JMS 302: HACKING THE MEDIA

CLASS 11: 9/23/2014



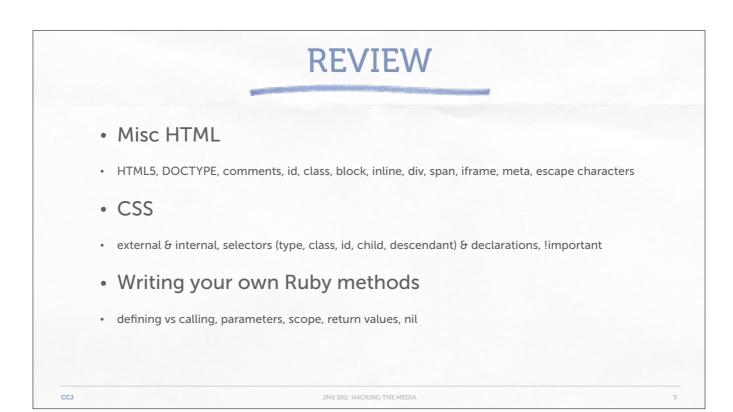


- Reading and Writing Files with Ruby
- CSS Colors, Text, [Images; Embedding with HTML]

CCJ

JMS 302: HACKING THE MEDIA

2



You've learned more methods, and technically this list is more than 18 methods. Some you don't need to memorize or use often (center, ljust). Some on this list are actually multiple methods (+ on a number vs + on a string).

WRITING FILES

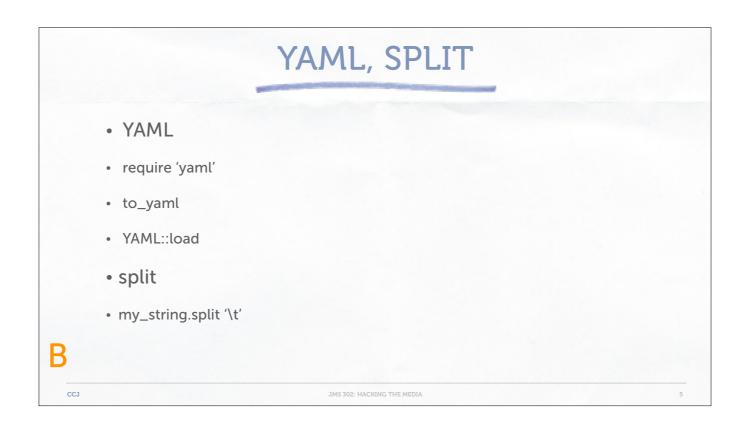
- B Command-line trick: >
- File
 - File.open('test.txt', 'w') { |f| f.write "test" }
 - my_contents = File.read 'test.txt'.
- B Dir

CCJ

• Dir['**/*.{JPG,jpg}'], Dir.chdir '/Users/jms340'

JMS 302: HACKING THE MEDIA

 $ls > my_files_and_folders.txt$



to_yaml gives you a YAML-formatted string load takes a YAML-formatted string and converts it back into the original form

DOUBLE-QUOTED STRINGS

- Special characters
- \n, \t, \s
- Interpolation
- variables (and expressions)
- puts "my string is #{ my_string } and the return value from my_method is #{ my_method }"



CCJ

JMS 302: HACKING THE MEDIA

6

CSS COLORS

- foreground, background
- C rgb, hex, named colors
- B rgba, hsla

CCJ

• alpha: opacity/transparency. 0 = fully transparent, 1 = fully opaque

JMS 302: HACKING THE MEDIA

CSS TEXT

- Typeface (serif, sans-serif, monospace)
- Weight (light, medium, bold, black)
- Style (normal, italic, oblique)
- Stretch (condensed, regular, extended)

CCJ

JMS 302: HACKING THE MEDIA

8

CSS TEXT

- font-family, font-size (px, %, em), font-weight, fontstyle, text-align, vertical-align, :link, :visited, :hover
- text-transform, text-decoration, text-indent, textshadow
- line-height, font-face, letter-spacing, word-spacing, :first-letter, :first-line, :active, :focus

JMS 302: HACKING THE MEDIA

- B: Know key font-family groupings for the key typefaces:
- Georgia, Times, serif
- Helvetica, Arial, Verdana, sans-serif
- "Courier New", Courier, monospace

C: Size: just know a way of specifying. Don't expect you to convert.

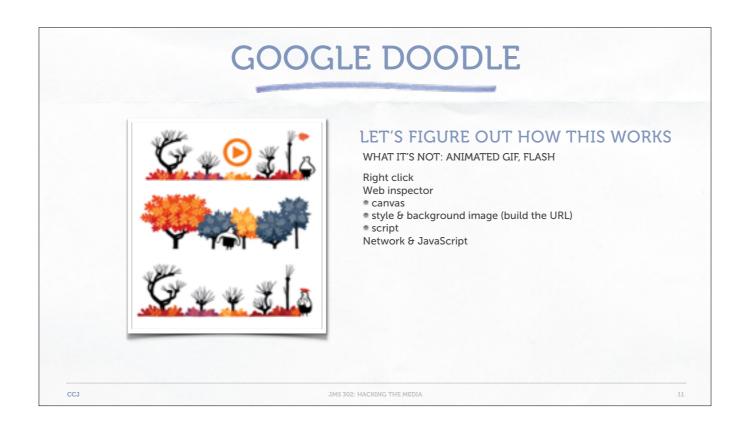
CCJ

MORE SELECTORS

SELECTOR	HEANING	EXAMPLE
EXISTENCE	(ahatow a specific attribute (ahatow its value)	p(class) Targets any 157 element with an attribute called class
EGUALITY	(=) Matches a specific attribute with a specific value	p[class="dog"] Targets any 1g2 element with an attribute called class whose value is dog
SPACE	[] Matches a specific attribute whose value appears in a space- separated fall of words	p(class—"dog") Targets any cip1 element with an attribute called class whose value is a first of space-separated words, one of which is dog
PREFIX	[1-] Matches a specific attribute whose value begins with a specific string	p(atter***) Targets any 4p1 element with an attribute whose value begins with the letter "0"
SUBSTRING	[14] Matches a specific attribute whose value contains a specific substring	p[attr="60"] Taggits any cp1 element with an attribute whose value contains the letters "60"
SUPPOX	[1-] Matches a specific attribute whose value ends with a specific string	p[att+E*g*] Targets any (g) element with an attribute whose value ends with the letter "g"

JMS 302: HACKING THE MEDIA

CCJ



Archive version: http://www.google.com/doodles/first-day-of-fall-2014