Lesson 7
Getting on the Rails

What is Ruby on Rails?

Ruby on Rails is a full-stack web application framework built in Ruby

Full stack – web server interaction, database interaction, template rendering, URI routing

Developed by David Heinemeier Hansson while with 37Signals

Released as open source in February 2005

What is Ruby on Rails?

Active Record
Database tables represented as classes whose objects represent individual records

Convention over configuration
Standard naming and location conventions varied only if necessary

Don't repeat yourself (DRY)
Every piece of knowledge should have a single source in the system

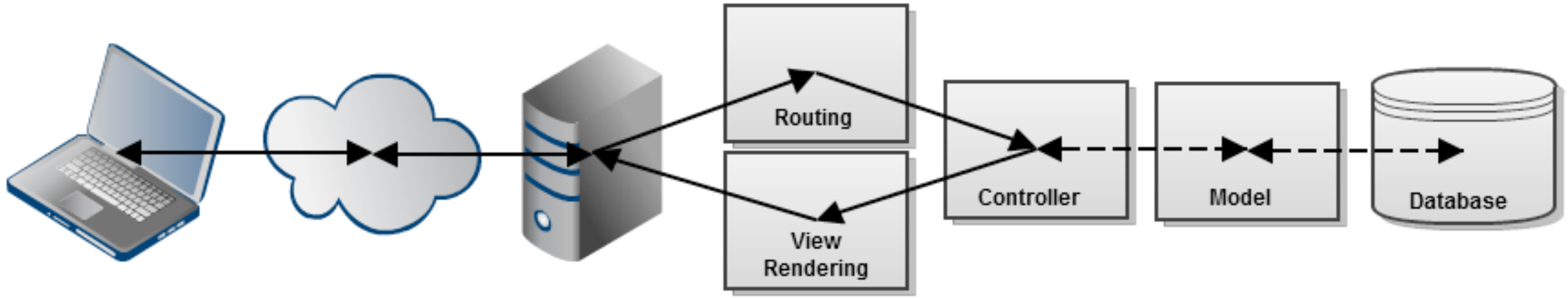
Model-View-Controller (MVC)
Data is kept distinct from both its appearance and how it is manipulated

Who is using Ruby on Rails?

Basecamp
Twitter
Groupon
Github
Soundcloud
Kongregate
Shopify
Livingsocial
and more

http://rubyonrails.org/applications

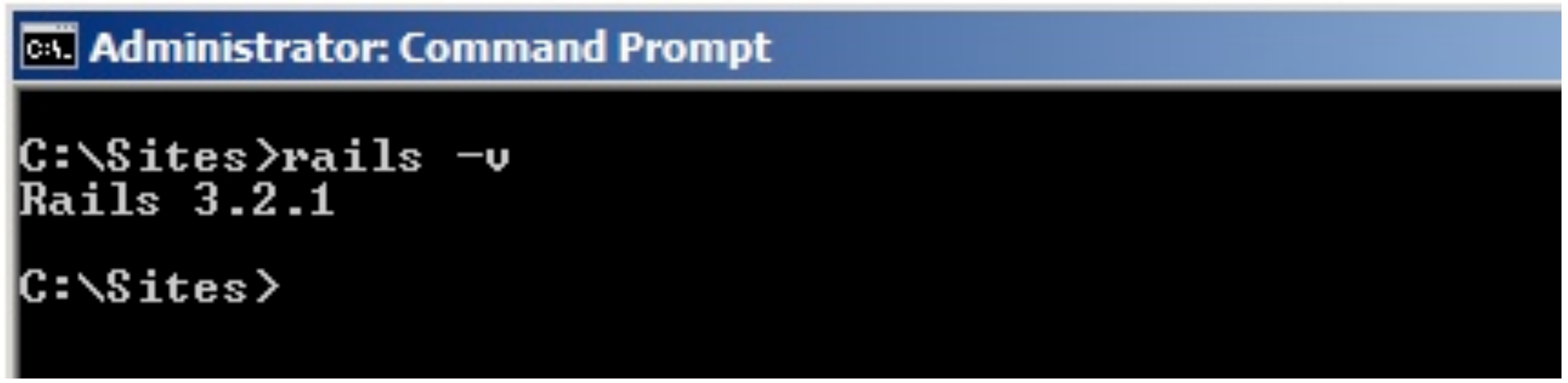
How does Rails process a web request?



Do I have Rails installed?

Open your terminal or command window

Type in `rails -v` command to check your installed version



```
C:\Sites>rails -v
Rails 3.2.1

C:\Sites>
```

How do I install Rails again?

You should have already have it but if you don't...

The harder way
http://rubyonrails.org/download

The very easy way
http://railsinstaller.org/

What are RubyGems?

RubyGems is the standard package manager for Rails

Package manager: a standard format, installer, and distribution server for programs and libraries

A library is reusable code used by a program achieve its purpose

A gem is Ruby library organized in a defined way

The command line tool `gem` installs specified gems

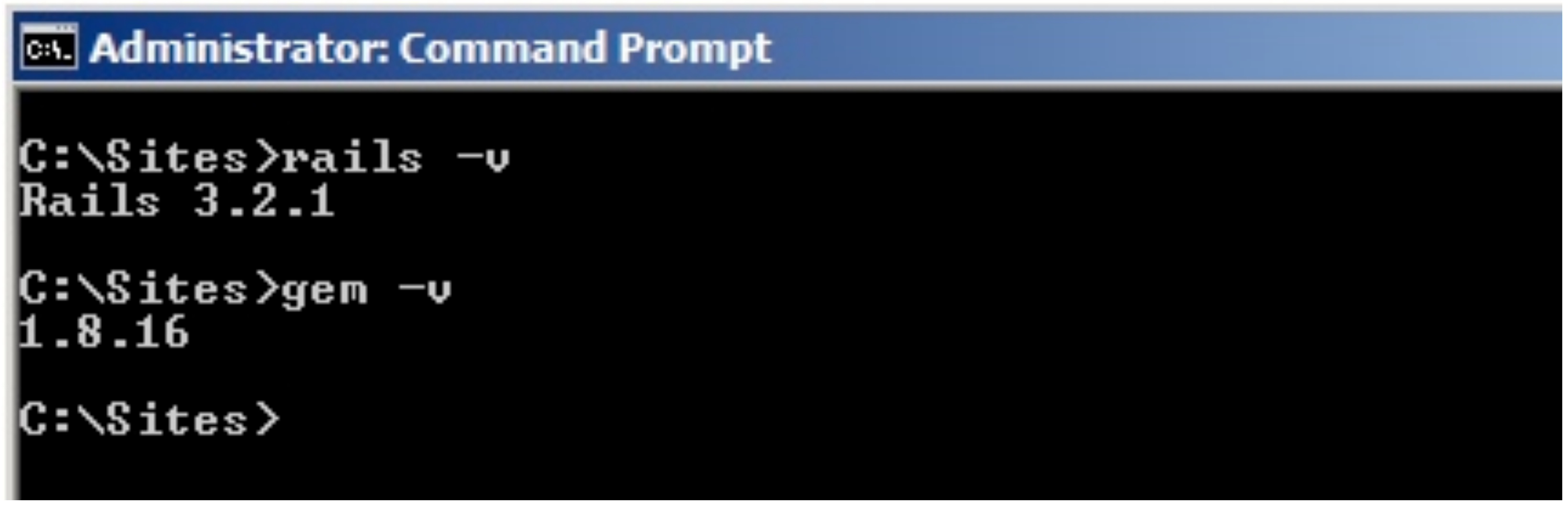
Where can I find gems?

https://www.ruby-toolbox.com/ is amazing.

Frequently used sources include:
http://rubygems.org/
http://rubyforge.org/

How do I know if RubyGems is installed?

Use the `gem -v` command to check for your installed version

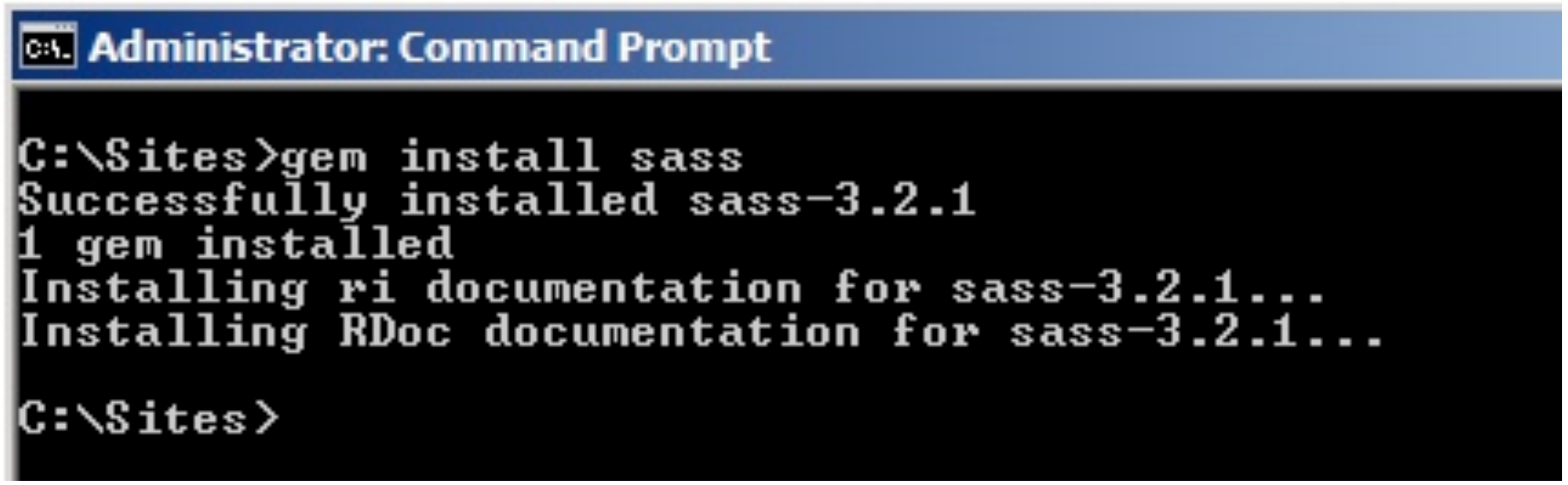


```
C:\Sites>rails -v
Rails 3.2.1

C:\Sites>gem -v
1.8.16

C:\Sites>
```

How do install a gem?

Use the `gem install <gem name>` command

e.g. Sass is a widely used gem for simplifing and extending CSS



```
C:\Sites>gem install sass
Successfully installed sass-3.2.1
1 gem installed
Installing ri documentation for sass-3.2.1...
Installing RDoc documentation for sass-3.2.1...

C:\Sites>
```

Once installed, gems are generally made available:
– as libraries you may require in your code.
– command line tools you may use

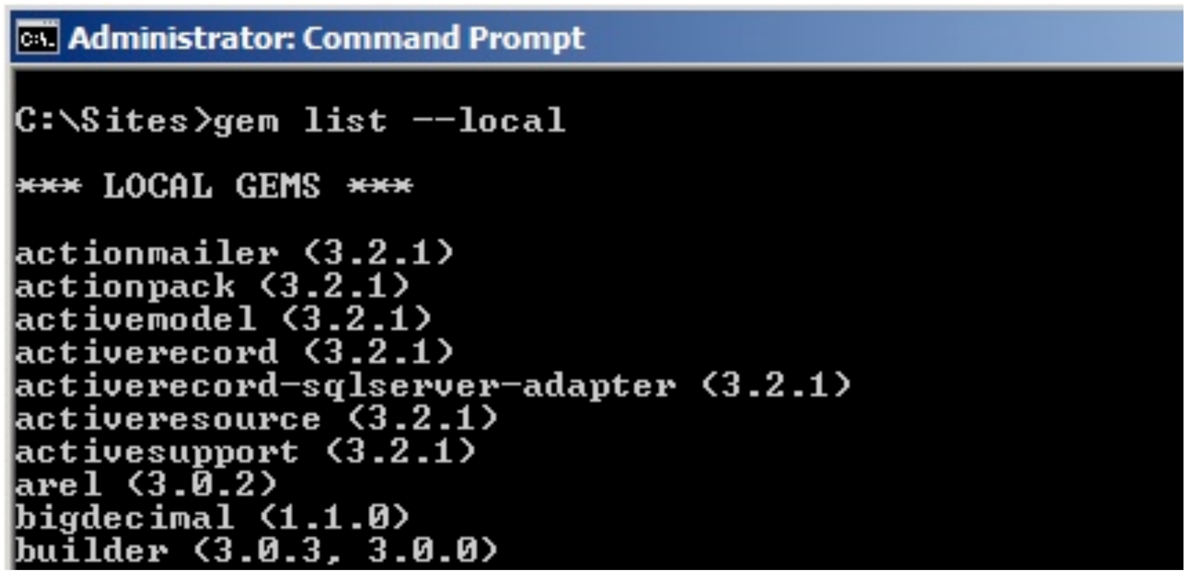
How do I know what Gems I have installed?

```
gem --help
```

to reach the help section

```
gem list --local
```

to see what gems you have installed


Administrator: Command Prompt

```
C:\Sites>gem list --local

*** LOCAL GEMS ***

actionmailer (3.2.1)
actionpack (3.2.1)
activemodel (3.2.1)
activerecord (3.2.1)
activerecord-sqlserver-adapter (3.2.1)
activeresource (3.2.1)
activesupport (3.2.1)
arel (3.0.2)
bigdecimal (1.1.0)
builder (3.0.3, 3.0.0)
```

Exercise:
Playing with gems

What are some important command line tools?

```
rails new <app_name>
```
creates a new application

```
rails server
```
launches a local development web server to enable local test requests

```
rails generate <generator> <args> <options>
```
generate boilerplate code for an asset type

```
rake
```
used to execute a wide variety of administrative tasks

```
rails console
```
create an IRB sandbox to safely test code for your application

Introducing Rails routing

URLs used with Rails apps do not point to a specific directory

Instead, Rails examines the URL of incoming HTTP requests and routes them to specified code

Which URIs are routed to which code is configured in
`config/routes.rb`

A default routing rule at the bottom of routes.rb can be enabled (uncommented) for simpler applications

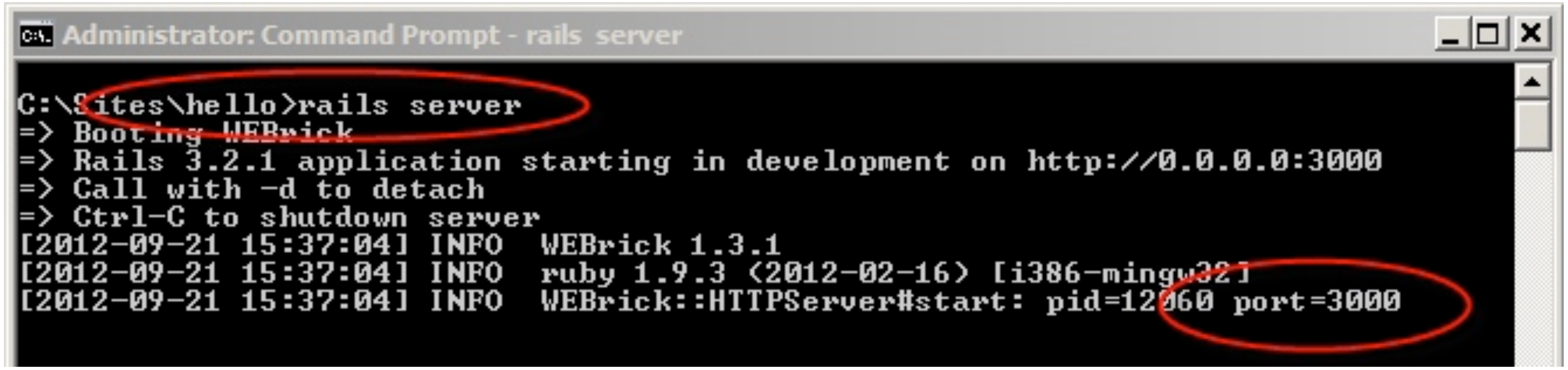
```
match ':controller(/:action(/:id))(.:format)'
```

Launching the development web server

Rails includes the WEBrick HTTP server to enable URI testing during development

In a terminal or command window, navigate to your application root folder, such as `/Sites/pfr/<app_name>`

Use the `rails server` command



```
C:\Sites\hello>rails server
=> Booting WEBrick
=> Rails 3.2.1 application starting in development on http://0.0.0.0:3000
=> Call with -d to detach
=> Ctrl-C to shutdown server
[2012-09-21 15:37:04] INFO  WEBrick 1.3.1
[2012-09-21 15:37:04] INFO  ruby 1.9.3 (2012-02-16) [i386-mingw32]
[2012-09-21 15:37:04] INFO  WEBrick::HTTPServer#start: pid=12060 port=3000
```

Exercise:
Running and requesting a Hello World application

What is in the app folder?

app/assets
Images, javascripts, and stylesheets to be requested by Rails templates

app/controllers
Switchboard classes which pass incoming data to models and providing outgoing data to views

app/helpers:
Utility classes which provide methods to be used across many views

app/mailers
support classes for sending email from the application

What is in the app folder?

app/models
Classes representing and providing access to database tables

app/views
ERB templates called to render data into pages

app/views/layouts
Template "master pages" to consolidate common view code (e.g., site menus, etc.)

What else is in Rails folder structure?

config
routing, database, and configuration files

db
scripts managing database tables

doc
location for documentation created with RubyDoc

lib
developer-built code to be shared across the application

log
system and developer generated log files

What else is in Rails folder structure?

public
Static HTML, standard error pagesand favicon.ico files

script
Code used by command line tools to generate other code

test
Code for mocks, unit tests, fixtures, and functional tests

tmp
Location for session variables, temp files, cached data, etc.

vendor
Location for external 3rd-party code as a plug-in

How do you create a controller and view?

A controller and view together create a requestable "page". To create a controller use the controller generator with the rails command:

```
rails generate controller <controller_name> <view_name>
```

If a view_name is provided then two things are created:

A template at app/views/<controller_name>/<view_name>.html.erb
An action named view_name in app/controllers/<controller_name>

As you may expect, an action and view named “index” will provide the default view for a controller

Exercise:
Creating an index page for the Hello World application

How does Rails understand a URI?

By default, Rails recognizes four main parts of a URI for routing purposes

http://server/controller/action/id.format

Compare this to the default route you saw in routes.rb
parentheses mean that portion is optional

```
match ‘:controller(/:action(/:id))(.:format)’
```

Parameter ("id") and format values are discussed in a later lesson

How does this relate to an actual URI?

```
http://server/controller/action/id.format
```

```
http://127.0.0.1:3000/hello/index.html
```

invokes the index action of the hello controller which calls index.html.erb to render the page

```
http://127.0.0.1:3000/hello/index/2.json
```

Invokes the index action of the hello controller, passing a parameter ("ID") value of 2 and calls index.json.erb to render the result

```
http://127.0.0.1:3000/hello/
```

invoke the index action of the hello controller by default

What is templating?

Templating is when a document (e.g. HTML for websites) has variables, conditions, and loops embedded in it

This allows a web page to be customized based on form or URL data sent in its request

A template is a text file designed to be processed by ERB

Ruby supports templating through its built-in ERB library

How does ERB work?

ERB loads and looks through a text file for (primarily) expressions and scriptlets. They are then run and the result is output wherever ERB is told.

By convention, ERB templates end with a .erb file extension

```
my_template.erb
```

By Rails convention, templates also include their format as a file extension. For example:

```
my_template.html.erb
my_template.json.erb
my_template.rss.erb
```

How does ERB work?

Expressions in the document are evaluated and replaced with their result

```
<p>
  Hi <%= @name %>! The date now is <%= Time.now %>
</p>
```

```
<p>
  Hi Fred! The date now is 2012-10-15 13:39:31 +0100
</p>
```

How does an ERB scriptlet work?

Scriptlets in the document are run as code in relation to any text they surround. Expressions and scriptlets are very often used in combination.

```
<ul>
    <% @employees.each do |worker| %>
    <li><%= worker %></li>
    <% end %>
</ul>
```

Lab