

- ❖ **Assets and the Rails Asset Pipeline**
- ❖ **Route Segments**
- ❖ **HTML Forms**
- ❖ **HTTP Methods + Routes**
- ❖ **Rails Models, Part 1**
- ❖ **RESTful Routes: Think in Resources!**

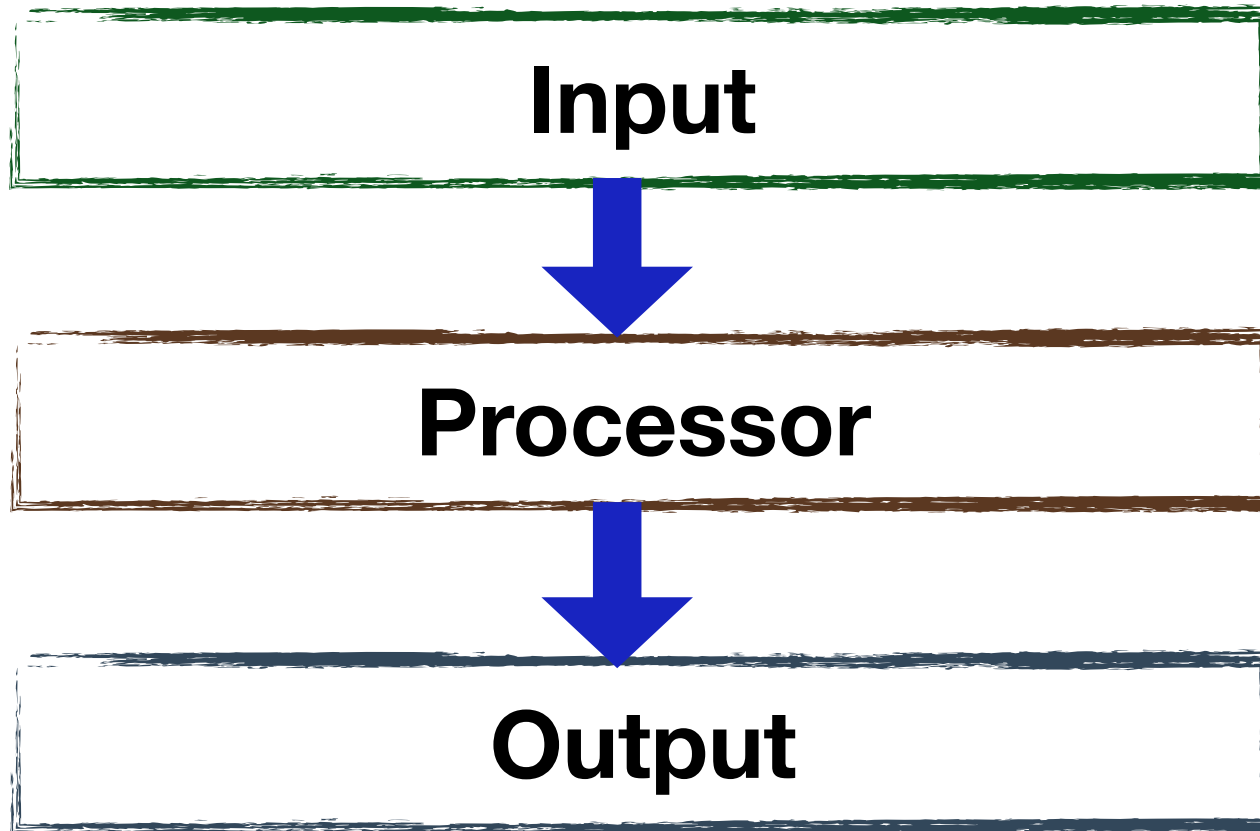
Key Ideas

Agile Principles

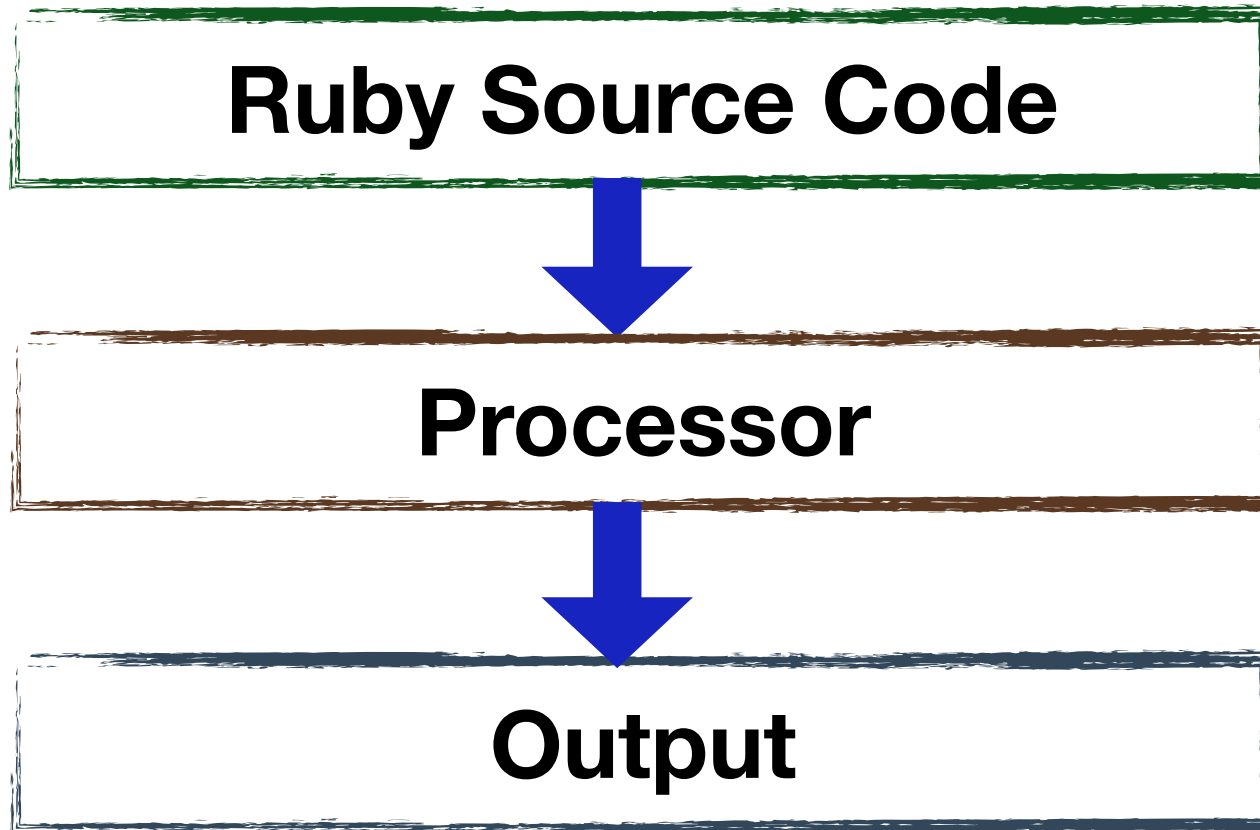
Single Responsibility

Clear Intention

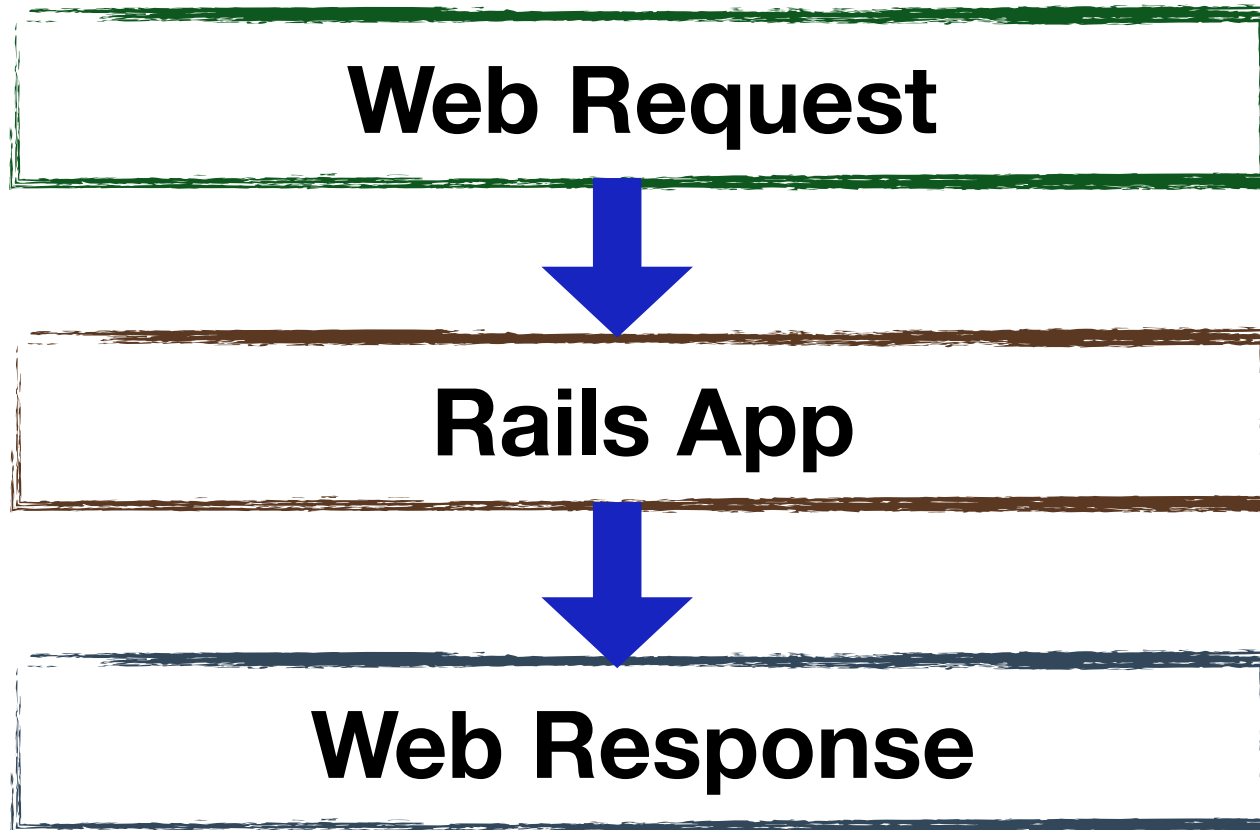
IPO



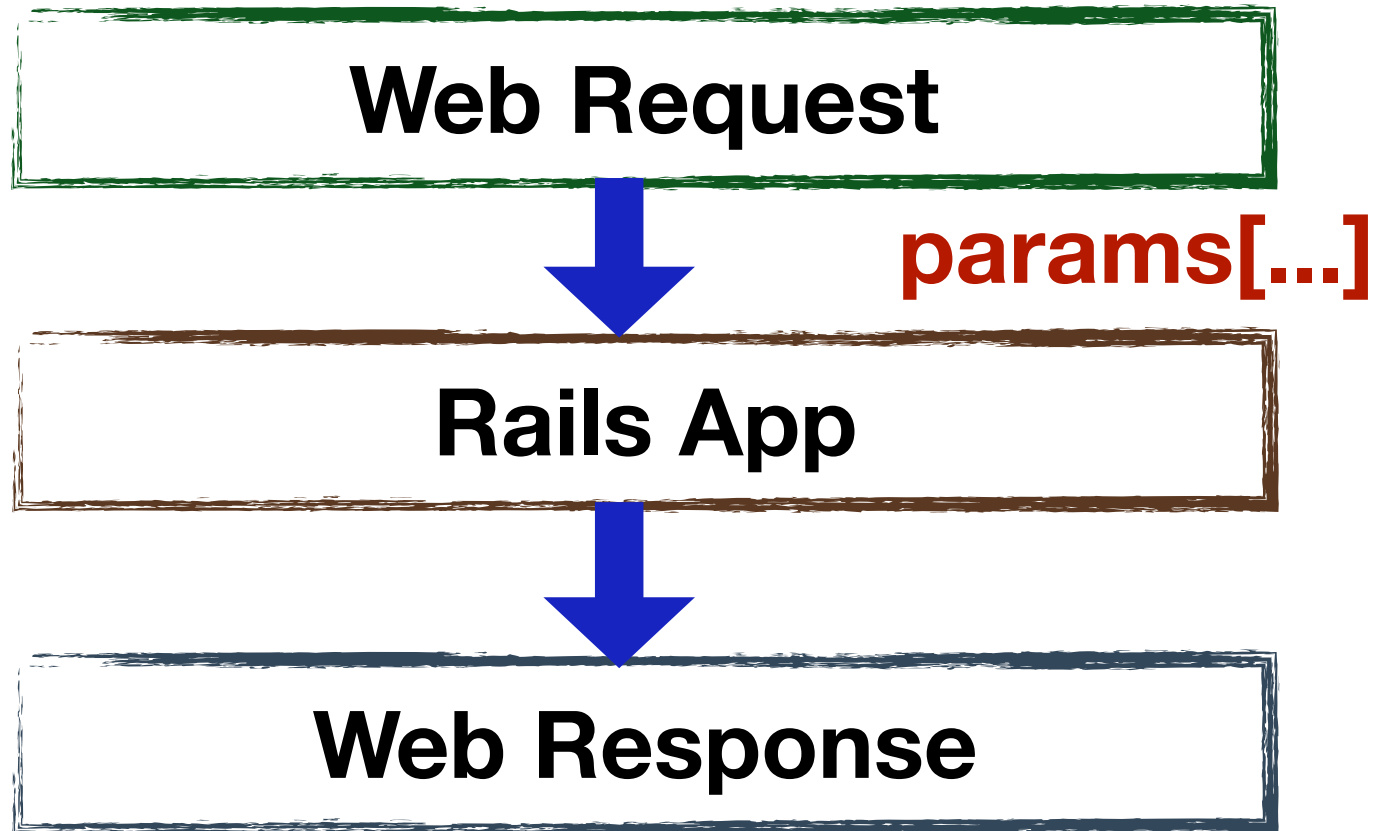
IPO



IPO



IPO



Clone week3

```
cd ~/dev/uc  
git clone git@github.com:cspp52553/week3.git  
cd week3/code/MoviesApp  
bundle install
```

**See the list of movies.
Notice the horrible appearance.**

Assets

Assets are files in your application that are static, can be cached for the lifetime of the application, and might be amenable to compression.

- **Images**
- **Javascript**
- **CSS Stylesheets**

Rails Routes: Dynamic Segments

How can we see the details for a particular movie?

Rails Routes: Dynamic Segments

http://localhost:3000/photos/5

```
get '/photos/:id',  
    :controller => 'products',  
    :action => 'show'
```

```
params # => { "id" => "5" }
```

Lab

`http://localhost:3000/movies/1`

Params hash

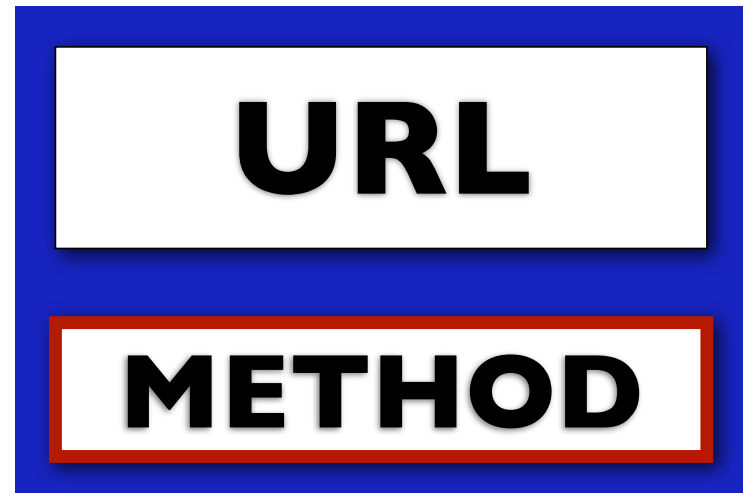
strings or symbols

HTTP Methods & Routes

Demo + Lab

As a user, I want to delete a movie from the list.

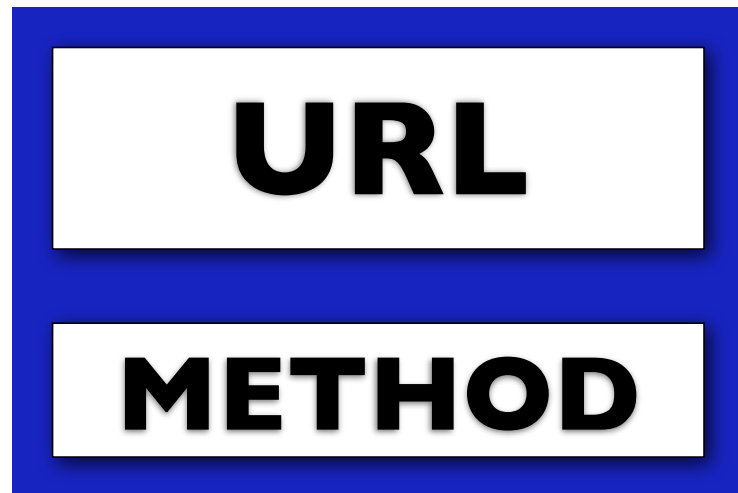
HTTP REQUEST



GET POST PUT DELETE

Routes

get `'/pictures/:id'` => `'Pictures#show'`
post `'/pictures/'` => `"Pictures#create"`



Routes

get `'/pictures/:id'` => `'Pictures#show'`
delete `'/pictures/:id'` => `"Pictures#destroy"`

/pictures/42

GET

Routes

get `'/pictures/:id'` => `'Pictures#show'`
delete `'/pictures/:id'` => `"Pictures#destroy"`

/pictures/42

DELETE

Demo + Lab

As a user, I want to delete a movie from the list.

Forms

Forms

HTML Forms are containers of *input elements*.



New Photo

Title

Image URL

Create Photo

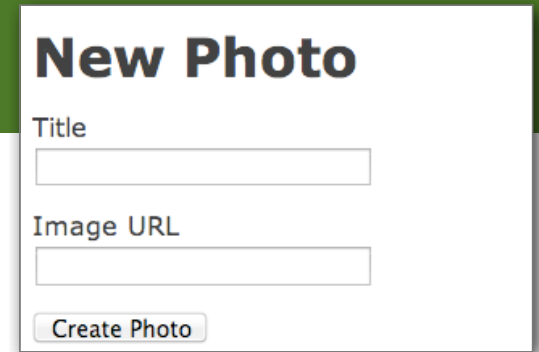
Each input element is given a name.

The user provides the value.

When the form is submitted, the browser sends the name/value pairs in the message body of the HTTP request.

Forms

HTML Forms are containers of *input elements*.



New Photo

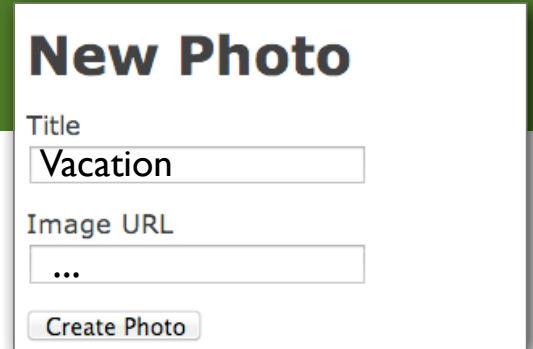
Title

Image URL

```
<form action="/photos" method="post">
  <label for="photo_title">Title</label><br />
  <input id="photo_title" name="photo[title]" size="30" type="text" />
  <label for="photo_image_url">Image URL</label><br />
  <input id="photo_image_url" name="photo[image_url]" size="30"
type="text" />
  <input name="commit" type="submit" value="Create Photo" />
</form>
```

Forms

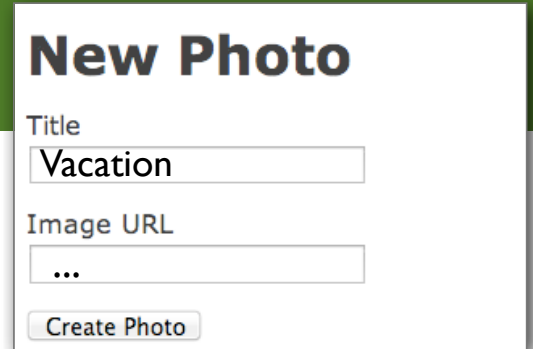
HTML Forms are containers of *input elements*.



```
<form action="/photos" method="post">
  <label for="photo_title">Title</label><br />
  <input id="photo_title" name="photo[title]" size="30" type="text" />
  <label for="photo_image_url">Image URL</label><br />
  <input id="photo_image_url" name="photo[image_url]" size="30"
type="text" />
  <input name="commit" type="submit" value="Create Photo" />
</form>
```

```
params # => { "photo" => { "title" => "Vacation", "image_url" => "..." } }
```


Demo + Lab



A screenshot of a web form titled "New Photo". It contains two input fields: "Title" with the text "Vacation" and "Image URL" with three dots "...". Below the fields is a button labeled "Create Photo".

New Photo

Title
Vacation

Image URL
...

Create Photo

<http://localhost:3000/pictures/new>

View Helpers

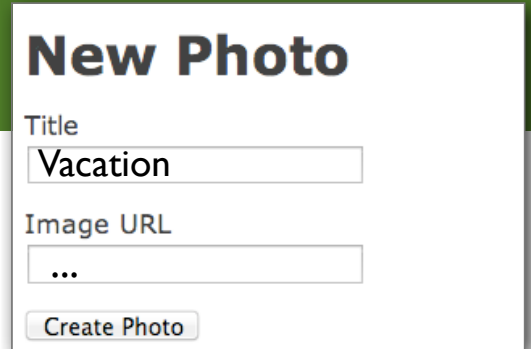
`http://localhost:3000/pictures/new`

`form_tag`

`label_tag`

`text_field_tag`

`submit_tag`

A screenshot of a web form titled "New Photo". It contains two text input fields: "Title" with the value "Vacation" and "Image URL" with the value "...". Below the fields is a "Create Photo" button.

New Photo

Title

Image URL

Models

Models are software objects that represent real-world objects.

We use Ruby classes to define models.

Models

Models go in the *models* folder.

app/models

Naming Conventions

app/models/landmark.rb

```
class Landmark  
end
```

Naming Conventions

app/models/orange_tree.rb

class OrangeTree
end

Database-Backed Models

Movie

```
class Movie < ActiveRecord::Base
  # ...
end
```

id	title	year
1	Apollo 13	2004
2	Star Wars	1977
3	Lincoln	2012

Database-Backed Models

rails generate **model** Movie title:string director:string year:integer

Generator

Model name

Attributes

pictures		
<i>id</i>	<i>url</i>	<i>note</i>
1	http://www.nhl.com/ice.jpg	Rink dimensions
2	http://cnn.com/news.png	Today's headlines

Database-Backed Models

new
save
create

update_attribute(...)
update_attributes(...)

destroy

find_by_id

all
first
last

count

Database-Backed Models

new
save
create

update_attribute(...)
update_attributes(...)

destroy

find_by_id

all
first
last

count

CREATE. READ. UPDATE. DELETE.