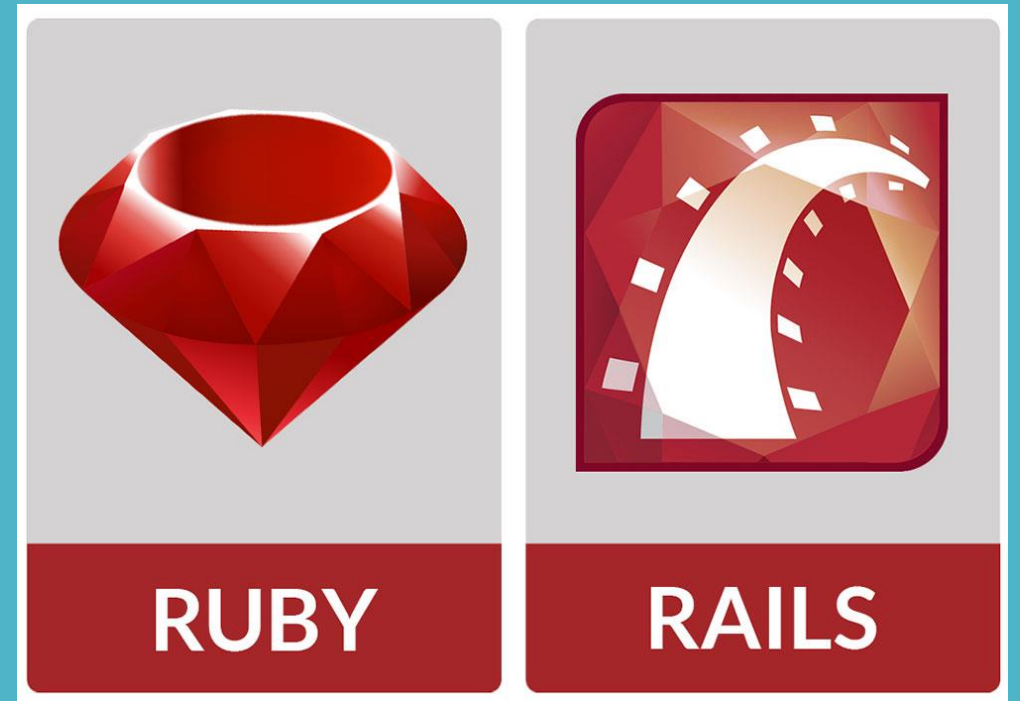


Ruby (Rails)

Austin Liu

ECE 458 – Language Bootcamp



Background on Ruby

- Created in 1995 by Yukihiro Matsumoto
- Multi-paradigm programming language
 - i. Procedural
 - ii. Functional (anonymous functions, closures, continuations)
 - iii. Introspection, reflection, metaprogramming
- Semantics: Designed for coder flexibility
 - i. Object-oriented (*Everything* is an object!)
 - ii. Blocks/Closures
 - iii. Exception handling
- Developed mostly on GNU/Linux
 - i. Works on many types of UNIX, Mac OS X, and Windows

Background on Rails

- A server-side web-app framework that utilizes the Ruby language
- Uses the Model-View-Controller (MVC) pattern to organize application
 - **Model:** models in the RoR framework maps to a table in a database **as well as** a Ruby file.
 - **View:** *erb* file that is converted to HTML at run-time.
 - **Controller:** RESTful routes mapped by *routes.rb*
- Programming Designs: “CoC”, “DRY”, “Fat models, skinny controllers”

```
<% page_title = "Demonstration of ERB" %>
<% salutation = "Dear programmer," %>

<html>

  <head>
    <title><%= page_title %></title>
  </head>

  <body>
    <p><%= salutation %></p>
    <p>This is an example of how ERB fills out a template.</p>
  </body>

</html>
```



```
<html>

  <head>
    <title>Demonstration of ERb</title>
  </head>

  <body>
    <p>Dear programmer,</p>
    <p>This is an example of how ERb fills out a template.</p>
  </body>

</html>
```

Benefits of Ruby on Rails

- **Extremely productive** web-application framework
- **ActiveRecord** link to database uses Object-Relational Mapping and is powerful
- **Built-in testing** makes life a lot easier with harnesses and fixtures
- **Automatic distinction of environments** – development, testing, production
- **Good foundations** force you to be a better developer as a side-effect of the framework being so productive
- **Lots of gems** – libraries for everything! A large community and great tooling

From playing around with Ruby for a couple minutes with scaffolding, you can link up an existing database, implement some simple CRUD operations on the data model, and display the data in the view in a couple of minutes!

Drawbacks of Ruby on Rails

- **Runtime Speed** – Compared to NodeJS, the performance of a Ruby application tends to be slower, but is rarely an issue for smaller-scaled applications
- **Documentation** – Languages such as Python, Java, and PHP are better documented
- **Dependence on ActiveRecord** – The design pattern forces your application to be reliant on ActiveRecord to interface with your persistent database
- **A bit of a learning curve** for people who have not been introduced to the Ruby language or web development
- **Not a silver bullet** – for intermediate/larger projects, need to understand more about the underpinnings and plugins and gems and a lot of other stuff... Lots of learning