

Ruby Cocoa

Native apps with a scripting language

Cocoa

- Apple's application Framework
- Object Oriented
- Objective-C

Ruby Cocoa

- Bridge between Objective-C and Ruby
- Ruby can talk to Objective-C
- Objective-C can talk to ruby
- Better fit than the Java runtime

Why Ruby Cocoa?

- ❖ Objective-C is ugly
- ❖ Write Cocoa applications with Ruby
- ❖ Use cocoa framework within a script
- ❖ Use cocoa framework within a rails application

What About Mac Ruby?

- ❖ Shiny
- ❖ The Future
- ❖ Ruby 1.9
- ❖ Zero toll bridge
- ❖ Not yet shipping in OS X

The Learning Curve

- Learn Ruby
- Learn Cocoa
- Learn more Cocoa

Where to Start?

- ▣ <http://www.rubycocoa.com/>
- ▣ [http://developer.apple.com/leopard/overview/
scriptingcocoa.html](http://developer.apple.com/leopard/overview_scriptingcocoa.html)
- ▣ <http://developer.apple.com/cocoa/>

Understanding The Docs

Understanding The Docs

- ⌘ Apple's documentation is in Objective-C
- ⌘ Get used to it

Language Differences

- ✖ Ruby is Memory Managed
- ✖ Objective-C methods can have inline arguments

Language Differences

Method Signature:

```
- (BOOL)setValue:(id)value forProperty:(NSString *)property
```

Objective-C:

```
[record setValue: value forProperty: property];
```

Ruby:

```
record.setValue_forProperty(value, property)
```

Ruby Cocoa and Leopard

Ruby Cocoa and Leopard

```
>> require 'osx/cocoa'  
=> true  
>> OSX::RUBYCOCOA_VERSION  
=> "0.13.1"
```

Ruby Cocoa and Leopard

```
#!/usr/bin/env ruby
```

Ruby Cocoa and Leopard

```
#!/usr/bin/env ruby  
  
require 'osx/cocoa'
```

Ruby Cocoa and Leopard

```
#!/usr/bin/env ruby

require 'osx/cocoa'

OSX::NSApplication.sharedApplication

OSX::NSApp.run
```

Everything else is an
implementation detail.

Ruby Cocoa and Leopard

```
#!/usr/bin/env ruby

require 'osx/cocoa'

OSX::NSApplication.sharedApplication

OSX::NSApp.run
```

Ruby Cocoa and Leopard

```
#!/usr/bin/env ruby

require 'osx/cocoa'
include OSX

OSX::NSApplication.sharedApplication

OSX::NSApp.run
```

Ruby Cocoa and Leopard

```
#!/usr/bin/env ruby

require 'osx/cocoa'
include OSX

NSApplication.sharedApplication

NSApp.run
```

Ruby Cocoa and Leopard

```
#!/usr/bin/env ruby

require 'osx/cocoa'
include OSX

class AppDelegate < NSObject

end

NSApplication.sharedApplication

NSApp.run
```

Ruby Cocoa and Leopard

```
#!/usr/bin/env ruby

require 'osx/cocoa'
include OSX

class AppDelegate < NSObject

    def applicationDidFinishLaunching(aNotification)

    end

end

NSApplication.sharedApplication

NSApp.run
```

Ruby Cocoa and Leopard

```
#!/usr/bin/env ruby

require 'osx/cocoa'
include OSX

class AppDelegate < NSObject

    def applicationDidFinishLaunching(aNotification)
        puts "#{aNotification.name} makes me say: Hello,
World!"
    end

end

NSApplication.sharedApplication

NSApp.run
```

Ruby Cocoa and Leopard

```
#!/usr/bin/env ruby

require 'osx/cocoa'
include OSX

class AppDelegate < NSObject

    def applicationDidFinishLaunching(aNotification)
        puts "#{aNotification.name} makes me say: Hello,
World!"
    end

end

NSApplication.sharedApplication
NSApp.setDelegate(AppDelegate.alloc.init)
NSApp.run
```

The Result?

NSApplicationDidFinishLaunchingNotification makes me say: Hello, world

Lets Build Something Else

- ❖ Address Book rails application.
- ❖ Some simple rubycocoa applications.