

# JMS 302: HACKING THE MEDIA

---

CLASS 13: 9/30/2014

## TODAY'S TASKS



- Review Quiz
- Hashes, <<EOF
- Exercise

# QUIZ REVIEW



# HASHES

- Similar to arrays, but we name each value rather than just the collection—i.e., we use a string instead of an index to refer to each element.

- `hash_name[key] = value`

- Example:

```
student_grades = {} # or Hash.new
student_grades['Joe'] = 94
student_grades['Jill'] = 100
student_grades.each do |name, grade|
  puts "#{name} earned a grade of #{grade}."
  # or puts name + " earned a grade of " + grade + "."
end
# same as last 3 lines
student_grades.keys do |name|
  puts "#{name} earned a grade of #{student_grades[name]}."
  # or puts name + " earned a grade of " + student_grades[name] + "."
end
puts student_grades
```

C

See pages 97–99 in Pine.

Actually any object can serve as the key, but a string is most common.

## <<EOF

- For big blocks of strings, use <<EOF or <<-EOF. This saves you from line after line of + or f.write.
- Example:

```
my_string = <<-EOF
  hello
  this is a new paragraph
  i can include #{variables} with interpolation.
EOF
puts my_string
```

B

- EOF stands for “end of file.”
- EOF must be on a line by itself. Without the -, it must be at the very beginning of the line.
- EOF is more common than -EOF, but the - means you can keep your indentation of EOF.
- You can use any string, not just EOF, but you want one not likely to occur in your other text.
- The string you use must be on a line by itself, at the beginning of the line.
- Note that there is no space between << and EOF
- Don’t use quotes around EOF. [You can use quotes around it after <<, and even there are special options available ([http://www.tutorialspoint.com/ruby/ruby\\_syntax.htm](http://www.tutorialspoint.com/ruby/ruby_syntax.htm)). But just don’t.]
- If this were all there were to the code, I don’t need the variable. I could just do `puts <<-EOF ....`

## EXERCISE

---

# Interactive CSS Zen Garden

# INTERACTIVE CSS ZEN

- We're going to write a ruby script that asks the user specific questions to determine the look of the page and then write that out to a css file. Use methods to do discrete tasks.
- Start simple. Work backwards.
- Ask for: background color, link color, underline links
- Extra credit: Make the first character of the preamble a drop-cap.

- Make sure you have the index.html file from [csszengarden.com](http://csszengarden.com). Look at the CSS file for ideas. Copy or rename to save original.
- edit html to point to root css
- File name. Hash. write\_file.
- ask
- format css, view html
- copy original css, modify background
  - <http://www.csszengarden.com/tr/en/001/> (basis for sample css)
  - base url: <http://www.csszengarden.com/001/> — we can put this in the css for anything with a url value
- add questions
- Bonus: look for preamble, add special css. You'll need float (left) to make this look right.