**RUBY ON RAILS NOTES**

-RUBY is an interpreted language its more OOP.

-RUBY is a web development framework.

Matsumoto wanted a language more powerful than PEARL and more OOP than python.

-No compile and build step to create a runnable program in RUBY.

-Everything from an integer to a string is considered to be an object

You do not need to designate MAIN.

You do not need to predeclare variables in RUBY.

# - Comment single line top or at end of a statement.

gets() reads in a string. () <----- optional.

== equal to

!= not equal to

Puts – spits out each argument on a new line all by itself.

Print displays.

Print does not add this new line.

Print all its arguments in a single line.

“\n” and tabs “\t”

Method (that is, a way) for an object to respond to a message.

In OOP you send a message to an object asking it to do something.

In a not so “OOP” would be to say that…say something is like a function with is bound to an object and can be called by using DOT notation.

Each object has built in methods

To use a method you need to put a dot after the object

“Hello world”.upcase

RUBY data types:

-integers

-floats

-string

You may optionally put one or more arguments separated by commas after a method name.

Read user input we’ll use **gets.**

gets, puts, and print referred to as methods.

elsif presents another conditional test while else is a fail-safe if none of the conditions are met.

Functions lower case – this\_long\_name

def if a function doesn’t return anything it can be called a procedure.

1. State the name of your function.
2. Any parameters it needs inside of paratheses.
3. Code that makes up the body
4. Close the end

Call the function by name (to enclose) the arguments.

Functions can use default values.

Raising an exception provides an alternate way to return form a function or procedure in a way that designates that some kind of failure occurred.

**Require\_relative** - “tests” is used to call script from another ruby file.

Arrays & iterators

Arrays store a set of values as a group.

Arrays can have mixed data types in RUBY.

ai[0] returns 1st at index 0

ai[3] returns 4th at index 3

y = %w(this is an array of strings).

.empty returns true if no elements.

.size

Stack – LIFO: Last in first out.

Natural tendency to reverse the order of data

Queue – FIFO: First in First out.

A Class is like a blueprint from which individual objects are constructed.

ob = MyClass.new

\*methods – “do stuff”

\*methods remove redundency

\*parameter & arguments – are the same thing.

Rails philosophy

-Don’t repeat yourself

-convention over configuration

Controller’s purpose is to receive specific requests for the applications.

Routing decides which controller receives which request. There is more than one route to each controller. Different routes serviced by different actions.

A view displays info controller not view where info is collected.

View templates written in eRUBY (embedded ruby)

To create a new controller – controller “Welcome”

With an action called index

Making an application for web requires a lot of pieces and parts

-Application framework

-a database system

-a neb service

- an object relational mapping library

RAILS contains all this software.

ActionPack – Generate data driven, iterative webpages a database system.

Abstract layer – supports a number of different database options – Sqlite3

A web server for rails PUMA

An object relational mapping library

Active record – makes the data look like a collection of simple RUBY objects.

Will load environment file

Boot file loads up your GEM file then runs

Bundler to setup those GEMS

In Application file it runs that boot file

Initializer

Database.yml – different environments – setup your database configurations for these develop/test/production

Environment file initializes your application file.

Routes are different URL locations in your applications.

App is where your code goes, views are layouts of your pages.

The term scaffolding refers to generated database code and webpages needed to perform CRUD(Create, read, update, delete)

Generate scaffold student

When rails generate the scaffolding it also generated a script called a migration

Which is used to create the appropriate table in the database.

Model code is connected to the data our application is managing. The model code manages how data will get read and written to the database. The model code uses objects to represent things in the problem domain of our application.

Codes are connected this way

View – controller – model

Rails wants views to talk to the controller to access the model information they need.

web 1– rails router 2 – 8 controller – 7 view – 6 controller 3 – 5 model 4 – database

database – model – controller – web

View code allows our application to render student info on a web page for the end user through their web browser

.erb file extension for embedded RUBY

How to update the model object and associated view pages.

<% %> tags alle you to write arbitrary ruby statements that get interpreted silently when erb is run.

<%=%> write an Ruby expression (typically an object value).

<%= tag will wind up with output you can see displayed in the browser = is only on the opening tag <%= %>

Form.label generates the HTML for a form label form.text.find generates the HTML for a entry field the symbol :firstname – easier to use them as string.

Rails generate scaffold show

Name: string

Seasons: integer

Network: string

Lead: string

Stars: integer

Controller – the real brains behind an application

Controllers – are used to send info to a view

Controllers are configured by routes

RECAP

We created a new resource called articles.

Needed to generate a controller articles

Need to add new methods to the article controller

Def new/ imdex/show/create

Needed to create a form so users can enter new articles and also a safe path to store info.

Need to create an article

Need to create an article model

Need to make a migrate and then Rake it to the database

Save data to in the controller

Show articles in the controller

Show list pf all articles in the controller

Set controller to access in views.

Rails generate scaffold

Rake db:migrate

Rails server -e development –binding =127.0.01

Rails console ---- exit

Database layer in rails supports Boolean, date, time, float, integer, string, and text.

Person.create (iname => “Barak Obama”, iage => 52, ialive => true)

The statement above uses symbols and hash arrows to initialize the data members of a person and save to the database.

INTRO CONTROLLERS

We need code that can extract data out of the database and provide it to a view this is the rule.

Model objects are singular

Controller objects are plural.

Controller are a class in RUBY.

Even though we added person model object code to show.html.erb

Nothing appear cause there is no route

We have the model data, a view and controller we don’t have is a rule for the controller to find the correct view

Routes are used to tell rails where your web pages is

Rails uses routes to map URL path to code

Get “persons/:id” => ‘persons#show”

Person 1 to the show.erb with the call.show

**Whenever a change is made to the routing file you always need to rake the route.**

Then update the controller to provide a show method.

-Next step will be to get the controller to pass a person model object (ie Barak Obama) to the view webpage .

-Next step have our controller get data out of the database and pass it to our view.

-We need to update the show method in the controller to call .find on the Person class.

-next update the view to the dynamic page which works with the passed model object

\*we’ve created the data object, views, controller, and have connected the controller and view through routes tied data from the passed object to the view.

So far we have connected the Person controller to a view with a route passing a person model data object.

<%= aperson.name%>

..take data at this location and add it to the view.

-create an index page with links to show the page.

To create an index page we will need to send the view not just a single object, but a whole array of objects each one display a link

This index method is creating an array of model data which will get stored in the global variable a persons for use inside the view web page.

Get “persons” = 7 “perons#index”

This new rule is mapping the URL /persons to the file index.erb..with a call to the index method of the Person controller.

Command to show your supported routes - **Rake Routes**

Index,html.erb – goal of this view is to create a hyperlink for each person model object the controller extracted from the database

INSERT NEW ROUTES

Instead of a controller send a person to a page. The controller needs to rend the submitted form fields, build an object from then and save it to the database.

This view will provide a web form for data entry purpose new.html.erb

Post ‘persons/new”=>’ persons#create

This added rule connect the web form submission to the person controller. Inside .create we need to build a Person object from the passed form filed and then save it to the database.

Deleting

Create a delete route

Get ‘persons/:id/delete=>”person#destroy

This added rule connects the link to the Person controller .destroy. inside destroy we need to select the person object from passed :id parameter and then delete that record.

Patter view .erb template + routes.rb+ rate routes+ controller operations.

**Building a search form**

Create a web form for users to search and connect that submission to a controller

New controller method .find.

This form doesn’t have any supporting model data objects that’s why we don’t see form\_for instead form\_tying\*

To say just once and have all the client workout pages use this form as part of their layout so we use layout in view.

It allows us to make a template that incorporates the info held inside their layout.

To .find method is creating an array of model data which will get stored in the global variable it will allow us to search either by client name or by trainer.

Next new routing map is needed mps the URL/find to the file find.html.erb with a call to the .find method