Ruby on Rails

Week 1- Intro and Basics

What do we want to cover?

- What is Ruby?
- Intro to unit testing in Ruby
- What is Rails?
- Hello World in Rails
- Code Quality

Ruby

General purpose object oriented language

Interpreted

Runs in an interpreter called ruby

Libraries are called gems, can be managed using a tool called bundler

- rubygems.org

Comes with a REPL called irb

Ruby

Everything is an object (even nil)

Parenthesis are (mostly) optional

Has notation for arrays, hashes, regex, and symbols

Uses metaprogramming and dynamic invocation

Relies on Convention over Configuration

Ruby - Naming

Classes - camel case

Variables - snake case

Methods - snake case

Constants - upper case

Files - snake case

Ruby - variables

4 scopes

- Instance - @x
- Class - @@x
- Local - x
- Global - $x (use under punishment of death)

Ruby - Special Method Names

Ending with ‘?’

- Example: hot?
- Meaning: The method is expected to evaluate to a boolean

Ends with ‘=’

- Example: color = ‘red’
- Meaning: assignment

Ends with ‘!’

- Example: sort!
- Meaning: Performs the action in-place

Ruby - Strings

Literal

- Single quotes - ‘hello John’

Interpolated

- Double quotes - “hello #{@name}”

Symbols

- Colon - :name
- Kind of like a string literal converted to a constant
- The thing named “name”

Ruby - Arrays and Hashes

Array (a.k.a List)

- [1,2,3]
- Array.new
- a[4]

Hash (a.k.a Dictionary, Map, Associative Array)

- {:x => 7, :y => 8, :z => 9}
- Hash.new
- h[:x]

Ruby - Blocks

```
customers.each { |c| puts c.name }

customers.each do |c|

    puts c.name

end
```

Ruby - Exception Handling

```
begin
    puts 1/0
rescue Exception => c
    # do something about it
end
```

Ruby - Organization

Classes are primary, like most OO languages

Modules are ways to create namespaces or group methods together

- Note: Cannot instantiate a module

Classes can be nested in other classes or in modules

Folder structure conforms to module and class nesting structure

Ruby - Accessing Other Ruby Code

‘require’ at top of file to reference other code or gems

‘include’ will add another module or class to your class as instance methods

‘extend’ will add another module or class to your class as static methods

RSpec

Unit testing framework in Ruby

DSL on top of Ruby to describe the behaviors of objects

```
describe ‘some object’ do

     it ‘behaves in some way’ do

          #arrange

          #act

          #assert

     end

end
```

Rails

Web application framework in Ruby

Uses and emphasizes

- MVC
- Fat models, skinny controllers
- Convention over configuration
- Scaffolding
- Active record pattern

Rails - Examples

Hello World

Hello World with some interaction

Simple Calculator

Rails - Hello World

```
ruby bin\rails new HelloWorld

ruby bin\rails server
```

Rails - Hello World with Input

```
ruby bin\rails new HelloWorldWithInput

ruby bin\rails generate controller Say who hello goodbye
```

Code Quality

Getting those Rails tests running

Code Smells

SOLID

Ruby gems

- Reek - looks for smells
- Excellent - looks for structural problems
- Flay - checks for duplication