

More Sinatra & Database Theory

**Any homework
questions?**

Sinatra Review

- First, you must require 'rubygems', then require 'sinatra'
- Routes are written using:

```
get '/hi' do
  <route actions>
end
```
- Parameters are written using colons

```
get '/hi/:name' do
```
- Their values are retrieved using the params hash

```
params[:name]
```

HAML

- Stands for “HTML abstraction markup language”
- “Markup should be beautiful”
- Used in place of the default rails templating language, ERB

Using HAML

- Add the HAML gem to your Gemfile, then bundle
- Require HAML in your project

Writing HAML

- HAML files end in .haml
- They should be placed in a subdirectory called views
- To render them in Sinatra, type
“haml :file_name”

A HAML Primer

Doctype

- To specify doctype in HTML5, this tag is typically used:
`<!DOCTYPE>`
- To do so in HAML, just type:
`!!!`

Tags

- Denoted using a % sign
 - %h1, %p, %div
- Any text afterwards will be inside the tag
 - %h1 This is the after text
- Any text indented one line underneath will be inside the tag
 - %h1
This is inside the H1 tag

Attributes

- HTML attributes are added using a Hash
 - `%a{:href => "http://google.com"}`
- Hash values can be Ruby variable names
 - `%a{:href => website}`

Ruby code

- To use ruby code in your HAML files:
- `=` to evaluate, and insert
 - `=@user.name`
- `-` to evaluate, but not insert
 - `-if @user.name.present?`
`@user.name`

Classes and IDs

- There is an easier notation for classes and IDs
- Classes can be indicated with a dot
 - `%strong.firsttitle`
- IDs with a `#`
 - `%strong#firsttitle`

Instance Variables

- What if you wanted to display data from a param inside of your HAML view?
- Instance variables are specified inside each of your routes, using an @ sign
- Can be accessed in HAML views with @ sign
 - @user = “Zach”

Comments

- Use the `/` to generate an HTML comment:
 - `/ This is a comment`
- Use the `-#` to generate a HAML comment -
 - won't render in HTML document
 - You can put lines underneath it that also won't be reflected.

String interpolation

- Ruby string interpolation can be done in plain text:
 - %p Check it out, #{@user.name}!

Layouts

- Using a layout.haml file provides a skeleton for your pages
- layout.haml is the header and footer of your page
- Use = yield to ensure your template is rendered in the layout

Use it or lose it

Create two routes that render HAML views, and a layout. Use instance variables, divs, classes, and attributes.

Database Theory

What is a database?

- A collection of organized data - typically in tables
- Each table is meant to model real-life objects
- A DBMS or database management system is an interface to deal with the data itself

Database types

- Oracle
- Sybase
- MySQL
- PostgreSQL
- SQLite

Ruby + Databases

- Ruby uses a variety of databases
 - <3s:
 - PostgreSQL
 - SQLite
- Ruby uses a special system called an ORM to interact with the database

What's an ORM?

- Stands for Object-relational mapping
- Create a “virtual object database”
- Makes interacting with database objects easy

ActiveRecord

- ActiveRecord is an ORM
- You could use DataMapper - but ActiveRecord is standard with Rails
- It makes using a database fun!

Understanding Relationships

- Database tables are related to each other in certain ways
- There are many ways they can be related to each other

One to one

- A row in a table is associated with only one row in another table
- A user has one spouse (hopefully)

One to many

- A row in a table can be associated with one or more rows in another table
- A hotel can have many rooms, but a room can only belong to one hotel

Many to many

- One or more rows in a table are associated with one or more rows in another table
- Customers can purchase many products, products can be purchased by many customers
- Through - done through a “join” table

Exercises

- Draw out the tables, columns, and relationships for the following scenarios:
 - Online store
 - Facebook-like website
 - Forum