

QUICK INTRO
TO
RUBY ON RAILS

... WHY LEARN IT?!

The Evolution Of Computer Programming Languages

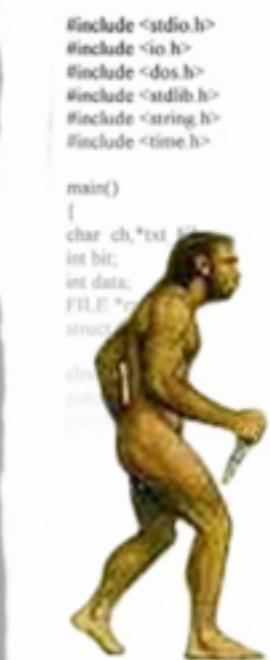


```
4D 54 68 64 00 00 00 06 00  
72 6B 00 00 00 61 00 F0 0A  
00 41 F7 00 B0 00 00 00 C0  
5A 32 01 00 00 00 00 00 00  
00 00 00 FF 51 03 06 8A 1B  
6A 6F 86 00 30 43 40 ZC 80  
00 42 00 04 90 41 5E 2C 00  
80 40 00 00 FF 27 00 4D 54  
85 00 00 00 C6 04 00 FF 7F  
00 00 00 00 00 00 00 00 00  
03 07 65 2D 70 69 61 6E 6F  
24 00 81 48 2A 40 XC 2A 00  
00 3F 00
```

Hex



Assembler



C



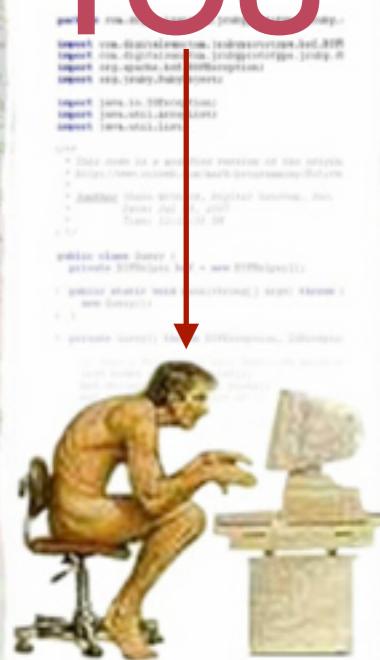
Fortran



C++



Java



Ruby

YOU

```
private void main() {  
    System.out.println("Hello World");  
}  
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

C, Java, C++, PHP, JavaScript, Python,
C#, Perl, SQL, Ruby, Shell, Visual
Basic, Assembly, Actionscript,
Objective C, Lisp, **Where do you**
start? Delphi, Pascal, Scheme,
Haskell, Tcl, Backbone, Fortran, Ada,
Lua, ColdFusion, Cobol, Erlang, D,
Scala, Smalltalk, Ocaml, Forth, Rexx,

They're all the same, just different

PHP

```
echo "Hello World";
```



Hello World

Python

```
print("Hello World")
```



Hello World

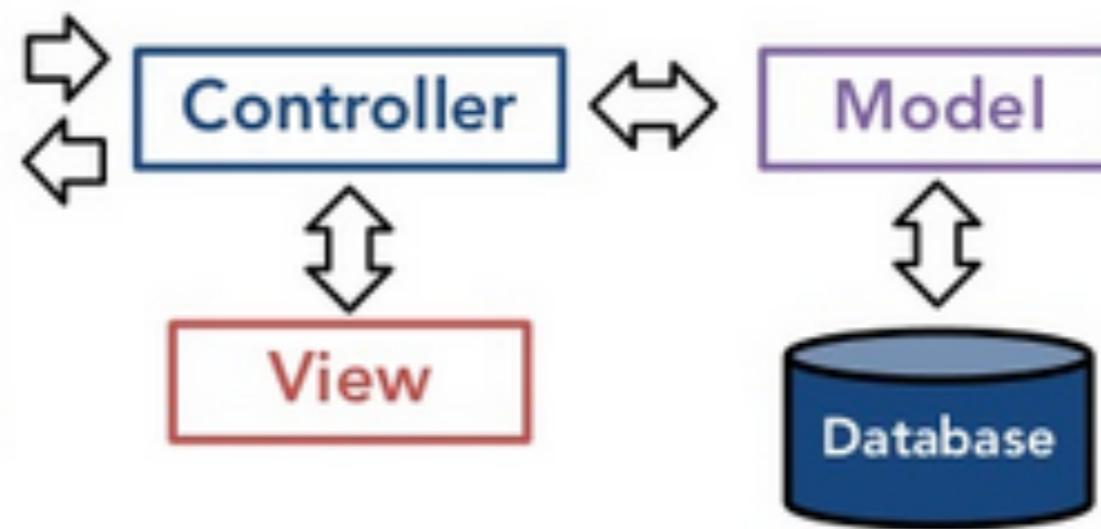
Ruby

```
puts "Hello World"
```

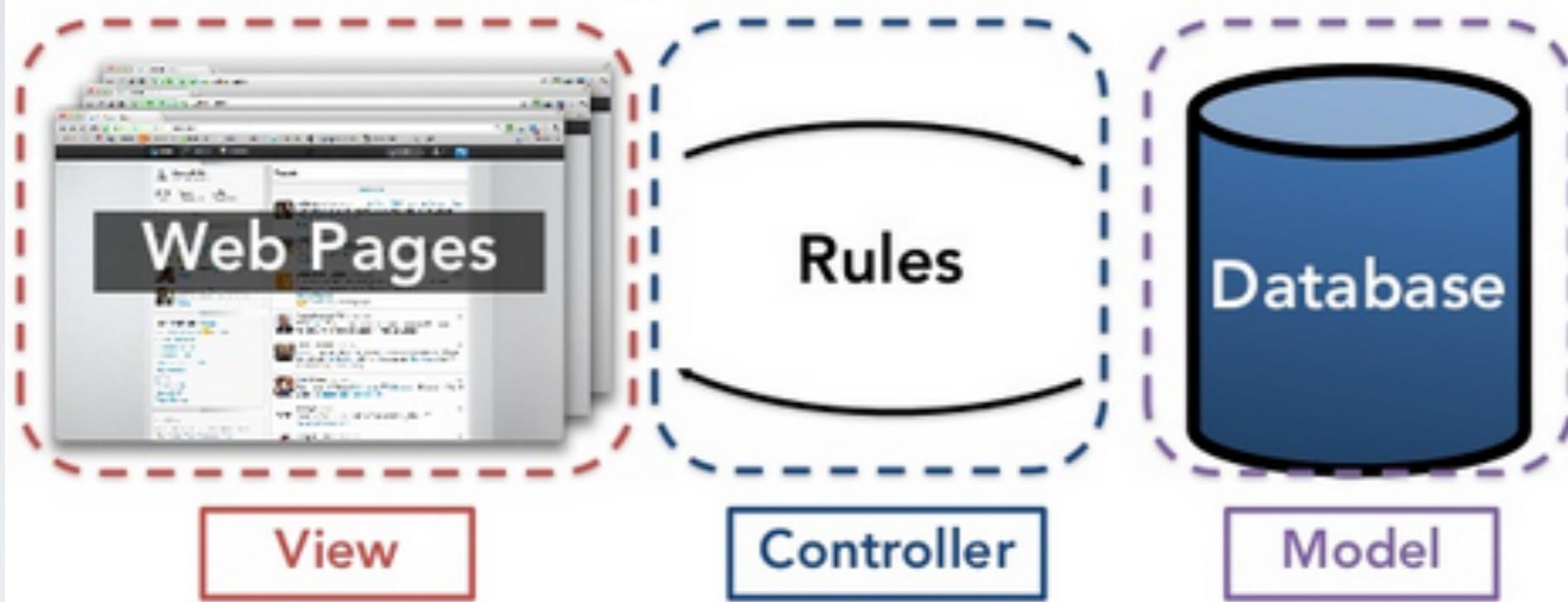


Hello World

Rails uses a Model View Controller **(MVC)** architecture



Rails uses a Model View Controller (MVC) architecture



**WHAT IS A
WEB APPLICATION?**

The **front-end** is what you see



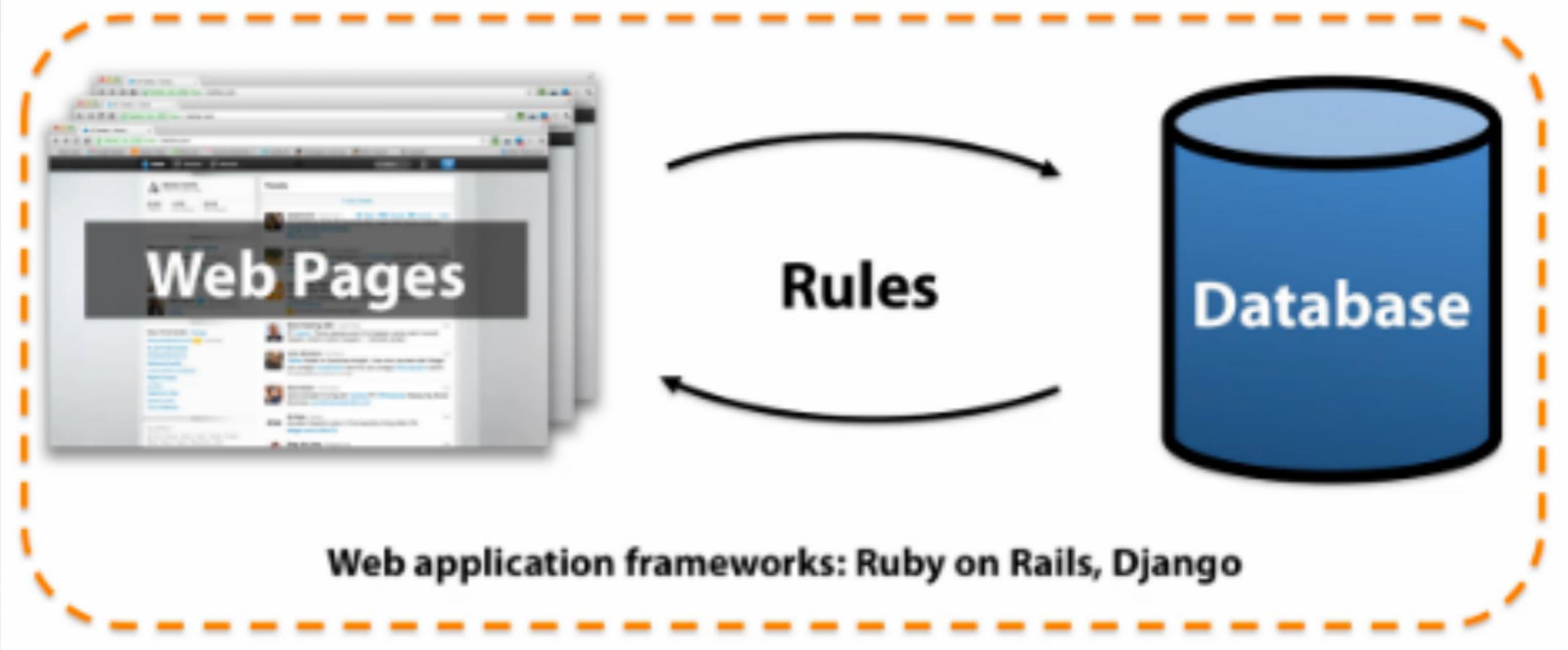
Front-end languages:

- **HTML**
- **CSS**
- **JavaScript**

The **back-end** is what you don't see



And then there are **web application frameworks**



Web application frameworks

“alleviate overhead associated with web development by providing libraries for database access, templating frameworks, and session management. Blah blah blah blah blah blah.”

**build web apps
really quickly**

If your goal is to build a
prototype, start with
Ruby on Rails
(aka ‘Rails’)



Ruby on Rails
is a web application framework
for Ruby

Rails is the **easiest to dive into**
(tons of resources and a huge community)

Rails **hides all the stuff you
really don't need to know**

Rails is just shortcuts like this

rails new application 

```
create
create README.rdoc
create Rakefile
create config.ru
create .gitignore
create Gemfile
create app
create app/assets/images/rails.png
create app/assets/javascripts/application.js
create app/assets/stylesheets/application.css
create app/controllers/application_controller.rb
create app/helpers/application_helper.rb
create app/mailers
create app/models
create app/views/layouts/application.html.erb
create app/mailers/.gitkeep
create app/models/.gitkeep
create config
create config/routes.rb
create config/application.rb
create config/environment.rb
.

.

create vendor/plugins
create vendor/plugins/.gitkeep
run bundle install
```

With those languages, you'll spend **MONTHS learning the basics so that you can do this:**



(make a yellow square)

**With Rails, fast and simple
is the name of the game**

Yes

Some of the most heavily trafficked
sites in the world use Rails



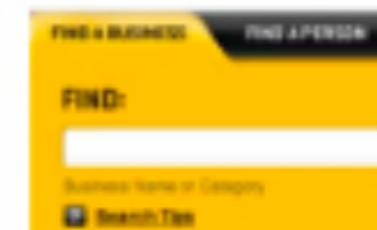
Basecamp



Twitter



Shopify



Yellow Pages



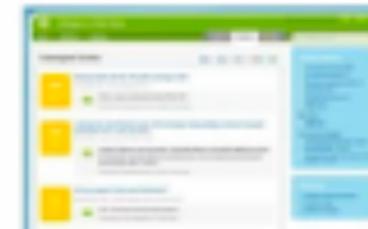
Hulu



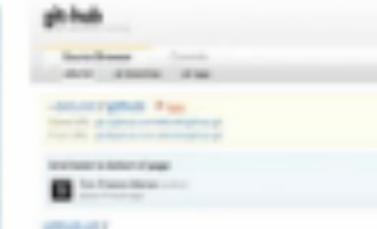
LivingSocial



Soundcloud



Zendesk



Github



Groupon



A lot of you just have
an idea

If you're expecting to find someone
to build that idea for you, it's
never going to happen

The point is for you to learn to
code well enough to
build a prototype

(aka Minimum Viable Product)

“Product with the **fewest
number of features needed to
achieve a specific objective.”**

– The Entrepreneurs Guide to Customer Development

More and more, the things we interact with in the world around us qualifies as “technology”

Society is quickly dividing into two groups:

1)

Those that know how to code –
they can manipulate the very structure
of the world around them.

2)

Those that don't – their lives are
being designed and directed by
those that do.

If you're interested in learning more, read Program or Be Programmed: <http://amzn.to/yzd6hm>



I get it,
it's scary to
jump in.

Ruby on Rails Tutorial Sample App | Home

localHost 3000

SAMPLE APP

Home Help Users Account

Example User
[view my profile](#)
50 microposts

0 following 50 followers

Compose new micropost...

Post

Micropost Feed

Katherine Brown
Repellendus velit tenetur pariatur et earum perferendis odio qui dolor.
Posted about 1 hour ago.

Bulah Hahn DVM
Repellendus velit tenetur pariatur et earum perferendis odio qui dolor.
Posted about 1 hour ago.

Miss Nelly Morris
Repellendus velit tenetur pariatur et earum perferendis odio qui dolor.
Posted about 1 hour ago.

Sincere Vandenberg
Repellendus velit tenetur pariatur et earum perferendis odio qui dolor.
Posted about 1 hour ago.

Example User
Repellendus velit tenetur pariatur et earum perferendis odio qui dolor.
Posted about 1 hour ago.

When you finish, you'll be able to build your own **basic web app**

Miss Nelly Morris
Aperiam ut deseruntibus quasi dolor minima placeat.
Posted about 1 hour ago.

Compose reply 2 years ago
Officiis ut consequuntur dicitur optio, necessitatibus dolorem

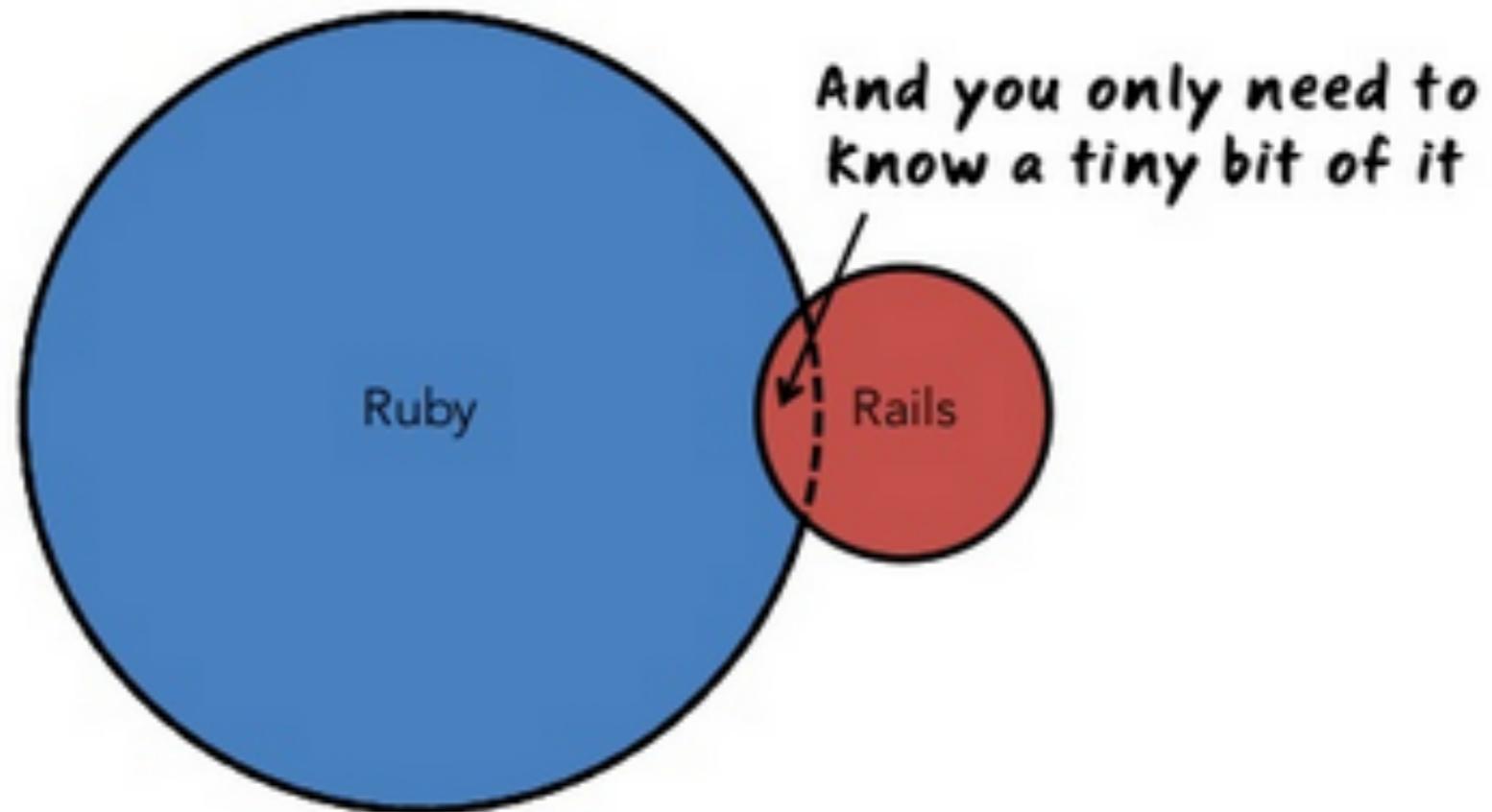
pmjq lotx omu psac mep sbb

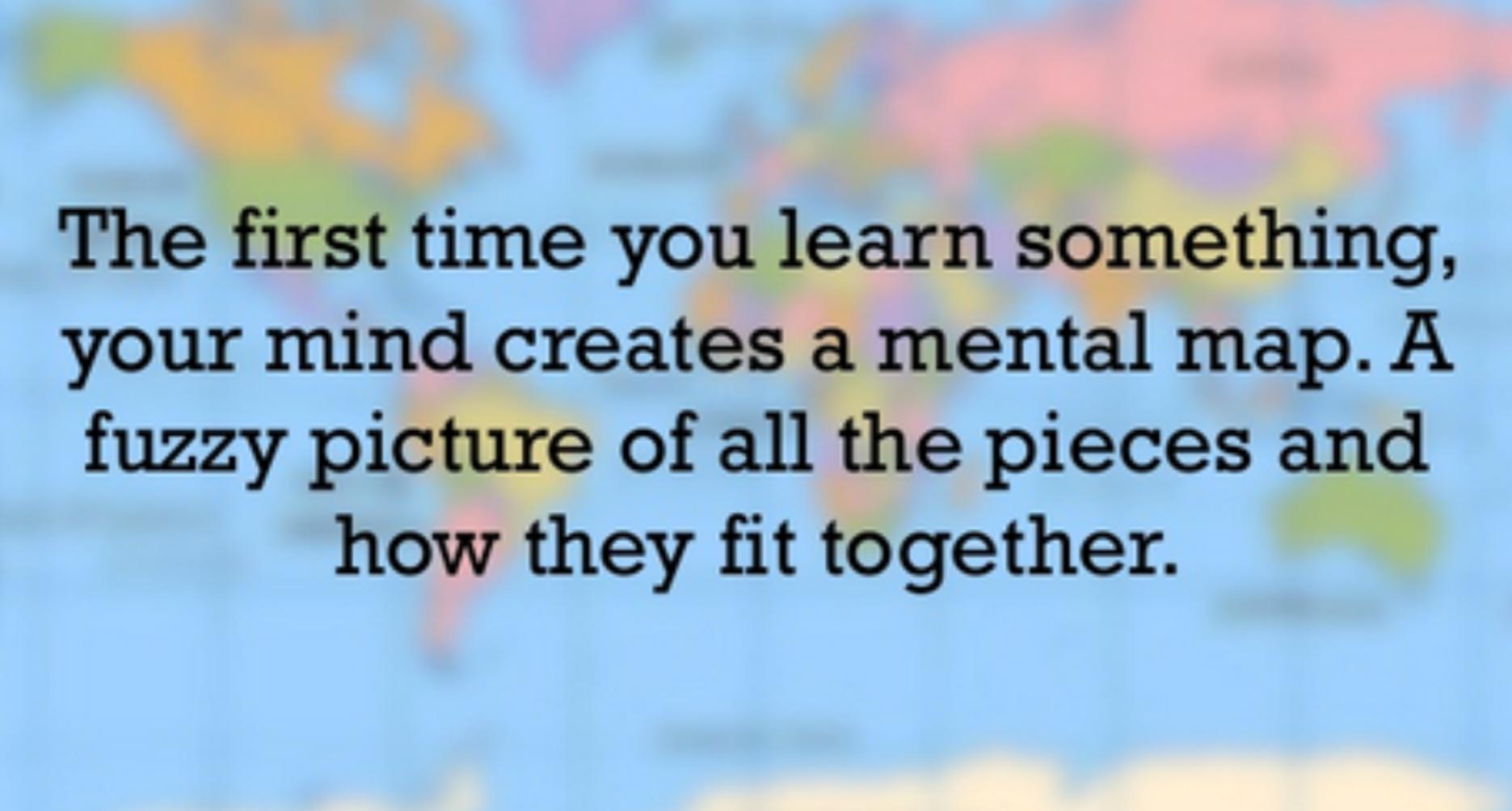
But there will be **specific**
features you don't know how
to implement

At some point along the way, you may feel a temptation to **STOP** and learn Ruby (or HTML or CSS)

Don't.

Ruby is huge





The first time you learn something,
your mind creates a mental map. A
fuzzy picture of all the pieces and
how they fit together.



Windows

A Fatal exception 0E has occurred at 0028:C0011E36 in UXD UMH(01) +
00010E36. The current application will be terminated.

- * Press any key to terminate the current application.
- * Press CTRL+ALT+DEL again to restart your computer. You will
lose any unsaved information in all applications.

Press any key to continue ...

Troubleshooting

troubleshooting



75% of coding is Googling*

*totally made up statistic

Questions Tags Users Badges Unanswered Ask Question

Top Questions Interesting featured hot week month

0 votes 0 answers 1 view Showing large UIImage causes jittering even though it is already in memory 46s ago Lope 574

0 votes 0 answers 3 views How can I leverage C++ code in the JVM? 1m ago sangho 348

0 votes 3 answers 36 views why is \$_SERVER['REMOTE_ADDR'] returning 127.0.0.1? 1m ago el7r 18

1 vote 1 answer 12 views MySQL Connector/J with multiple SQL statements 1m ago GregNash 22

0 votes 0 answers 2 views Using Maven, in which directory should I place the image files for my icons? 1m ago GregNash 22

Stack Overflow

Help

Log in | careers | chat | meta | about | faq | search

Hello World!

This is a collaboratively edited question and answer site for professional and enthusiast programmers. It's 100% free, no registration required.

about » faq »

{TEX}

How to make the optional title of a theorem bold with amsthm?

A lot of times you know what feature you want to build, but **you don't know what it's called**



For example: **Cron jobs** let you schedule tasks in your app

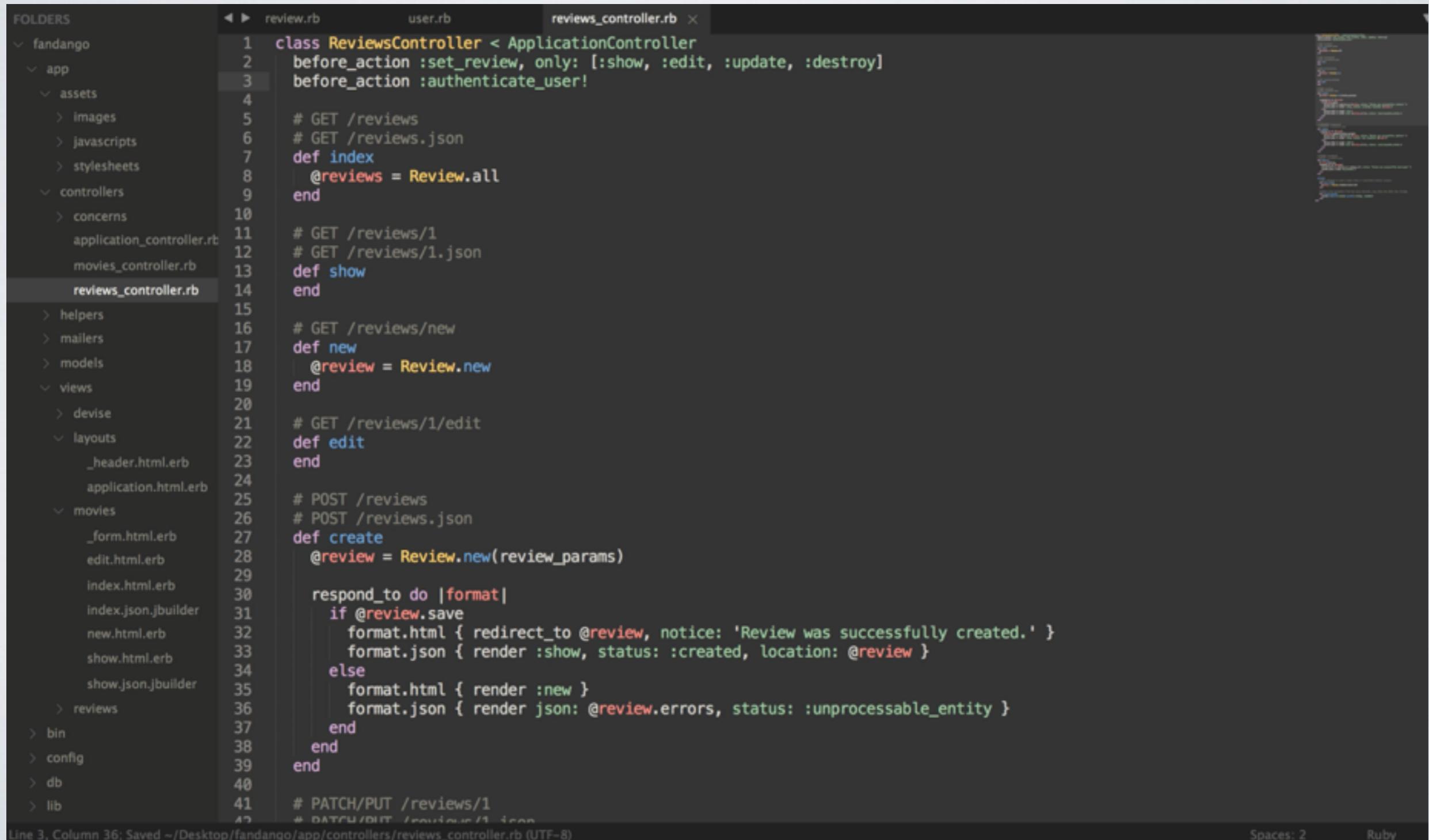
Rails is very popular at
the moment and has a

LARGE
community for support

TOOLS YOU'LL USE

The Text Editor is where you write your code (I prefer Sublime Text) ...

it's also part of the giveaway for taking this course!



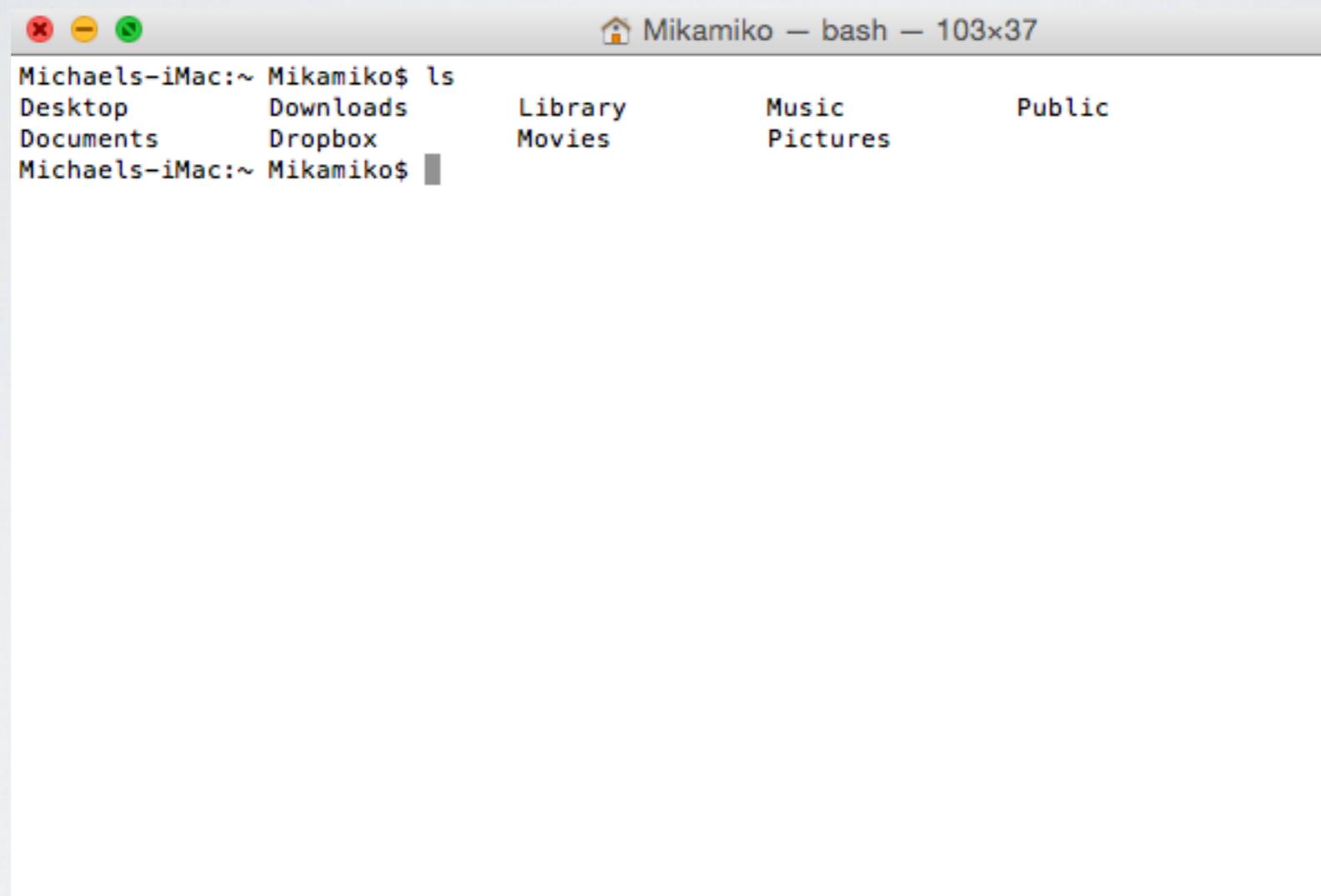
A screenshot of the Sublime Text code editor. On the left, there is a file tree titled "FOLDERS" showing the project structure of a Ruby on Rails application named "fandango". The "controllers" directory contains files like "application_controller.rb", "movies_controller.rb", and "reviews_controller.rb", with "reviews_controller.rb" currently selected. The main pane displays the content of "reviews_controller.rb". The code defines a `ReviewsController` class that inherits from `ApplicationController`. It includes two before actions: `:set_review` (only for :show, :edit, :update, :destroy) and `:authenticate_user!`. The controller has methods for index, show, new, edit, and create. The `create` method uses `respond_to` to handle both HTML and JSON formats, saving the review and redirecting or rendering accordingly. The `show` method handles PATCH/PUT requests. The code is color-coded for syntax highlighting.

```
review.rb          user.rb           reviews_controller.rb ×
1  class ReviewsController < ApplicationController
2    before_action :set_review, only: [:show, :edit, :update, :destroy]
3    before_action :authenticate_user!
4
5    # GET /reviews
6    # GET /reviews.json
7    def index
8      @reviews = Review.all
9    end
10
11   # GET /reviews/1
12   # GET /reviews/1.json
13   def show
14     end
15
16   # GET /reviews/new
17   def new
18     @review = Review.new
19   end
20
21   # GET /reviews/1/edit
22   def edit
23     end
24
25   # POST /reviews
26   # POST /reviews.json
27   def create
28     @review = Review.new(review_params)
29
30     respond_to do |format|
31       if @review.save
32         format.html { redirect_to @review, notice: 'Review was successfully created.' }
33         format.json { render :show, status: :created, location: @review }
34       else
35         format.html { render :new }
36         format.json { render json: @review.errors, status: :unprocessable_entity }
37       end
38     end
39   end
40
41   # PATCH/PUT /reviews/1
42   # PATCH/PUT /reviews/1.json
```

Line 3, Column 36; Saved ~/Desktop/fandango/app/controllers/reviews_controller.rb (UTF-8)

Spaces: 2 Ruby

The command line is where you run your code



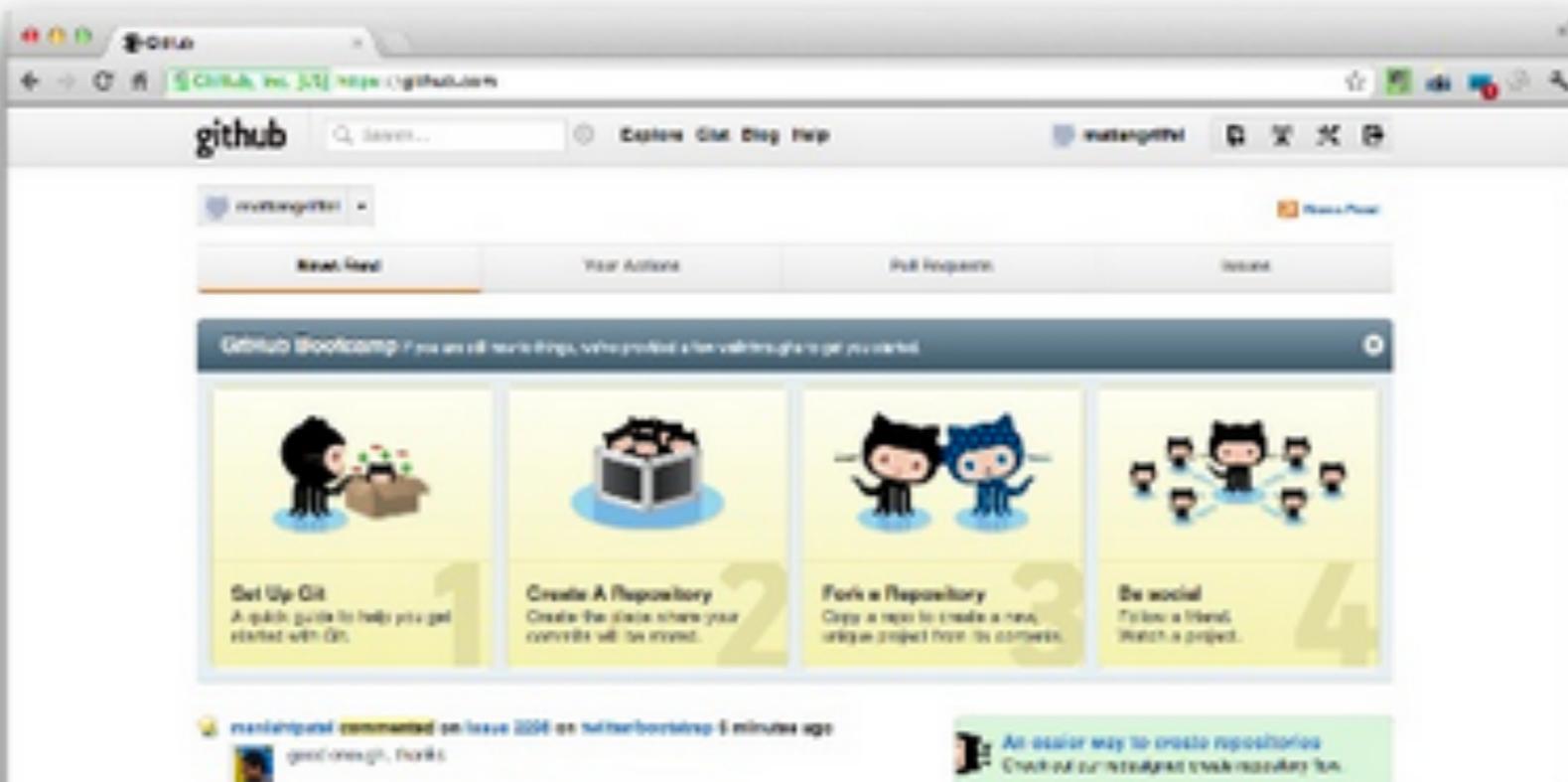
```
Michaels-iMac:~ Mikamiko$ ls
Desktop      Downloads      Library      Music       Public
Documents    Dropbox       Movies       Pictures
Michaels-iMac:~ Mikamiko$
```

The browser is
where you see
the result



Google Chrome

Github is where you save versions of your project files



LET'S BEGIN
THE
CHALLENGE!