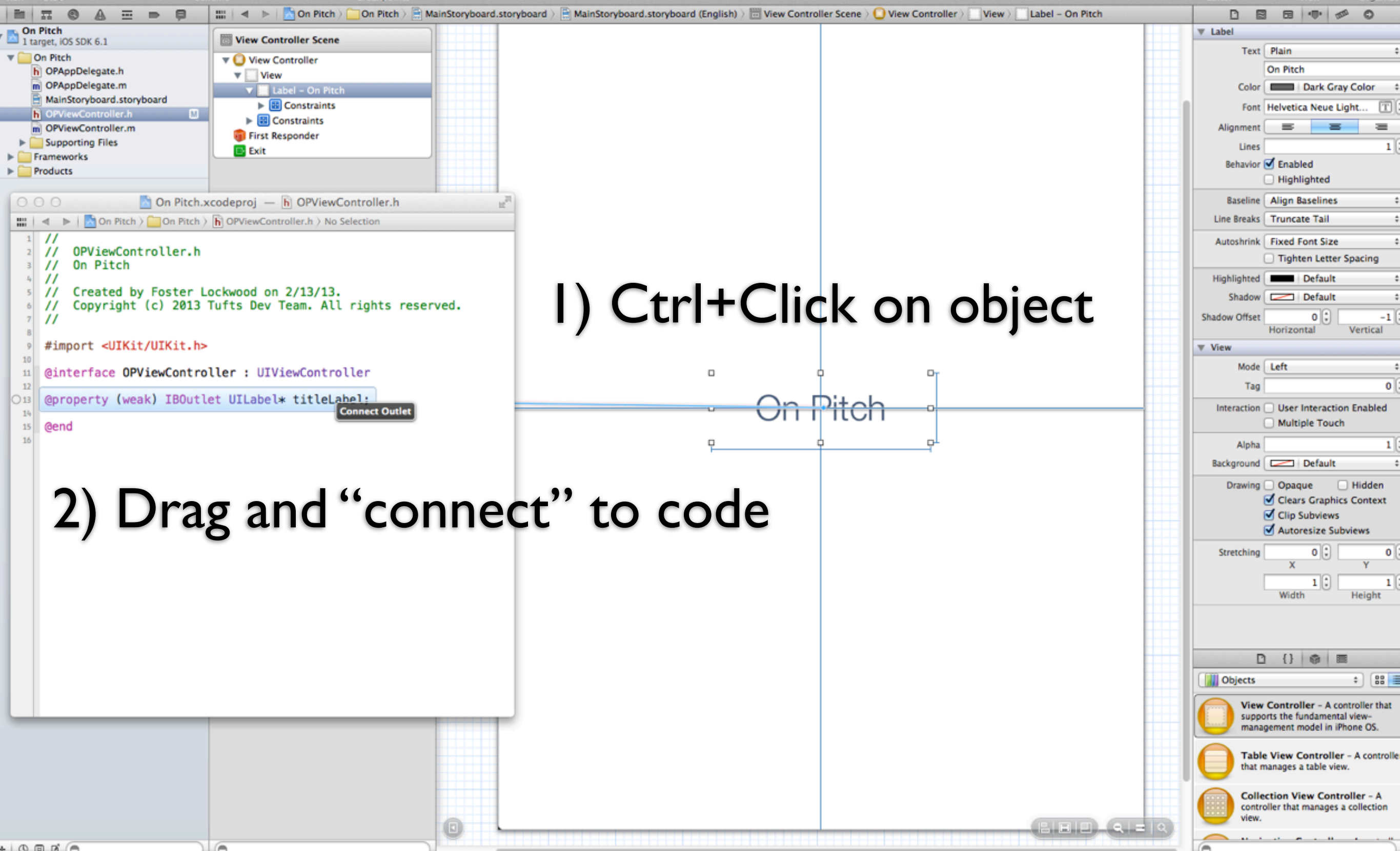
@interface – All public declarations

@implementation – All definitions

@property – Use for member variables

Use the type qualifier “IBOutlet” for connections in the Interface Builder

Also, @end denotes end of @interface or @implementation.



There are a few different ways to accomplish this, each are equivalent and enable the object to be reference in code.

Interface (Class.h)

```
@interface Class : SuperClass

@property (strong) NSString *objectName;
@property (strong) NSArray *importantArray;
@property (assign) NSInteger someCount;

+ (NSInteger)universalNumberForClass;

- (id)initWithName:(NSString*)name;
- (void)performOperationWithObject:(id)object;
- (BOOL)doesArrayContainObject:(id)object;

@end
```

Implementation (Class.m)

```
@implementation Class

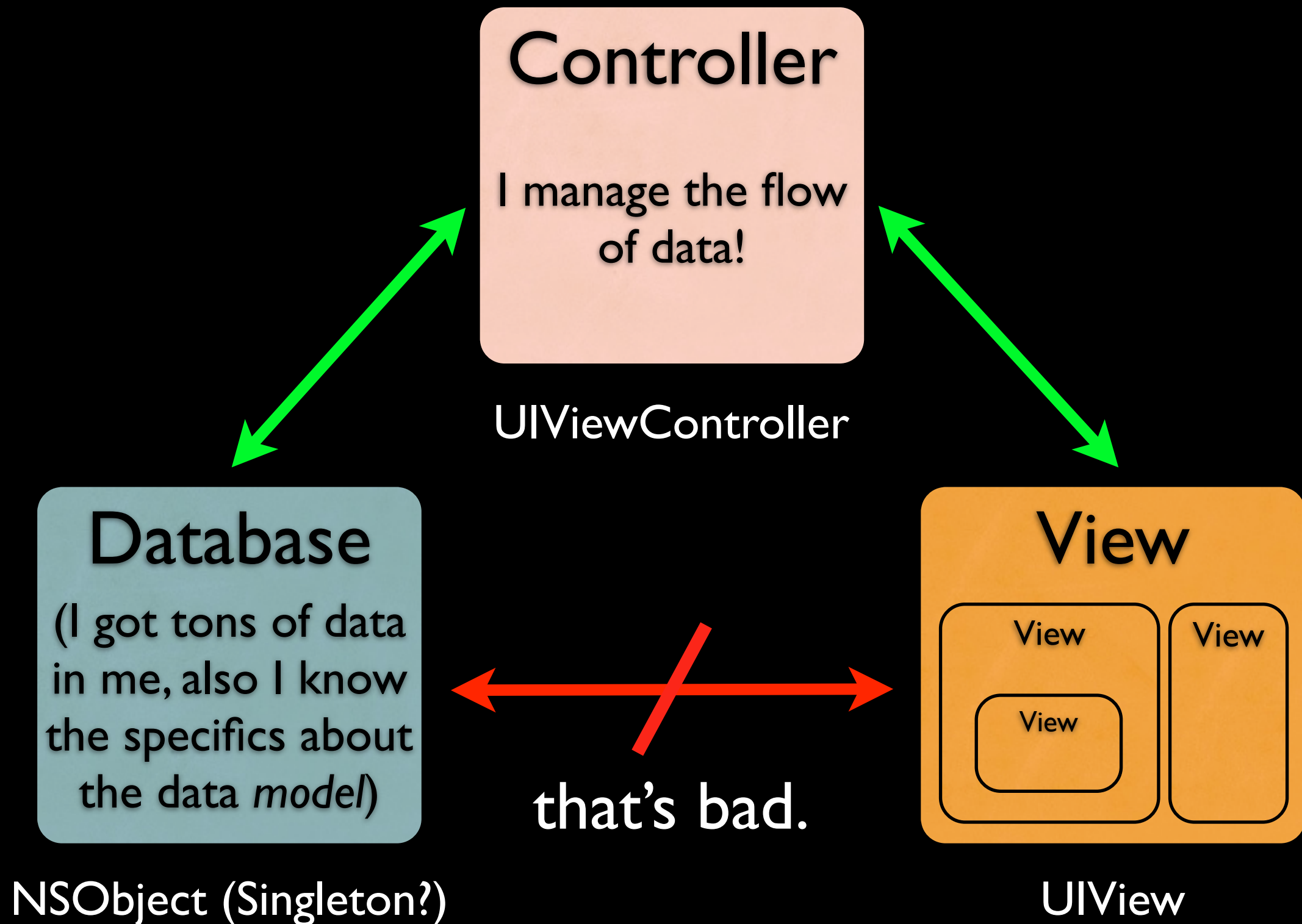
+ (NSInteger)universalNumberForClass {
    return 42;
}

- (id)initWithName:(NSString*)name {
    self = [super init];
    if (self) {
        self.objectName = name;
        self.importantArray = nil;
        self.someCount = 0;
    }
    return self;
}

// - (void)performOperationWithObject:(id)object;
// - (BOOL)doesArrayContainObject:(id)object;

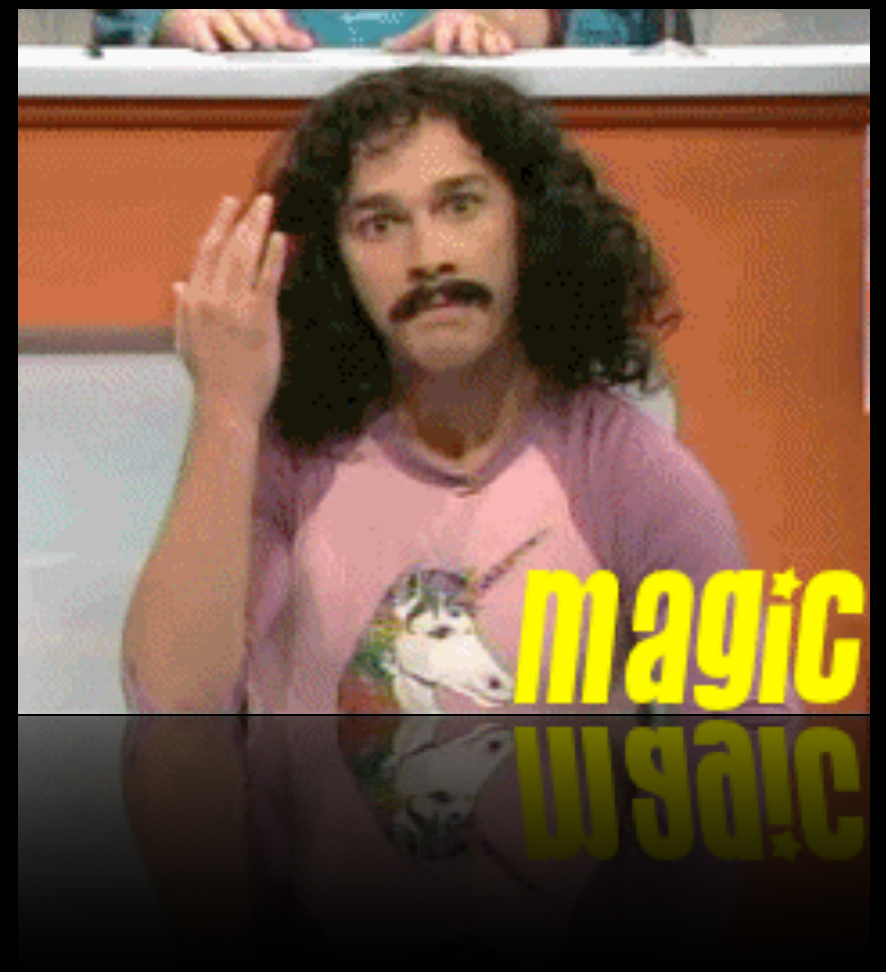
@end
```

Model - View - Controller




The Magic

- Apple's Frameworks
- Device Capabilities
- Terrific third party code



Frameworks

(aka the coolest things since ever.)

- All the Basics - “Foundation” 
- Location Data - “CoreLocation”
- Audio/Video Library - “AVFoundation”
- Accelerometer - “CoreMotion”
- Network Operations - “CFNetwork”
- AND SO MUCH MOOOOOORE!

- NSObject
- NSString
- NSArray
- NSDictionary
- NSMutableArray
- NSMutableDictionary
- And more...

Eek...! 

Plug Into Those APIs!

MKNetworkKit

<https://github.com/MugunthKumar/MKNetworkKit>

Currently the easiest and best way to use RESTful APIs,
which should be a'plenty in the upcoming hack-a-thon!

MKNetworkKit Example

```
// ... in some method of some class
self.networkEngine = [[MKNetworkEngine alloc
                      initWithHostName:@"api.google.com"];

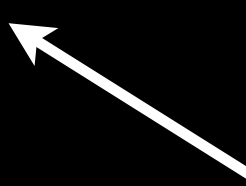
MKNetworkOperation* op = [self.networkEngine
                          operationWithPath:@"some/api/path"
                          params:nil
                          httpMethod:@"GET"];

[op onComplete:^(MKNetworkOperation *completedOperation) {
    NSLog(@"%@", [completedOperation responseJSON]);
}
   onError:^(NSError* error) {
    NSLog(@"%@", [error localizedDescription]);
}];

[self.networkEngine enqueueOperation:op];

// The completion block will be asynchronously
// called if a response was received, else error!
```

Logistics

- iOS Developer Program (\$99/year)  Not cheap.
- Enables physical device testing
- 100 allowed ad-hoc devices
- Unlimited submission to Store
- What you get for free (non-members)
 - Xcode, iOS Simulator, full SDK
 - GitHub third party source

Q&A

email - foslock@gmail.com

reference - [Apple's Documentation](#)

the best website ever - [StackOverflow](#)

Maybe an example or walkthrough?