

- ❖ **Ruby Classes, Objects, & Instances**
- ❖ **JSON APIs**
- ❖ **Rails 101**
- ❖ **MVC Architecture**
- ❖ **User Input**

Get The Code

```
cd ~/dev/uc
```

```
git clone git://github.com/cspp52553/week2.git
```

Demo + Lab

Goal: Display a list of Chicago landmarks

Each landmark should have two attributes:

- Name**
- Admission Fee**

Use an Array of Landmark instances.

Use puts statements to display the data.

Consuming JSON APIs

What is an API?

JSON notation

Converting JSON into a Ruby hash

What is an API?

**An Application Programming Interface
enables
computer-to-computer
communication.**

What is an API?

**An Application Programming Interface
is an agreement that enables
computer-to-computer
communication.**

Broad Categories of APIs

XML-RPC

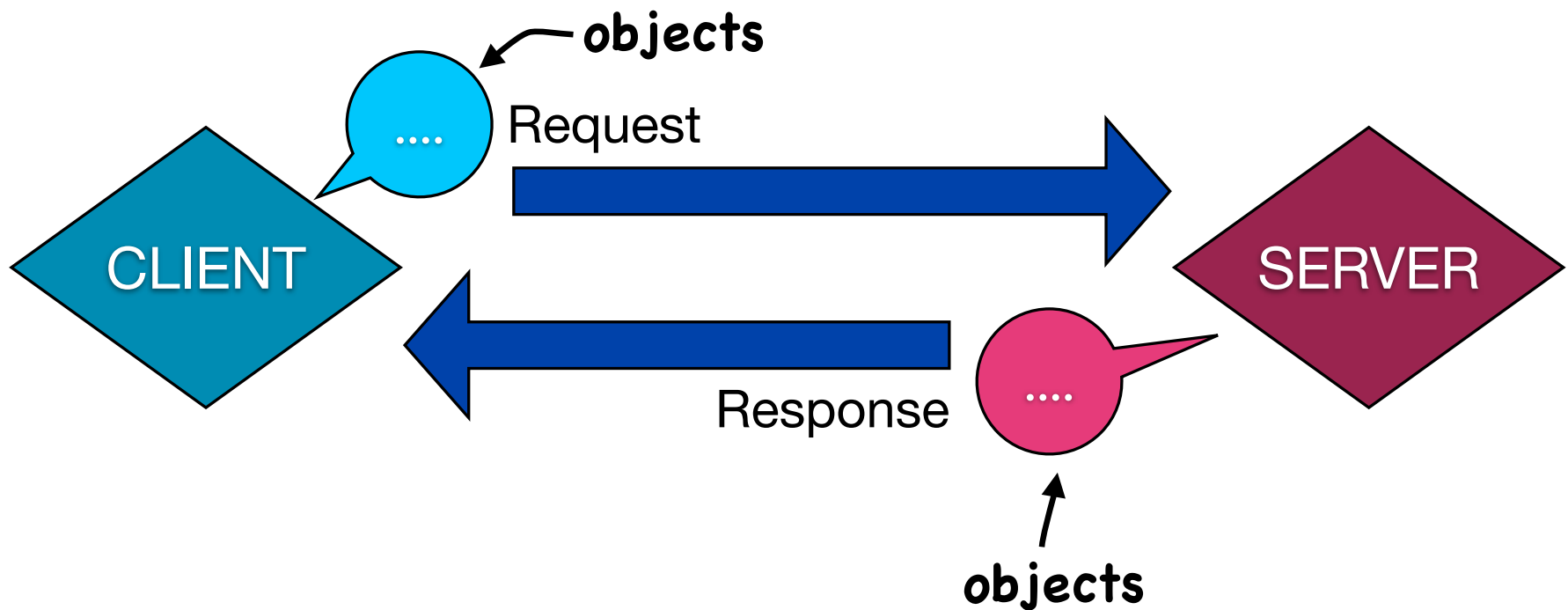
SOAP

"RESTful"

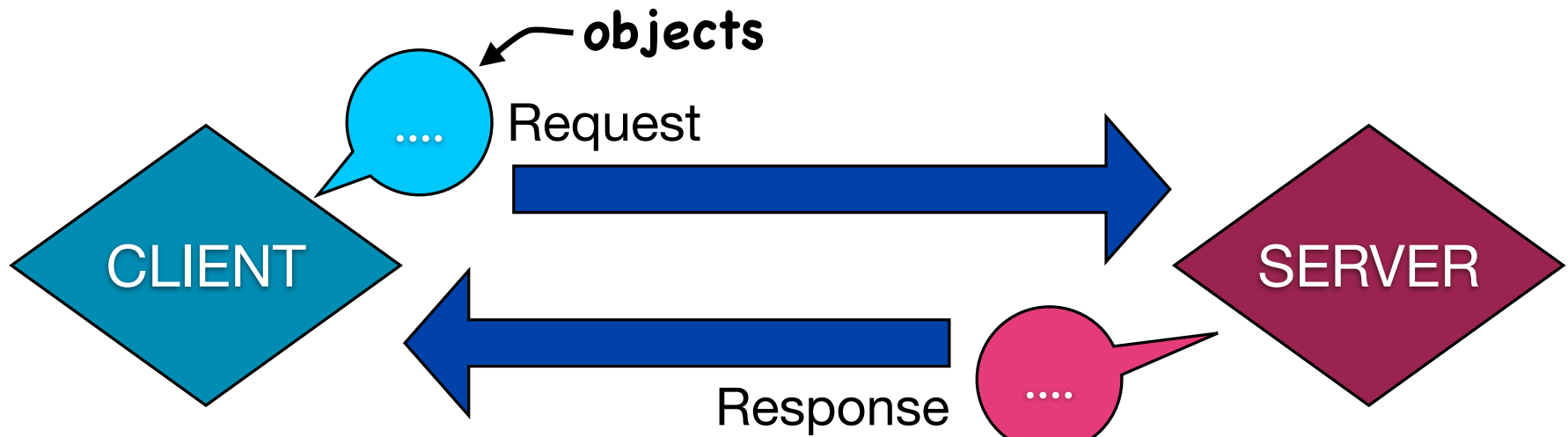
Broad Categories of APIs



How Does An API Work?



JSON APIs



**Javascript Object Notation
(JSON)**

(JSON)

Calling an HTTP API with Ruby

```
require 'json'  
require 'open-uri'
```

```
data = open("http:...").read
```

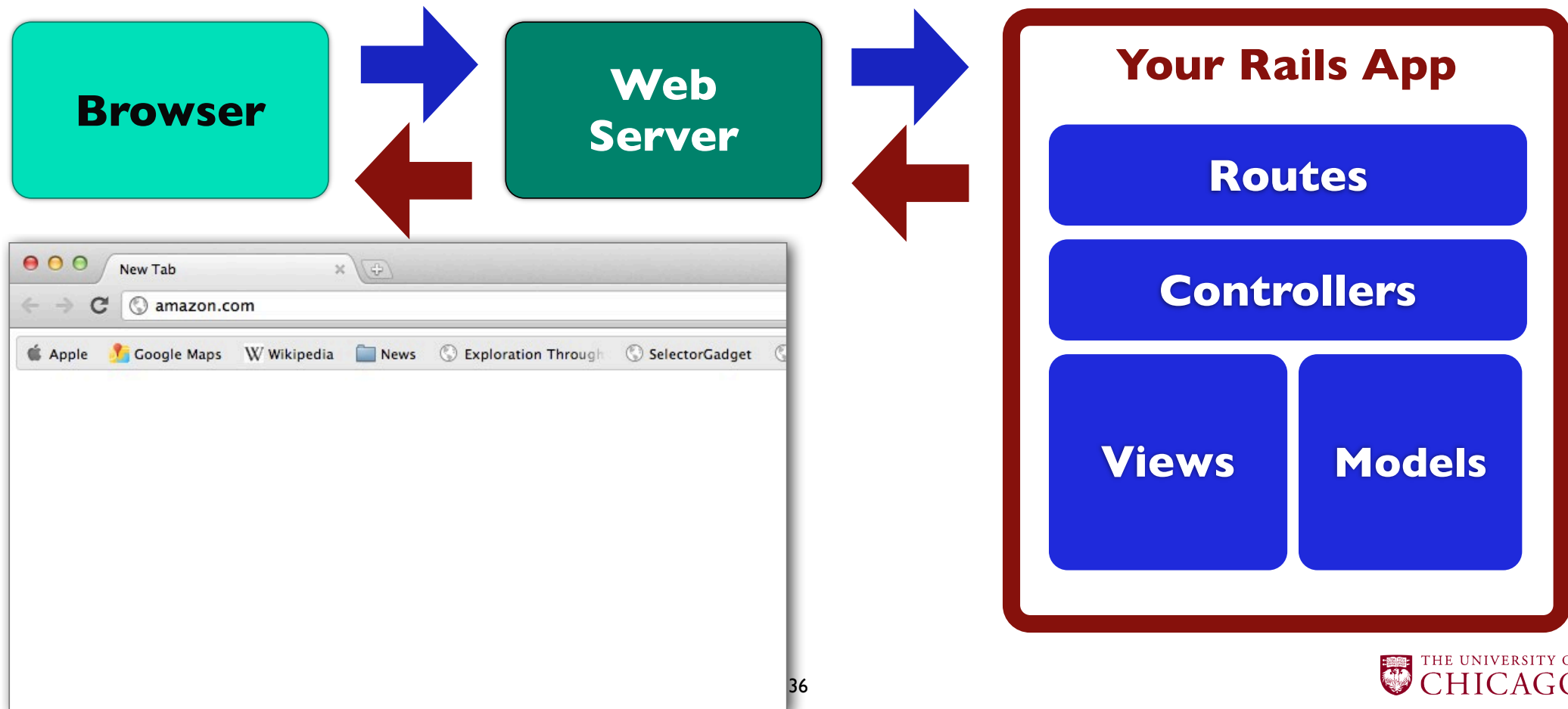
```
h = JSON.parse(data)
```

Challenge: HTTP APIs

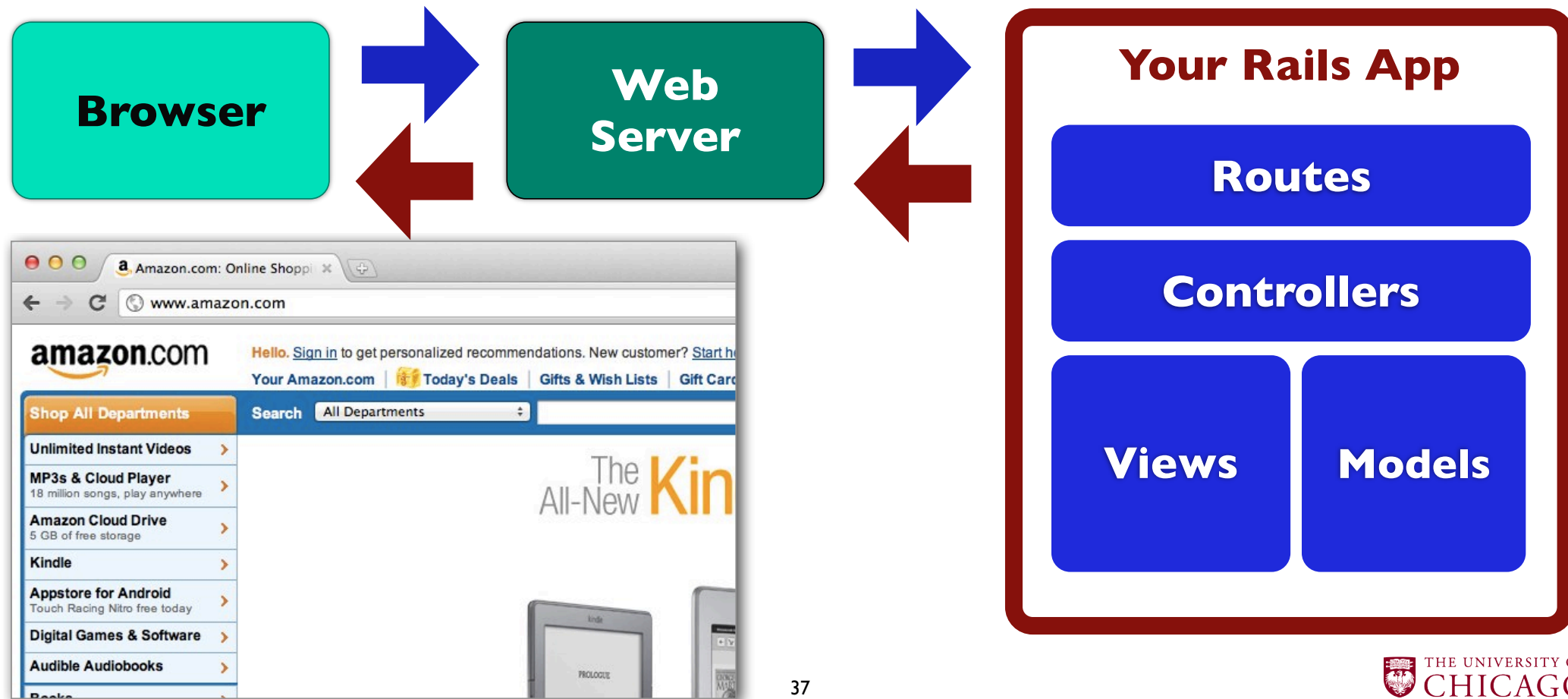
- `require 'open-uri'`
- `require 'json'`
- `f = open("http://cspp52553.com/scrabble/hello.json")`
- `data = f.read`
- Try to display the number of scrabble points

HTTP

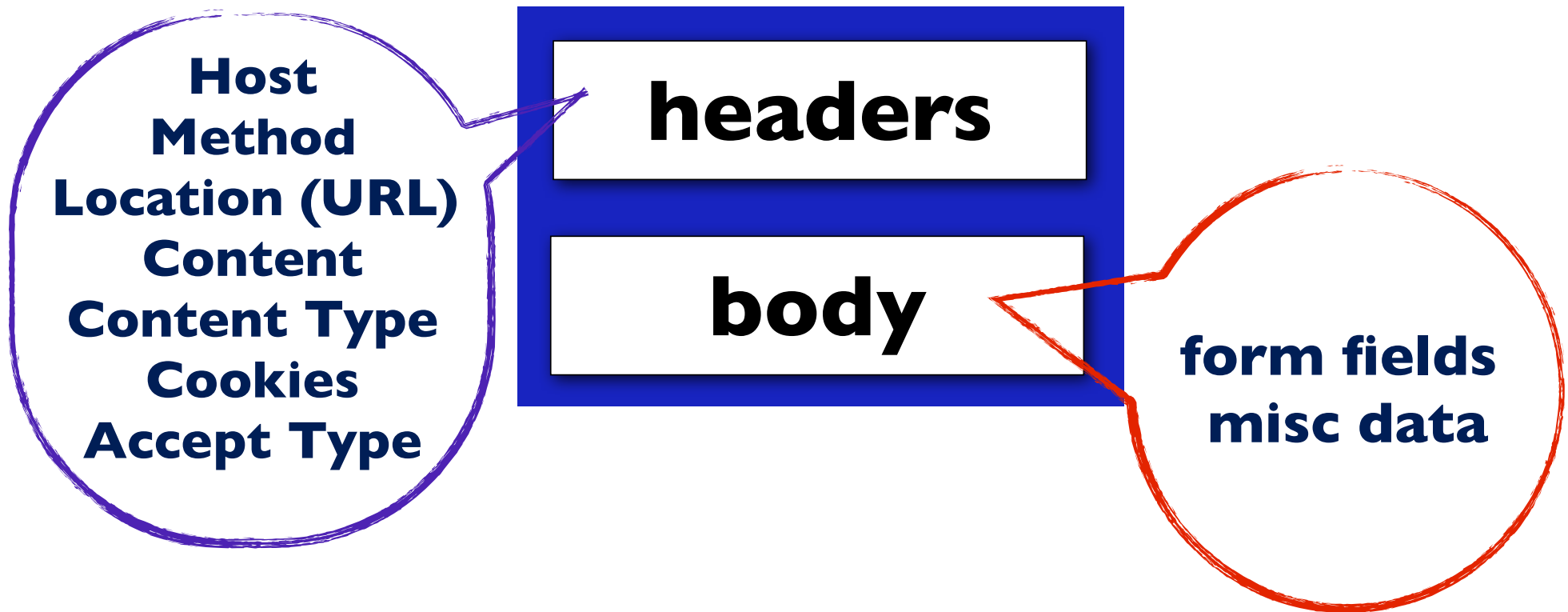
HTTP Overview



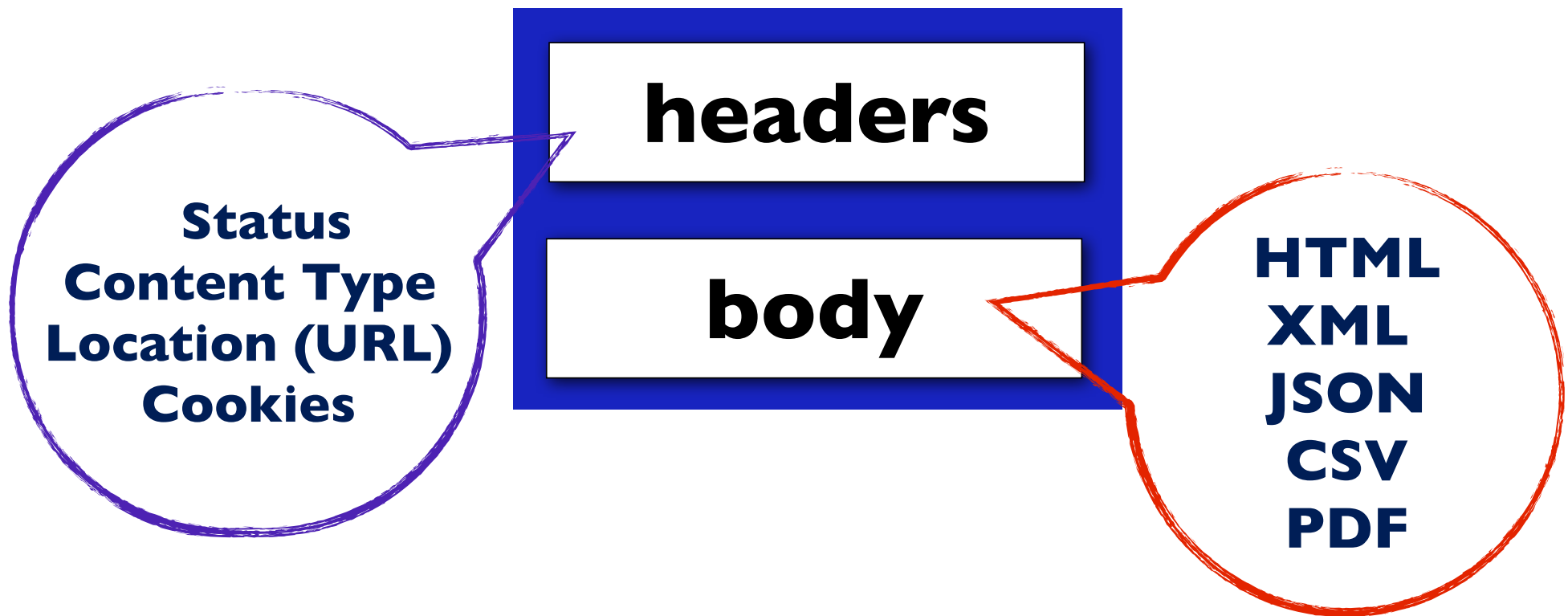
HTTP Overview



HTTP Request



HTTP Response



HTTP Playground

Playing is the best way to learn!

- ❖ **curl**
- ❖ **Chrome Inspector**
- ❖ ***search for "http sniffer"***

Ruby on Rails



Why Rails?

**Database-Backed
Web Applications**

**Convention Over
Configuration**

Agile Development

Why Rails?

STTCPW

YAGNI

DRY

SRP

Why Rails?

Responsibility

Rails QuickStart

1. `cd ~/dev/uc/week2`
2. `rails new myapp`
3. `cd myapp`
4. `rails server`
5. Go to `http://localhost:3000`

Domain Specific Languages

omg lol i can haz ur speech

Rails: Routes



Rails 101: Routes

Basic Definition

**A route connects
a URL
to a specific web page
in your app.**

Rails 101: Routes

A Better Definition

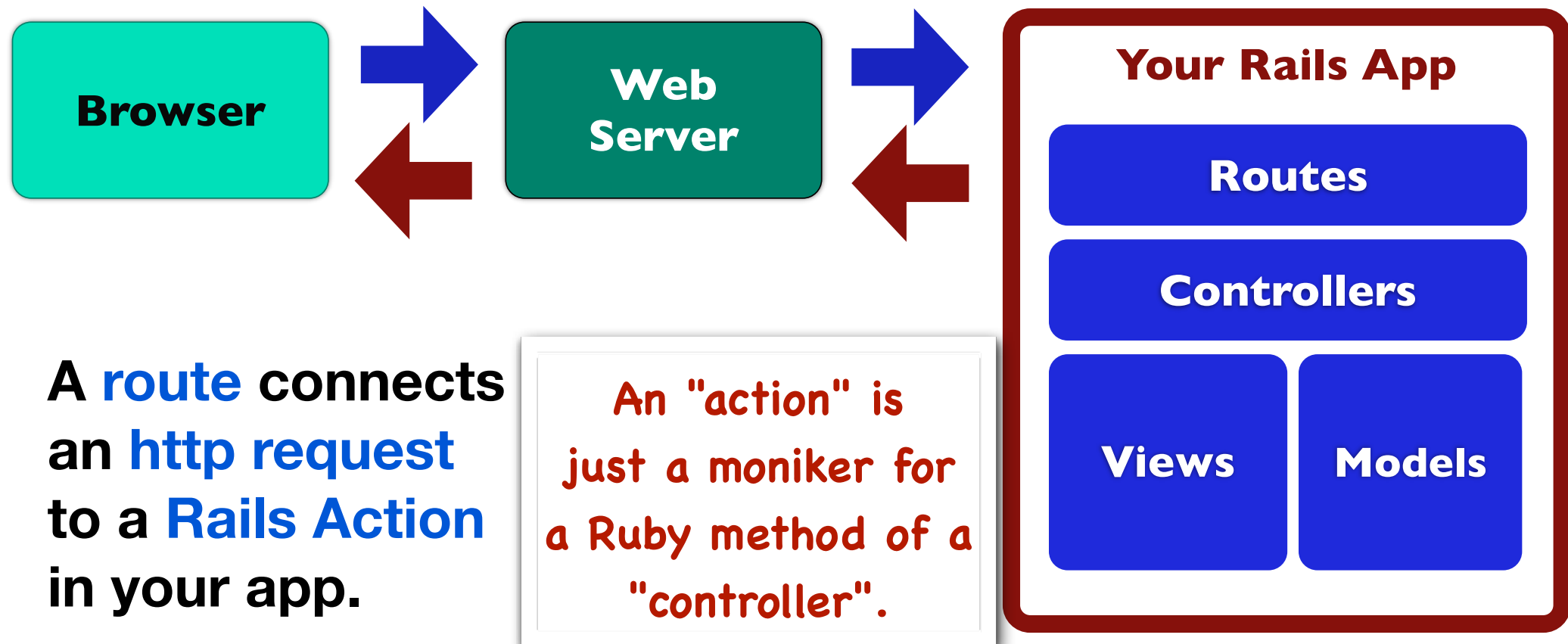
**A route connects
an http request
to a specific Ruby method
in your app.**

Rails 101: Routes

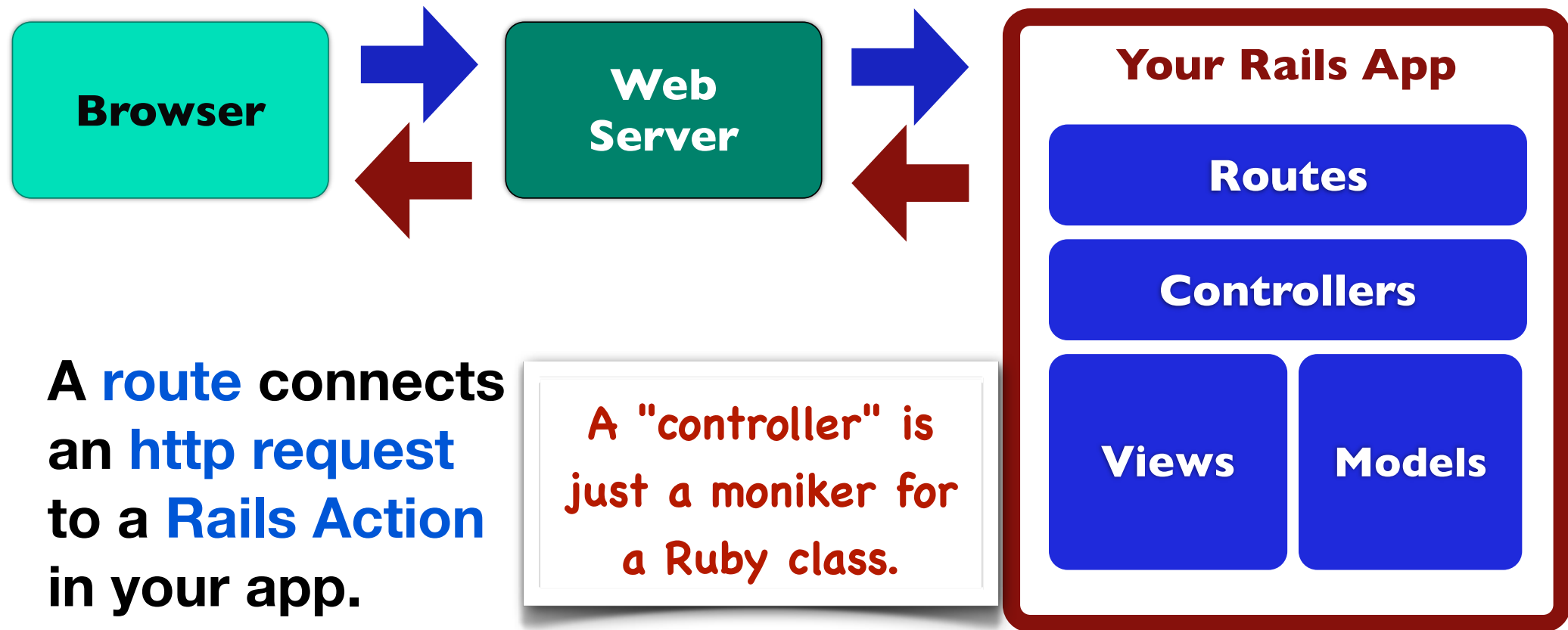
Best Definition

**A route connects
an http request
to a Rails Action
in your app.**

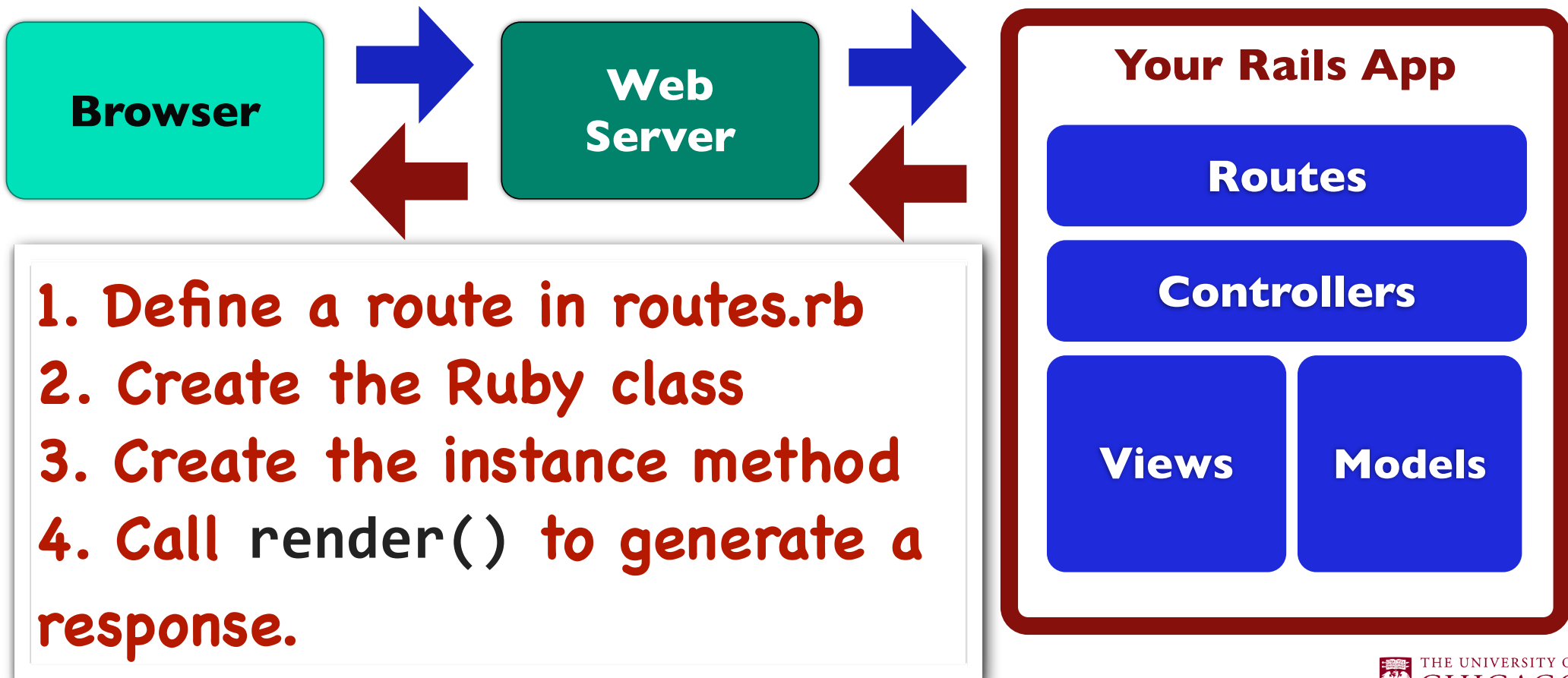
Rails 101: Routes



Rails 101: Routes



Rails 101: Routes



Rails: Actions

HTTP Response Headers

HEADER	MEANING
Content-Type	Type of data
Location	Resource location
Status	Response code
Set-Cookie	Cookie data

HTTP Response Headers

An **action** generates a response by using the **render** method or an **HTML view template**.

Rendering in HTTP Response

```
def greeting  
  render(:text => "Hello!", :status => 200)  
end
```

Generates an HTTP response with a body and header status code.

Rendering in HTTP Response

```
def greeting
  render(:text => "Hello!",
        :status => 301,
        :location => "http://www.apple.com")
end
```

Generates an HTTP response with a body, header status code, and header location value.

Rails Views

Rendering a view template

```
def greeting  
  render 'greeting'  
end
```

Generates an HTTP response by using a *view template* named *greeting.html.erb*

Rendering conventions

```
def greeting  
  render  
end
```

The filename can be omitted if it's the same as the name of the action method.

Rendering conventions

```
def greeting  
end
```

The render statement can be omitted if you want to render a view with the same name as the action method.

Demo + Lab

Goal: Create a web page of favorite things.

Use an HTML unordered list.

The URL should be:

<http://localhost:3000/favorites>

Data-Driven Views

Instance variables in action methods can be used inside the view template for that action!

```
def greeting  
  @salutation = "Wazzzzup!"  
end
```

Data-Driven Views

We use embedded Ruby to insert logic and expression evaluation:

```
<h1><%= @salutation %></h1>
```

Demo + Lab

Goal: Convert the view into a data template.

Create an array of items in the controller.

Then use ERb to *generate* an HTML unordered list.

The RCAV Recipe

1. Define a **route**
2. Create a **controller** class
3. Create an **action** method
4. Create a **view** template

Demo + Lab

Goal: Create a web page of your favorite photo.

Use the `` tag.

The URL should be:

`http://localhost:3000/photo`

Demo + Lab

Goal: Use the *image_tag* view helper.

Demo + Lab

Goal: Add links between each page.

Use `<a>` tags.

Demo + Lab

Goal: Use the *link_to* view helper.

Rails Routes: Named Routes

Naming a route makes it possible to do a reverse-lookup, and allows us to answer the question:

Given a route, what's the URL?

User Input

URL Query String Parameters

/photos?photo_id=5

URL Query String Parameters

/photos?photo_id=5

URL Query String Parameters

`/photos?photo_id=5`

`/photos?photo_id=5&zoom=10`

URL Query String Parameters

`/photos?photo_id=5`

`/photos?photo_id=5&zoom=10`

Use the **params** hash

Demo + Lab

Goal: Enhance the web page for one favorite to display the favorite given by the ID in the querystring:

`/favorites?id=3`