

RubyMotion





@jonathanpenn



@jonathanpenn



Navel Labs



@jonathanpenn



Navel Labs



COCOA MANIFEST

Goodies

<http://cocoamanifest.net/features>

Who are you?

Let's build an app...



Desktop •



memories • |

```
1 # -*- coding: utf-8 -*-
2 $.unshift("/Library/RubyMotion/lib")
3 require 'motion/project'
4
5 Motion::Project::App.setup do |app|
6   # Use `rake config` to see complete project settings.
7   app.name = 'memories'
8 end
```



memories • █

```
1 class AppDelegate
2   def application(application, didFinishLaunchingWithOptions:launchOptions)
3     true
4   end
5 end
```



memories • █

```
1 class AppDelegate
2   def application(application, didFinishLaunchingWithOptions:launchOptions)
3     true
4   end
5 end
```

```
1 class AppDelegate
2   def application(application, didFinishLaunchingWithOptions:launchOptions)
3
4     screenBounds = UIScreen.mainScreen.bounds
5     @window = UIWindow.alloc.initWithFrame(screenBounds)
6
7     @window.makeKeyAndVisible
8     true
9   end
10 end
```



memories • █

```
memories • rake
```

```
rake
```

```
(in /Users/jonathan/Desktop/memories)
```

```
  Build ./build/iPhoneSimulator-6.0-Developme
```

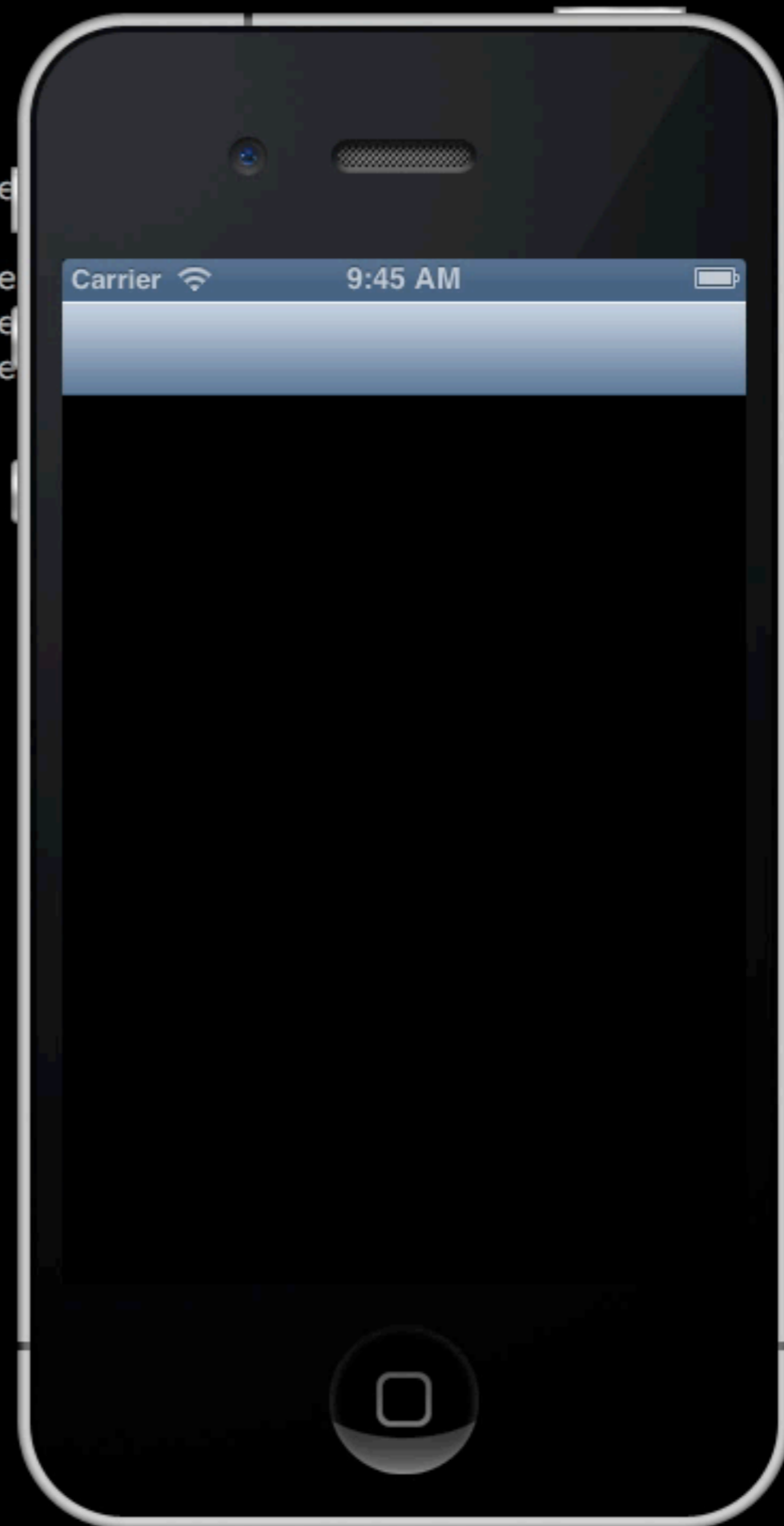
```
  Compile ./app/app_delegate.rb
```

```
  Link ./build/iPhoneSimulator-6.0-Developme
```

```
  Create ./build/iPhoneSimulator-6.0-Developme
```

```
  Simulate ./build/iPhoneSimulator-6.0-Developme
```

```
(main)> █
```



```
1 class AppDelegate
2   def application(application, didFinishLaunchingWithOptions:launchOptions)
3
4     screenBounds = UIScreen.mainScreen.bounds
5     @window = UIWindow.alloc.initWithFrame(screenBounds)
6
7     navigationController = UINavigationController.alloc.init
8     @window.rootViewController = navigationController
9
10    @window.makeKeyAndVisible
11    true
12  end
13 end
```

```
1 class AppDelegate
2   def application(application, didFinishLaunchingWithOptions:launchOptions)
3
4     screenBounds = UIScreen.mainScreen.bounds
5     @window = UIWindow.alloc.initWithFrame(screenBounds)
6
7     navigationController = UINavigationController.alloc.init
8     @window.rootViewController = navigationController
9
10    memoriesController = MemoriesTableViewController.alloc.init
11    navigationController.setViewControllers([memoriesController])
12
13    @window.makeKeyAndVisible
14    true
15  end
16 end
```



memories • |

```
memories • rake
```

```
rake
```

```
(in /Users/jonathan/Desktop/memories)
```

```
  Build ./build/iPhoneSimulator-6.0-Developme
```

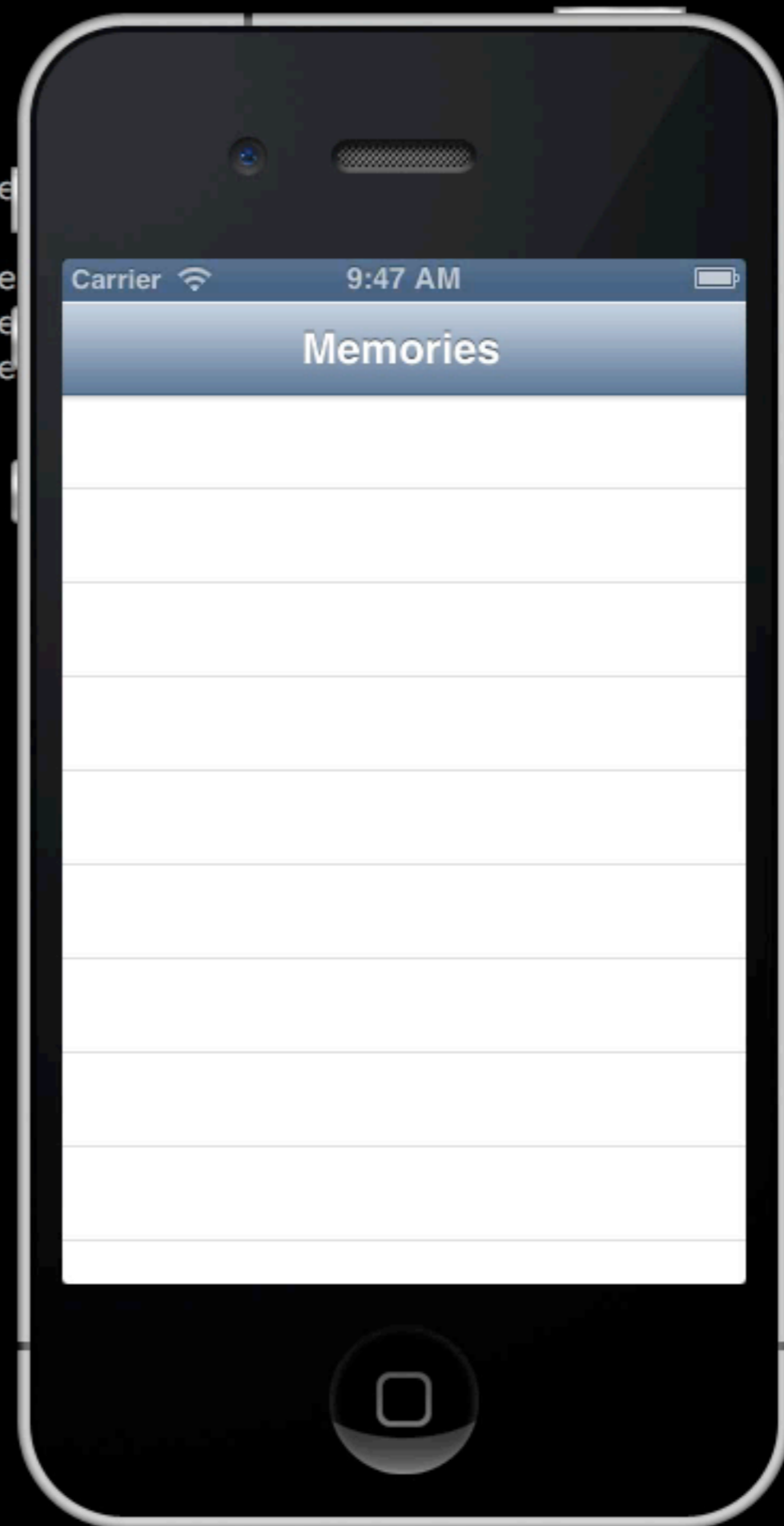
```
  Compile ./app/app_delegate.rb
```

```
  Link ./build/iPhoneSimulator-6.0-Developme
```

```
  Create ./build/iPhoneSimulator-6.0-Developme
```

```
  Simulate ./build/iPhoneSimulator-6.0-Developme
```

```
(main)> █
```



```
1 class AppDelegate
2   def application(application, didFinishLaunchingWithOptions:launchOptions)
3
4     screenBounds = UIScreen.mainScreen.bounds
5     @window = UIWindow.alloc.initWithFrame(screenBounds)
6
7     navigationController = UINavigationController.alloc.init
8     @window.rootViewController = navigationController
9
10    memoriesController = MemoriesTableViewController.alloc.init
11    navigationController.setViewControllers([memoriesController])
12
13    @window.makeKeyAndVisible
14    true
15  end
16 end
17
18 class MemoriesTableViewController < UITableViewController
19
20   def viewDidLoad
21     super
22     self.title = "Memories"
23   end
24
25 end
```



memories • █

RubyMotion

RubyMotion

what?

A subset of Ruby



Yukihiro Matsumoto

Old

Papa Smalltalk

Papa Smalltalk
+ Mother Perl

Papa Smalltalk
+ Mother Perl

son of perdition?

What if?

Objective C

Objective C
+ Ruby Syntax

Objective C
+ Ruby Syntax
+ LLVM

Objective C
+ Ruby Syntax
+ LLVM
+ ARC

Objective C
+ Ruby Syntax
+ LLVM
+ ARC

RubyMotion

Objective C Preprocessor

What does Ruby give us?

```
NSFileManager *mgr = [NSFileManager defaultManager];
NSArray *paths = [mgr contentsOfDirectoryAtPath:@" /etc"
                error:nil];

[paths enumerateObjectsUsingBlock:^(id obj, ...) {
    NSLog(@"%@", obj);
}];
```

```
Dir["/etc"].each do |name|  
  puts name  
end
```

“Syntactic Sugar”

```
BW::Location.get do |result|  
  lat = result[:to].latitude  
  lon = result[:to].longitude  
  puts "You are at #{lat}, #{lon}"  
end
```

OMG!

History Lesson

```
id age = objc_msgSend(objc_getClass("NSNumber"),
                      sel_getUid("numberWithInt"), 20);

id params = objc_msgSend(objc_getClass("NSMutableDictionary"),
                          sel_getUid("dictionaryWithObjectsAndKeys"),
                          @"Sam", @"name",
                          age, @"age",
                          nil);

objc_msgSend(params, sel_getUid("setObject:forKey:"),
             "Hello", "greeting");
```

```
NSMutableDictionary *params;  
params = [NSMutableDictionary dictionaryWithObjectsAndKeys:  
    @"Sam", @"name",  
    [NSNumber numberWithInt:20], @"age",  
    nil];  
  
[params setObject:@"Hello" forKey:@"greeting"];
```

```
NSDictionary *immutableParams = @{@"name": @"Sam",  
                                   @"age": @(20)};
```

```
NSMutableDictionary *params = [immutableParams mutableCopy];  
params[@"greeting"] = @"Hello";
```

Syntax Matters

Mixins

Mixins

```
module CapitalNames
  def capitalized_name
    name.upcase
  end
end
```

Mixins

```
class User
  include CapitalNames

  def initialize name
    @name = name
  end

  def name
    @name
  end
end
```

```
user = User.new("Jonathan")
puts user.capitalized_name
```

Mixins

```
user.extend(CapitalizeNames)
```

Objective C Does Mixins!

(kinda)

```
objc_getMetaClass(...)
```

```
class_addMethod(...)
```

Open Classes

Open Classes

```
class Fixnum
  def days
    self * 24 * 60 * 60
  end

  def ago
    Time.now - self
  end
end
```

Open Classes

3 . days . ago

Open Classes

```
@implementation NSNumber (TimeDiff)

- (NSNumber *)days {
    // ...
}

- (NSDate *)ago {
    // ...
}

@end
```

Open Classes

```
NSDate *date = [[@(3) days] ago];
```

Open Classes

```
NSDate *date = @(3).days.ago;
```

Caveat Developer

Namespaces!!

```
[[JPPlayerModel alloc] init];
```

```
UIViewAutoresizingFlexibleBottomMargin;
```

```
NSUbiquitousKeyValueStoreDidChangeExternallyNoti
```

```
module Models
```

```
    VERSION = "1.0"
```

```
    class Player
```

```
        # ...
```

```
    end
```

```
end
```

```
Models::Player.new("First")
```

```
# Or...
```

```
include Models
```

```
Player.new("Second")
```

```
puts VERSION
```

Bottom line...

Learn Ruby

WITH THE NEO RUBY KOANS

The Koans walk you along the path to enlightenment in order to learn Ruby. The goal is to learn the Ruby language, its structure, and some common functions and libraries. We also teach you culture. Testing is not just something we pay lip service to, it is something we live. It is essential in your quest to learn and do things in the language.

Ruby Koans
rubykoans.com

So...what's the catch?

So...what the #\$\$%^@?!

No “require”

No “require”

No gems

No “require”

No gems...as we know them

No “require”

No gems...as we know them

No “binding” on blocks

No “require”

No gems...as we know them

No “binding” on blocks

No static typing

No “require”

No gems...as we know them

No “binding” on blocks

No static typing

No weak references

No “require”

No gems...as we know them

No “binding” on blocks

No static typing

No weak references

Mmmmm...Layers!

No “require”

No gems...as we know them

No “binding” on blocks

No static typing

No weak references

Mmmmm...Layers!

Debugging...is complicated

```
class AppDelegate
  def applicationDidFinishLaunching application
    DoSomethingAsync.new.upload "some text" do
      puts "called back"
    end
  end
end
```

Spot the bug...

```
class DoSomethingAsync
  def upload data, &block
    @callback = block
    pretend_to_upload_data
  end

  def pretend_to_upload_data
    Dispatch::Queue.main.after 1 do
      @callback.call
    end
  end
end
```

```
class AppDelegate
  def applicationDidFinishLaunching application
    @uploader = DoSomethingAsync.new
    @uploader.upload "some text" do
      puts "called back"
      @uploader = nil
    end
  end
end
```

Spot the bug...

```
class DoSomethingAsync
  def upload data, &block
    @callback = block
    pretend_to_upload_data
  end

  def pretend_to_upload_data
    Dispatch::Queue.main.after 1 do
      @callback.call
    end
  end
end
```

Sending Messages

Normal Ruby

```
def something arg1, options={}  
  puts arg1, options[:andOther]  
end
```

```
something "a", andOther: "b"
```

```
send "something", "a", {andOther: "b"}
```

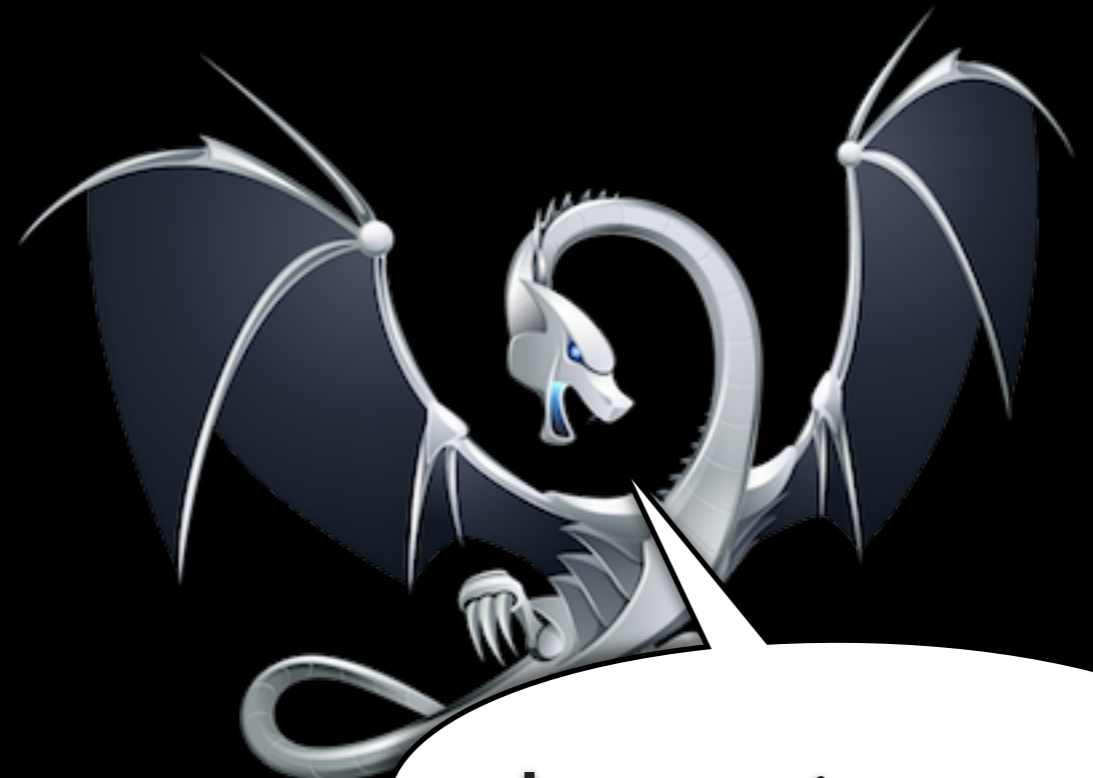
RubyMotion

```
def something arg1, andOther: arg2  
  puts arg1, arg2  
end
```

```
something "a", andOther: "b"
```

```
send "something:andOther:", "a", "b"
```





I come in peace!

Tell 'em
what they've won!

Test support

Test support
Live REPL

Test support

Live REPL

Command Line Workflow

Test support

Live REPL

Command Line Workflow

RubyMine IDE

Test support

Live REPL

Command Line Workflow

RubyMine IDE

Motion gems

Test support

Live REPL

Command Line Workflow

RubyMine IDE

Motion gems

CocoaPods!

Test support

Live REPL

Command Line Workflow

RubyMine IDE

Motion gems

CocoaPods!

Drop in Objective C

Test support

Live REPL

Command Line Workflow

RubyMine IDE

Motion gems

CocoaPods!

Drop in Objective C

Build as a static library

Test support

Live REPL

Command Line Workflow

RubyMine IDE

Motion gems

CocoaPods!

Drop in Objective C

Build as a static library

Hipbyte

What do I use it for?

Quick Experiments

Quick Experiments

Domain Model Prototypes

Quick Experiments
Domain Model Prototypes
Static Libraries

Quick Experiments

Domain Model Prototypes

Static Libraries

Study Cocoa Touch

DEMO!



Resources

Slides and other info

<http://cocoamanifest.net/features>

Awesome Tutorial on RubyMotion and iOS

<http://rubymotion-tutorial.com>

Samples

<https://github.com/HipByte/RubyMotionSamples>

Low Level Presentation on MacRuby

<http://bostonrb.org/presentations/month/June-2011>

RubyMotion





Jonathan Penn

@jonathanpenn

jonathan@cocoamanifest.net

for more info...

<http://cocoamanifest.net/features>