



Ruby (on Rails)

CSE 190M, Spring 2009

Week 3

Web Programming in Ruby

- Ruby can be used to write dynamic web pages
- Similar to PHP, chunks of Ruby begins with "<%" and ends with "%>"
- Ruby code goes between these tags
- Our web pages will have file extensions of .erb
- Just like PHP, we cannot view our .erb files from our hard drive; upload it and view it from webster

erb

- We have been using the Ruby interpreter for our pure Ruby programs
- For web pages, we need to use a Ruby parser that knows how to deal with embedded Ruby in an HTML file, just like PHP
- This detail does not really matter...
- ...But for completeness, we are using erb/eruby
- Thanks to Morgan, this is already setup and configured on webster, so you do not need to worry about it

erb syntax

- Code Blocks
- Equivalent to `<?php ... ?>` in PHP

`<%`

ruby statements

`%>`

- Printing expression values
- Equivalent to `<? = ... ?>` in PHP

`<%= expression %>`

99 Bottles of Beer

- Write an embedded Ruby file for the 99 Bottles of Beer example from lecture
- It should generate the HTML for 99 Bottles of Beer

<https://webster.cs.washington.edu/rctucker/99bottles.erb>

99 Bottles of Beer

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head><title>CSE 190 M: Embedded Ruby</title></head>
  <body>
    <%
      num = 99
      while num > 0
    %>
      <p> <%= num %> bottles of beer on the wall, <br />
        <%= num %> bottles of beer. <br />
        Take one down, pass it around, <br />
        <%= num - 1 %> bottles of beer on the wall. </p>
    <%
      num = num - 1
    end
  %>
</body>
</html>
```

CGI

- CGI: Common Gateway Interface
- "CGI Programming" is a fancy way of saying that we will be writing server side programs that will produce output as a response to a visitors request
- Ruby has a cgi library to make web programming in Ruby easier

require

- We want to use functionality from Ruby's cgi library
- In Java, we would use the keyword "import"
`import java.io.*`
- To do this in Ruby, we use "require"
- If we wanted to use the Ruby CGI library:
`require 'cgi'`

Query Parameters

- To get query parameters, we must create a CGI object
- Then, we can access the query parameters of a web request through the associative array, with the parameters as the keys
- Example

```
require 'cgi'           # requires the cgi library
cgi = CGI.new
name = cgi['name']     # example.erb?name=Ryan
```

Query Parameters

- Accessing the values of parameters returns them as Strings
- If you want to use them as other values, you need to tell them to behave that way

```
num = cgi['quantity'].to_i
```

- You can also check if a parameter is set

```
cgi.has_key?('name')           # equivalent to PHP's isset()
```

99 Bottles of Beer (revisited)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head><title>CSE 190 M: Embedded PHP</title></head>
  <body>
    <%
      require 'cgi'
      cgi = CGI.new

      if cgi.has_key?('bottles')
        num = cgi['bottles'].to_i
      else
        num = 99
      end
    %>
  </body>
</html>
```

99 Bottles of Beer (revisited)(cont.)

```
    while num > 0
%>
<p> <%= num %> bottles of beer on the wall, <br />
<%= num %> bottles of beer. <br />
    Take one down, pass it around, <br />
    <%= num - 1 %> bottles of beer on the wall. </p>

<%
    num = num - 1
end
%>
</body>
</html>
```

Query Params and Check Boxes

- Accessing query parameters like a hash on our CGI object works fine for typical inputs where there is only one value per key
- But, if we try to get all of the values of a set of checkboxes, only the first value is returned
- To get all the values, we will use the CGI `params()` method

Query Params and Check Boxes

- The CGI params() method returns a hash with the query parameters as the keys of the hash
- The value of each key is an array of the values of the parameter
- For parameters with one value, this still returns an array of size one
- For checkboxes, we are able to get all of the values instead of just the first one

form.html

```
<form action="params.erb">
  <label><input type="checkbox" name="reason" value="ruby" />
    You learn Ruby!</label><br />
  <label><input type="checkbox" name="reason" value="rails" />
    You will learn Rails!</label><br />
  <label><input type="checkbox" name="reason" value="not_early" />
    It is not early in the morning!</label><br />
  <label><input type="checkbox" name="reason" value="hw_optional" />
    Homework is optional!</label><br />
  <input type="submit" />
</form>
```

params.erb

```
<%  
  require 'cgi'  
  cgi = CGI.new  
%>
```

```
<h1> Using the params() method </h1>  
<% cgi.params.each do |key, value| %>  
<%= "#{key}: #{value}" %> <br />  
<% end %>
```

```
<h1> Not using the params() method</h1>  
<% cgi.keys.each do |param| %>  
<%= "#{param}: #{cgi[param]}" <br />" %>  
<% end %>
```

Debugging in erb

- It is not easy...
- Most errors, especially syntactical errors, result in a "HTTP Error 500: Internal Server Error"
- It says to email Morgan Doocy... but don't, it is just an error in the script
- Debugging will be easier when we start to use Rails where Rails provides better built-in error handling