Ruby on Rails
web application dev

about me

thomas mak
thomas@cpttm.org.mo

agenda

Ruby Basic

Installation

Creating Rails Proj

Ruby Basic

Ruby Basic

Variable

Method

Class

```
foo = “bar”
```

Ruby Basic

Variable

Method

Class

```ruby
def method_name
  do_something
end
```

Ruby Basic

Variable

Method

Class

```
def hello(name)
  puts “hello #{name}”
end
```

Ruby Basic

Variable

Method

Class

```
“hello #{name}”
```

we can use **#{ }** to put variable in **double quoted** string.

Ruby Basic

Variable

Method

Class

```
hello “Thomas”
hello(“Thomas”)
```

when we call the method with the parameters,
we can either use the () or not.

Ruby Basic

Variable

Method

Class

```ruby
# Class Def
class Foo
  def hello
    puts "Hello Ruby"
  end
end
```

Ruby Basic

Variable

Method

Class

```
# Class Def
```

for comments, we use # symbol.
the comment valid till the line ending.

Ruby Basic

Variable

Method

Class

```ruby
class Foo

  def hello
    puts "Hello Ruby"
  end

  private
  def internal_method
    do_something
  end
end
```

all methods below the **private** keyword are private.

Ruby Basic

Variable

Method

Class

```ruby
def self.anther_method
  something_not_related_to_instances
end
```

definite the method with **self.** makes it a **class method**.

Ruby Basic

Variable

Method

Class

```
class Foo
  def initialize
    init_things_here
  end
end
```

an initialize method is a constructor method.

Ruby Basic

Variable

Method

Class

```ruby
class Foo
end

obj = Foo.new
```

creating instance from class defition.

Ruby Basic

Variable

Method

Class

```ruby
class Foo
  def initialize
    @bar = 123
  end
end
```

@variable with @ sign is instance variable.

Ruby Basic

Variable

Method

Class

```ruby
class Foo
  attr_reader :bar
  def initialize
    @bar = 123
  end
end

obj = Foo.new
obj.bar # 123
obj.bar = 456
# NoMethodError: undefined method 'bar='
```

instance variable with **read-only** public access.

Ruby Basic

Variable

Method

Class

```
class Foo
  attr_accessor :bar
  def initialize
    @bar = 123
  end
end
```

instance variable with **read-write** public access.

Ruby Basic

Variable

Method

Class

```ruby
class Foo
  attr_accessor :bar
  def initialize
    @bar = 123
  end

  def b=(value)
    @bar = value
    puts "Setting bar value to #{value}"
  end
end
```

and we can define custom setter method.

Ruby Basic

Variable

Method

Class

```
obj = Foo.new
obj.bar # 123
obj.b = 321 # Setting bar value to 321
obj.bar # 321
obj.b
# NoMethodError: undefined method 'b'
```

testing the Foo class with custom setter method.

try ruby

Using the Prompt

Good! You did a bit of math. See how the answer popped out?

Ruby recognizes numbers and mathematic symbols. You could try some other math like:

- `4 * 10`
- `5 - 12`
- `40 / 4`

Type `next` to move to the next lesson when you're finished.

```
Interactive ruby ready.
> next
> next
>
  => nil
  Success!
> 2+6
  => 8
  Success!
>
```

Enter / Return → Submit code
Up → Cycle through submitted code
`clear` → Clear the editor window
`back` → Return to the previous lesson
`next` → Move to the next lesson

http://tryruby.org

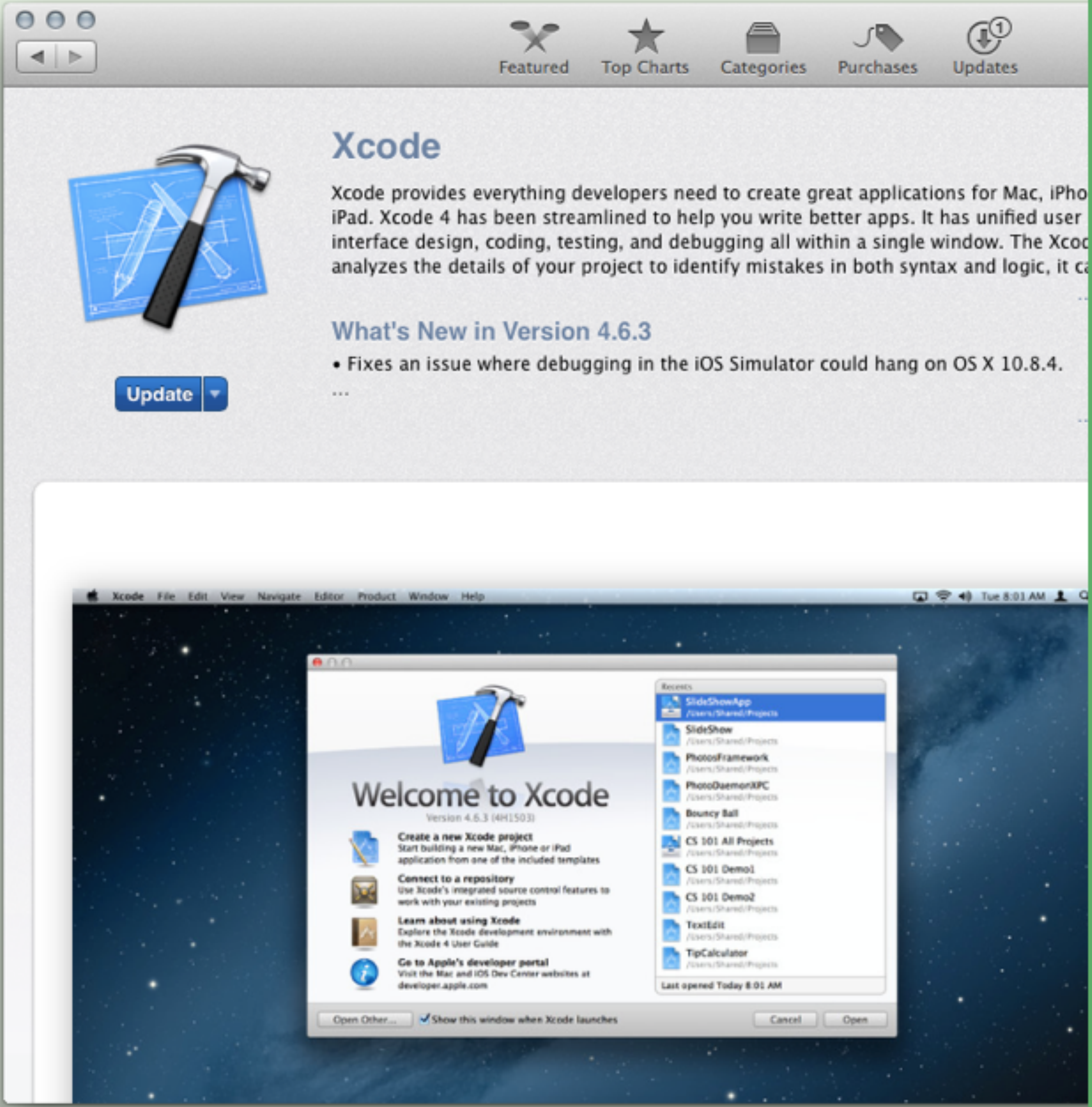
Installation

Installation

Mac

Unix

Win



Xcode

Xcode provides everything developers need to create great applications for Mac, iPho
iPad. Xcode 4 has been streamlined to help you write better apps. It has unified user
interface design, coding, testing, and debugging all within a single window. The Xcod
analyzes the details of your project to identify mistakes in both syntax and logic, it c

What's New in Version 4.6.3

• Fixes an issue where debugging in the iOS Simulator could hang on OS X 10.8.4.

...

Welcome to Xcode

Version 4.6.3 (4H1503)

Create a new Xcode project
Start building a new Mac, iPhone or iPad application from one of the included templates

Connect to a repository
Use Xcode's integrated source control features to work with your existing projects

Learn about using Xcode
Explore the Xcode development environment with the Xcode 4 User Guide

Go to Apple's developer portal
Visit the Mac and iOS Dev Center websites at developer.apple.com

Recents

SlideShowApp /Users/Shared/Projects
SlideShow /Users/Shared/Projects
PhotosFramework /Users/Shared/Projects
PhotoDaemonXPC /Users/Shared/Projects
Bouncy Ball /Users/Shared/Projects
CS 101 All Projects /Users/Shared/Projects
CS 101 Demo1 /Users/Shared/Projects
CS 101 Demo2 /Users/Shared/Projects
TextEdit /Users/Shared/Projects
TipCalculator /Users/Shared/Projects

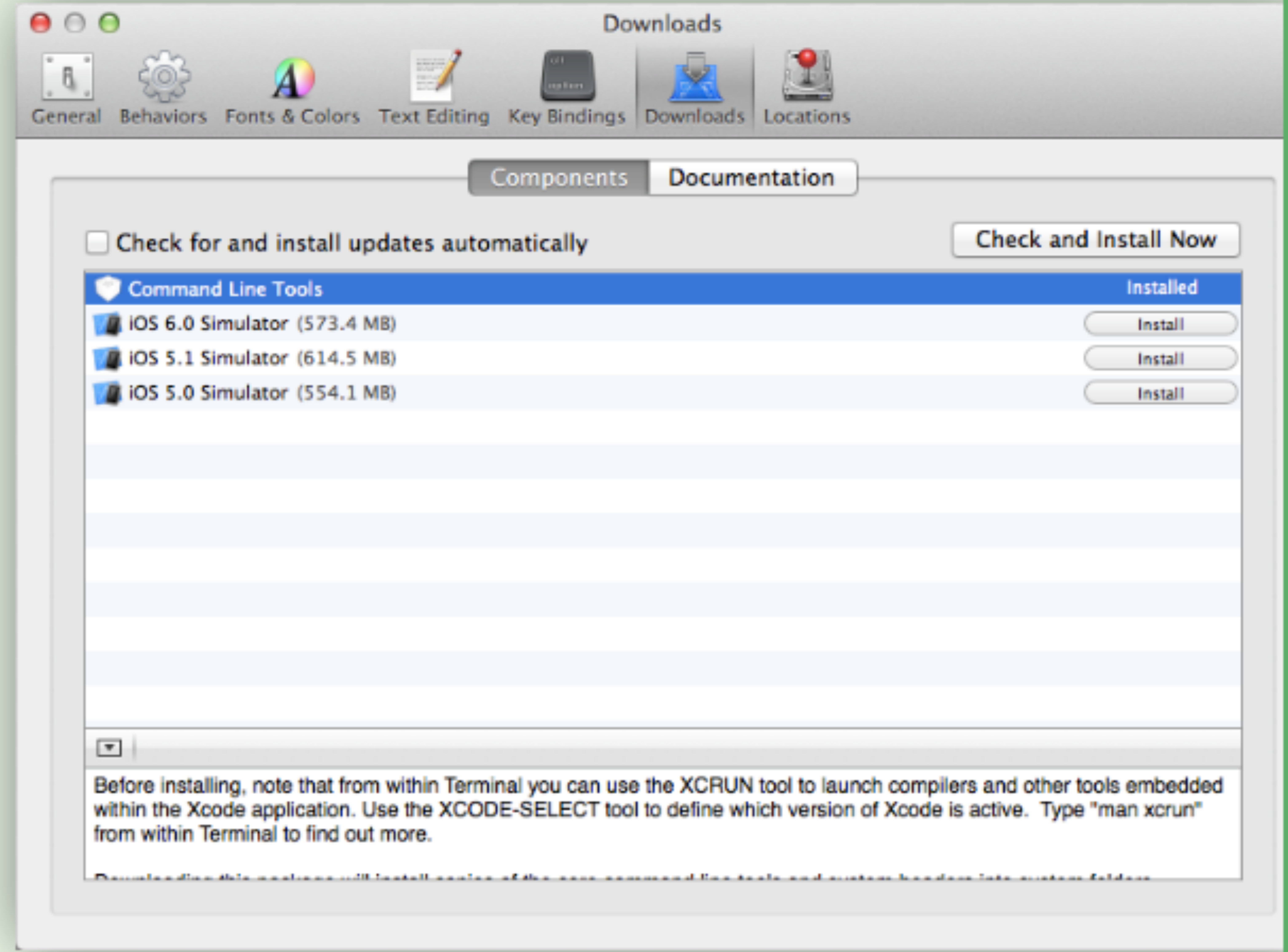
Last opened Today 8:01 AM

Installation

Mac

Unix

Win



Check for and install updates automatically

Command Line Tools	Installed
iOS 6.0 Simulator (573.4 MB)	Install
iOS 5.1 Simulator (614.5 MB)	Install
iOS 5.0 Simulator (554.1 MB)	Install

Before installing, note that from within Terminal you can use the XCRUN tool to launch compilers and other tools embedded within the Xcode application. Use the XCODE-SELECT tool to define which version of Xcode is active. Type "man xcrun" from within Terminal to find out more.

Installation
Mac
Unix
Win

Installing RVM
Installing Ruby
Installing Git
Installing MySQL
Installing Pow

Installation

Mac

Unix

Win

```
$ curl -L https://get.rvm.io | bash -s stable
```

installing RVM via one-liner command.
or check http://rvm.io for installation details.

Installation

Mac

Unix

Win

```
$ rvm install 1.9.4
$ rvm install 2.0.0
```

with RVM, we can manage mulitple versions of Rubies.

Installation

Mac

Unix

Win

```
$ rvm list

rvm rubies

=* ruby-1.9.3-p286 [ x86_64 ]
   ruby-2.0.0-p247 [ x86_64 ]

# => - current
# =* - current && default
#  * - default
```

listing all managed ruby versions.

Installation

Mac

Unix

Win

```
$ rvm use 1.9.3
$ gem install rails -v 3.2.8
```

specific the ruby version to use,
and installing the rails with version specified.

Installation	What is gem?
Mac	software package.
Unix	as a library.
Win	reusability.

Installation

Mac

Unix

Win

http://rubyinstaller.org

Creating Rails Project

Rails Project

create project

framework

file structure

```
$ rails new app_name
```

creating new rails project.

Rails Project

create project

framework

file structure

```
$ rails _3.2.8_ new app_name
```

specify rails version when creating project.

Rails is a framework

carefully designed for web app.

philosophy of best practices.

design pattern injected.

Rails Project

create project

framework

file structure

app/

config/

db/

log/

public/

vendor/

Gemfile

Rails Project

create project

framework

file structure

app/

major project files.

Rails Project

create project

framework

file structure

app/

assets/	for js/css files.
controllers/	business logic.
models/	data and validation.
views/	html outputs.

Rails Project

create project

framework

file structure

config/

all the configurations.

databases conncetion.

url routing.

Rails Project

create project

framework

file structure

db/

database schema.

database versions control.

Rails Project

create project

framework

file structure

log/

for debugging.

Rails Project

create project

framework

file structure

public/

files that can be view publicly.

images assets.

Rails Project

create project

framework

file structure

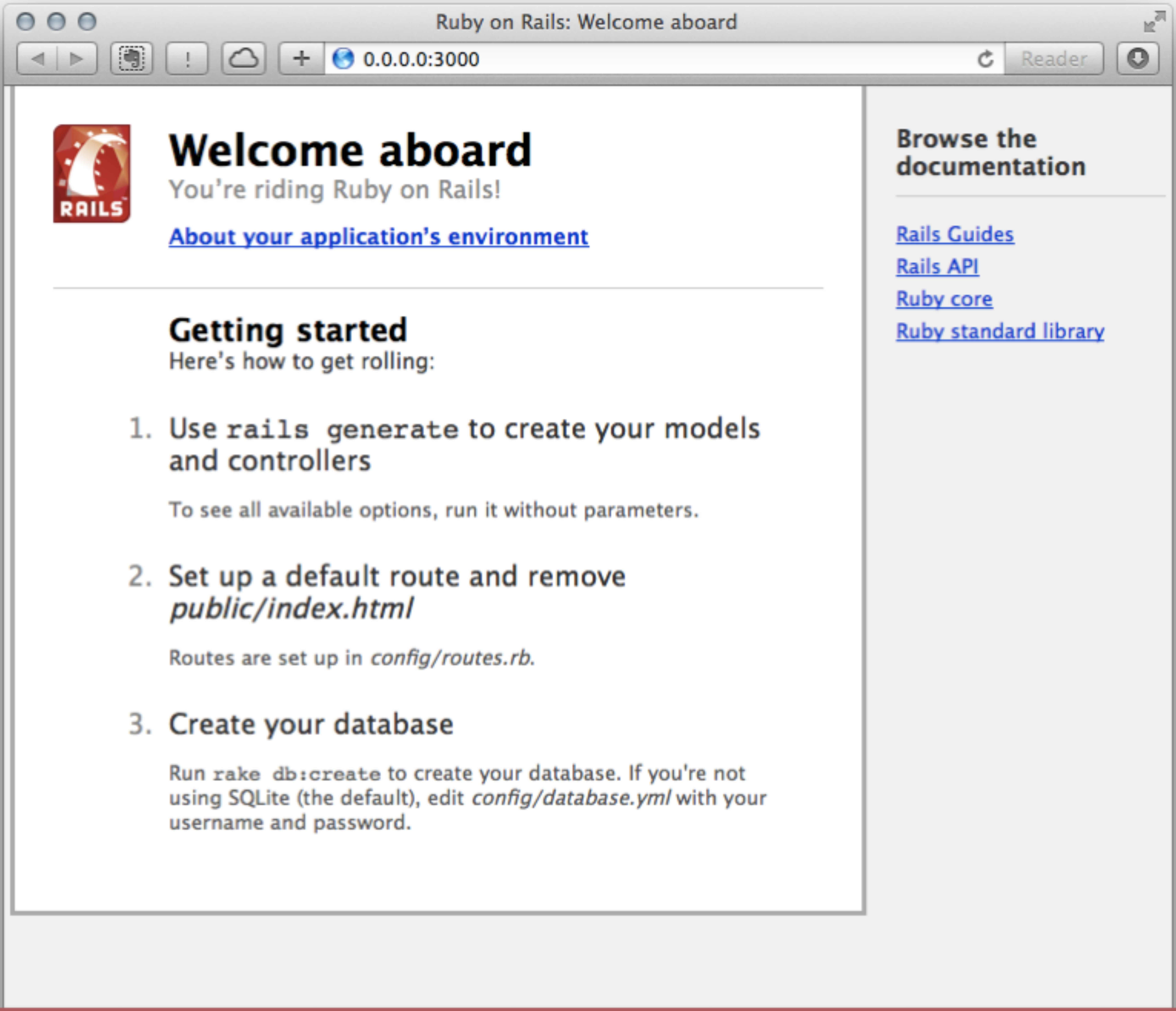
Gemfile

gem dependency control.

running the development server.

```
$ rails server
=> Booting WEBrick
=> Rails 3.2.8 application starting in development on http://0.0.0.0:3000
=> Call with -d to detach
=> Ctrl-C to shutdown server

[2013-09-06 14:32:54] INFO  WEBrick 1.3.1
[2013-09-06 14:32:54] INFO  ruby 1.9.3 (2012-10-12) [x86_64-darwin12.2.0]
[2013-09-06 14:32:54] INFO  WEBrick::HTTPServer#start: pid=17299 port=3000
```



Welcome aboard

You're riding Ruby on Rails!

About your application's environment

Getting started

Here's how to get rolling:

1. Use `rails generate` to create your models and controllers

   To see all available options, run it without parameters.

2. Set up a default route and remove *public/index.html*

   Routes are set up in *config/routes.rb*.

3. Create your database

   Run `rake db:create` to create your database. If you're not using SQLite (the default), edit *config/database.yml* with your username and password.

Browse the documentation

- Rails Guides
- Rails API
- Ruby core
- Ruby standard library

recources	tryruby.org
	nitrous.io
	guides.rubyonrails.org