Ruby Explorations VI

Mark Keane...CSI...UCD

De Basics V

Part V: A Fist Full of Basics

- A: equality, comparisons and modules
- B: class security & stuff
- C: regular expressions & string hacking
- D: more than you every wanted know about gems
- E: getting out of the house...

Part A:

Equality, comparisons and modules

String Equality

nb , **=**
and **==**

will cause bugs

REM: LECT4

```
>> str_a = "foobar"
=> "foobar"
>> str_b = "FooBar"
=> "FooBar"
>> str_a == str_b
=> false
>> str_a == str_a
=> true
>> str_a.eql?(str_b)
=> false
>> str_a != str_b
=> true
```

Array Equality

REM: LECT4

```
>> arr1 = ["foo”,”bar"]
=> ["foo”,”bar"]
>> arr2 = ["foo”,”bim"]
=> ["foo”,”bim"]
>> arr3 = ["foo”,”bar"]
=> ["foo”,”bar"]
>> arr4 = [“bar”, “foo”]
>> arr1 == arr2
=> false
>> arr1 != arr2
=> true
>> arr1 == arr3
=> true
>> arr1 == arr4
=> false
>> arr1.eql?(arr3)
=> true
```

http://www.ruby-doc.org/core/classes/Array.html

Steps in *method* lookup*

message = “hello”; message.show_it

it searches the **String** class of **“hello”** for an instance method named **show_it**; if no method is found in the class then...

it searches the **Comparable** and **Enumerable** modules of the **String** class for **show_it** (backwards); if no method is found then...

the search moves up the hierarchy to the superclass **Object** but it has no **show_it** method either

so, it searches the **Kernel** module included in **Object** but no luck there so it finds the **method_missing** method in **Kernel** and does it

REM: LECT6

* ignoring eigenclasses

Recall, Modules and Mixins

module is degenerate class, no instances but methods

we can introduce it to a class; **include Pred**

it can actually do more, if you define one specialization of a module's methods, it can be propagated to others

so, you define one method and get 10 for free

Enumerable is a module that provides **sort**, **count**, **each**, **map**, **zip**...to arrays and other classes

Comparable is a module that provides **<, <=**, **==**, **>**, **<=, <=>** and **between?**

Comparable Operators

a < b returns **true** if **a** is less than **b**

a > b returns **true** if **a** is greater than **b**

a <=> b returns
-**1** if **a** is less than **b**,
0 if **a** is equal to **b**
1 if **a** is greater than **b**

```
def ascii(stringy)
 stringy.each_byte {|c| puts c}
end

str_a = "foobar"
str_b = "foobas"

ascii(str_a)
puts "\n"
ascii(str_b)
puts "\n"

p str_a == str_a
p str_a == str_b

p str_a < str_b
p str_a > str_b
p str_b < str_a

p str_a <=> str_b
p str_a <=> str_a
p str_b <=> str_a
```

compare.rb

Comparable Operators

```
$ ruby compare.rb
102
111
111
98
97
114

102
111
111
98
97
115
```

```
def ascii(stringy)
 stringy.each_byte {|c| puts c}
end

str_a = "foobar"
str_b = "foobas"

ascii(str_a)
puts "\n"
ascii(str_b)
...
```

compare.rb

Comparable Operators

a < b returns **true** if **a** is less than **b**

a > b returns **true** if **a** is greater than **b**

a <=> b returns
-**1** if **a** is less than **b**,
0 if **a** is equal to **b**
1 if **a** is greater than **b**

```
def ascii(stringy)
 stringy.each_byte {|c| puts c}
end

str_a = "foobar"
str_b = "foobas"

ascii(str_a)
puts "\n"
ascii(str_b)

p str_a == str_a
p str_a == str_b

p str_a < str_b
p str_a > str_b
p str_b < str_a

p str_a <=> str_b
p str_a <=> str_a
p str_b <=> str_a
```

```
$ ruby compare.rb
true
false
true
false
false
-1
0
1
```

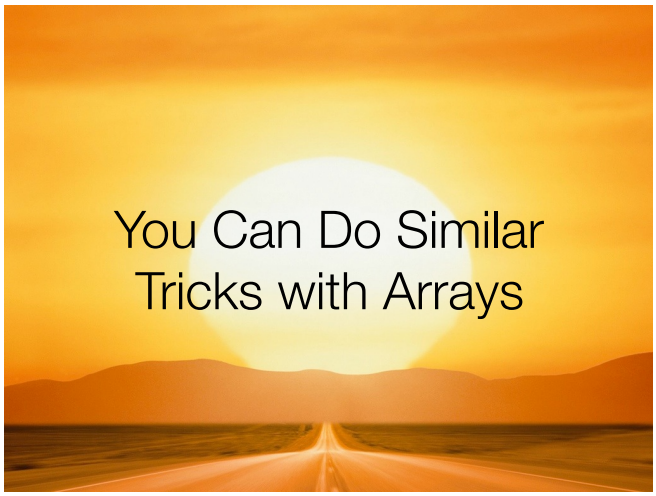
compare.rb

Comparable Operators

a < b returns **true** if **a** is less than **b**

a > b returns **true** if **a** is greater than **b**

a <=> b returns
-**1** if **a** is less than **b**,
0 if **a** is equal to **b**
1 if **a** is greater than **b**



Recall, Modules and Mixins

module is degenerate class, no instances but methods

we can introduce it to a class; **include Pred**

it can actually do more, if you define one specialization of a module's methods, it can be propagated to others

so, you define one method and get 10 for free

Enumerable is a module that provides **sort**, **count**, **each**, **map**, **zip**...to arrays and other classes

Comparable is a module that provides **<, <=**, **==**, **>**, **<=, <=>** and **between?**

Consider Song

```
class Song
  attr_accessor :name, :album, :artist, :time
  def initialize(name, album, artist, time)
 @name = name
 @album = album
 @time = time
 @artist = artist
  end

  def to_s
    puts "<< #{@name} >> by #{@artist} in their album #{@album}.\n"
  end
end
foo = Song.new("Bring in the Clowns","Krusty's Greatest","Krusty", 4.3)
bar1 = Song.new("Bring in the Clowns","Krusty's Greatest","Krusty", 4.3)
bar2 = Song.new("Bring in the Clowns","Liza's Greatest","Minelli", 4)

p foo == foo
p foo == bar1
p foo <=> bar1
p bar2 < bar1
```

song_again.rb

Consider Song

```
class Song
  attr_accessor :name, :album, :artist, :time
  def initialize(name, album, artist, time)
 @name = name
 @album = album
 @time = time
 @artist = artist
  end

  def to_s
    puts "<< #{@name} >> by #{@artist} in their album #{@album}.\n"
  end
end
foo = Song.new("Bring in the Clowns","Krusty's Greatest","Krusty", 4.3)
bar1 = Song.new("Bring in the Clowns","Krusty's Greatest","Krusty", 4.3)
bar2 = Song.new("Bring in the Clowns","Liza's Greatest","Minelli", 4)

p foo == foo
p foo == bar1
p foo <=> bar1
p bar2 < bar1
```

```
$ ruby song_again.rb
true
false
nil
undefined method `<' for #<Song:0x27498> (NoMethodError)
```

song_again.rb

Consider Song

```
class Song
  include Comparable
  attr_accessor :name, :album, :artist, :time
  def initialize(name, album, artist, time)
 @name = name
 @album = album
 @time = time
 @artist = artist
  end
  ...
  def <=>(song2)
     self.time <=> song2.time
  end
end
```

compare objs on song length

```
foo = Song.new("Bring in the Clowns","Krusty's Greatest","Krusty", 4.3)
bar1 = Song.new("Bring in the Clowns","Krusty's Greatest","Krusty", 4.3)
bar2 = Song.new("Bring in the Clowns","Liza's Greatest","Minelli", 4)

p foo == foo
p foo == bar1
p bar2 <=> foo
p foo <=> bar1
p foo <=> bar2
p bar2 < bar1
p bar2 > bar1
```

song_again2.rb

Consider Song

```
class Song
  include Comparable
...
end

foo = Song.new("Bring in the Clowns","Krusty's Greatest","Krusty", 4.3)
bar1 = Song.new("Bring in the Clowns","Krusty's Greatest","Krusty", 4.3)
bar2 = Song.new("Bring in the Clowns","Liza's Greatest","Minelli", 4)

p foo == foo
p foo == bar1
p bar2 <=> foo
p foo <=> bar1
p foo <=> bar2
p bar2 < bar1
p bar2 > bar1
```

```
# ruby song_again2.rb
true
true
-1
0
1
true
false
```

song_again2.rb

== works on your defined objects but ,<=> does not?

Module **Comparable**

In: compar.c

Ruby version:

The `Comparable` mixin is used by classes whose objects may be ordered. The class must define the `<=>` operator, which compares the receiver against another object, returning -1, 0, or +1 depending on whether the receiver is less than, equal to, or greater than the other object. If the other object is not comparable then the `<=>` operator should return nil. `Comparable` uses `<=>` to implement the conventional comparison operators (`<`, `<=`, `==`, `>=`, and `>`) and the method `between?`.

```
class SizeMatters
  include Comparable
  attr :str
  def <=>(anOther)
    str.size <=> anOther.str.size
  end
  def initialize(str)
    @str = str
  end
  def inspect
    @str
  end
end

s1 = SizeMatters.new("Z")
s2 = SizeMatters.new("YY")
s3 = SizeMatters.new("XXX")
s4 = SizeMatters.new("WWWW")
s5 = SizeMatters.new("VVVVV")

s1 < s2                       #=> true
s4.between?(s1, s3)           #=> false
s4.between?(s3, s5)           #=> true
[ s3, s2, s5, s4, s1 ].sort   #=> [Z, YY, XXX, WWWW, VVVVV]
```

Methods

`<` `<=` `==` `>` `>=` between?

Part B:

Class security and stuff

Privacy in Classes I

OOP succeeds by breaking things into manageable portions and using typing to handle things...

Lisp is the wild west by comparison

in the same spirit, Ruby provides good security around the use/abuse of methods

methods within a class can be **public**, **protected**, **private** (mean slightly different things in Ruby)

Privacy in Classes II

public methods can be invoked from anywhere, no access control is enforced; all methods are public by default (except **initialize** which is always private)

private methods are internal to the implementation of a class, can only be invoked by other instance methods of the class (or its sub-classes); always invoked in *functional style*, not on ***object*** *or* ***self***

protected methods are like private in that they are internal to the class, but can be invoked on **self** (but it is more complicated than this; see p.233 F&M)

Privacy in Classes II

public methods can be called by anyone -- no access control is enforced; all methods are public by default (except **initialize** which is always private)

except “global” methods defined outside classes, like at the top: **REM**

Privacy in Classes II

public methods can be invoked from anywhere, no access control is enforced; all methods are public by default (except **initialize** which is always private)

private methods are internal to the implementation of a class, can only be invoked by other instance methods of the class (or its sub-classes); always invoked in *functional style*, not on ***object*** *or* ***self***

protected methods are like private in that they are internal to the class, but can be invoked on **self** (but it is more complicated than this; se p.233 F&M)

Privacy Eg.1a

```
class Testo
 def initialize
     @foo = "heeeee"
 end

 def santa
     @foo + "  Hooooo"
 end

 def wicked_elf
     @foo + @foo + @foo
 end
 def snooty
     "I do not laugh"
 end
end

test = Testo.new
test.initialize
#privacy.rb:21: private method `initialize' called for #<Testo:0x27f38
#@foo="heeeee"> (NoMethodError)

p test.santa
p test.wicked_elf
p test.snooty
```

$ ruby privacy.rb
"heeeee Hooooo"
"heeeeeheeeeeheeeee"
"I do not laugh"

privacy.rb

Privacy Eg.1b

```
class Testo
 def initialize
     @foo = "heeeee"
 end

 def santa
     @foo + "  Hooooo"
 end

 def wicked_elf
     @foo + @foo + @foo
 end
 def snooty
     "I do not laugh"
 end
    public :santa
    protected :wicked_elf
    private :snooty
end
test = Testo.new
p test.santa
p test.wicked_elf
p test.snooty
```

$ ruby privacy.rb
=> "heeeee Hooooo"
=> protected method `wicked_elf' called for #<Testo:0x27aec @foo="heeeee"> (NoMethodError)
=> private method `snooty' called for #<Testo: 0x27b64 @foo="heeeee"> (NoMethodError)

privacy.rb

Privacy Eg.2a

```
lass Testo
 def initialize
     @foo = "heeeee"
 end

 def santa
     p @foo + "  Hooooo, " + snooty
     p self.wicked_elf
 end

 def wicked_elf
     @foo + @foo + @foo
 end

 def snooty
     "I do not laugh"
 end

 public :santa
 protected :wicked_elf
 private :snooty
end

test = Testo.new
test.santa
puts “**********”
test.wicked_elf
```

```
$ ruby privacy_a.rb
 "heeeee  Hooooo, I do not laugh"
 "heeeeeheeeeeheeeee"
**********
protected method `wicked_elf' called for <Testo:
0x27a10 @foo="heeeee"> (NoMethodError)
```

privacy_a.rb

Privacy Eg.2b

```
lass Testo
 def initialize
     @foo = "heeeee"
 end

 def santa
     puts "Hooooo " + self.wicked_elf
     puts self.snooty
 end

 def wicked_elf
     “I am a laughing elf”
 end

 def snooty
     "I do not laugh"
 end

 public :santa
 protected :wicked_elf
 private :snooty
end

test = Testo.new
test.santa
```

```
$ ruby privacy_b.rb
 "Hooooo I am a laughing elf"
...private method `snooty' called for #<Testo:
0x11a594> (NoMethodError)
```

privacy_b.rb

Top-level Methods

REM

...are instance methods of **Object** (but self is not **Object**, self is **main**)

...are always private (don't ask...)

why ?

since they are methods of Object they can (in theory) be used with any object

since they are private they must be invoked like functions with no explicit receiver

Top-level Methods

REM

...are instance methods of **Object** (but self is not **Object**, self is **main**)

...are always private (don't ask...)

why ?

since they are methods of Object they can (in theory) be used with any object

since they are private they must be invoked like functions with no explicit receiver

This is a sort of compromise between having functions and having OOP

Part C:

regular expressions and string hacking

Regular Expressions

...are used to test for text patterns in strings...

shown as **/foo/** where the slashes are like holders you grab as you pass the **foo** pattern along the string...

```
>> name = "sdkfhksdjg
dsjdsflkjfooasdjshadkjh"
=> "sdkfhksdjg
dsjdsflkjfooasdjshadkjh"
>> name =~ /foo/
=> 20
>> reg1 = /foo/
=> /foo/
>> reg1
=> /foo/
>> reg1.class
=> Regexp
>> fi = Regexp.new("ksd")
=> /ksd/
>> fi
=> /ksd/
>> name =~ fi
5
```

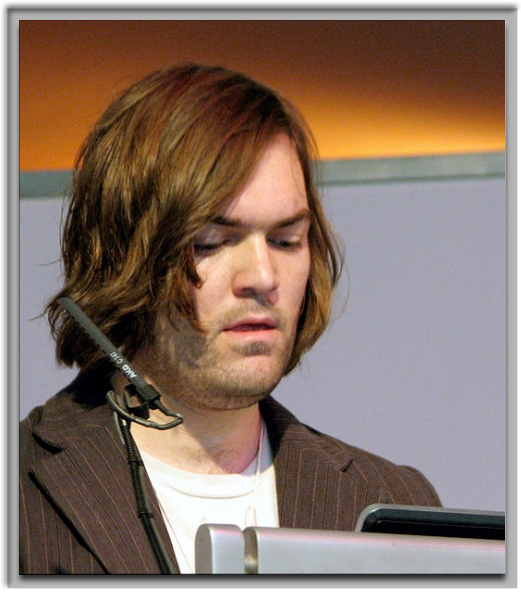
homage to *_why*...celebrate *whyday*

_why

_why aka Jack Black wrote "...(poignant) Ruby"

did Hpricot library

disappears (?) Aug 19 2009



http://en.wikipedia.org/wiki/Why_the_lucky_stiff

sub for substitution

```
>> tale = "The boy stood on the burning deck, saying oh boy..."
=> "The boy stood on the burning deck, saying oh boy..."
>> tale.sub(/boy/, "girl")
=> "The girl stood on the burning deck, saying oh boy..."
>> tale
=> "The boy stood on the burning deck, saying oh boy..."
>> tale.sub(/\sb/, " c")
=> "The coy stood on the burning deck, saying oh boy..."
>> tale.sub(/\sb/, "c")
=> "Thecoy stood on the burning deck, saying oh boy..."
>>
```

gsub for substitution

```
>> tale = "The boy stood on the burning deck, saying oh boy..."
=> "The boy stood on the burning deck, saying oh boy..."

>> tale.gsub(/boy/,"girl")
=> "The girl stood on the burning deck, saying oh girl..."

>> tale.gsub(/\sb/," c")
=> "The coy stood on the curning deck, saying oh coy..."

>> tale.gsub(/\sb/, "c")
=> "Thecoy stood on thecurning deck, saying ohcoy..."

>> tale.gsub(/.o./,"sos")
=> "The sos ssosdsos the burning deck, sayingsos sos..."

>> website = "/www.manic_repressives.com/home/denial/who.html"
=> "/www.manic_repressives.com/home/denial/who.html"

>> website.gsub(/\//,"-")
=> "-www.manic_repressives.com-home-denial-who.html"
```

Part D:

More than your ever wanted to know about gems and were afraid to ask....

Gems

if you install from source then your gems list may be empty

if you install from one-click or a packaged with software they may be pre-loaded

```
$ gem list
*** LOCAL GEMS ***
...
fastercsv (1.5.4 ruby)
highline (1.6.2)
k_means (0.0.7)
...
$ gem install k_means
Successfully installed k_means-0.0.7
1 gem installed
Installing ri documentation for
k_means-0.0.7...
Updating class cache with 503 classes...
Installing RDoc documentation for
k_means-0.0.7...
```

OR

```
$ sudo gem install k_means
```

Can you all do this...

NB: if you are not ROOT then you may have to use: **sudo gem install k_means**

have you multiple versions of ruby/irb/ gem loaded

if NO good, if YES clean up your commands

```
$ gem install k_means
Successfully installed k_means-0.0.7
1 gem installed
Installing ri documentation for
k_means-0.0.7...
Updating class cache with 503 classes...
Installing RDoc documentation for
k_means-0.0.7...

$ gem list
*** LOCAL GEMS ***
fastercsv (1.5.4 ruby)
highline (1.6.2)
k_means (0.0.7)
...
```

IF **sudo** its in root directories; /usr/bin or /opt/local/bin

IF **not** it may be in home directory; /Users/markkean/.gem/ruby/2.0.0

Watch out...

if have you multiple versions of ruby/irb/gem loaded, you may get different answers for **gem1.8 list**, **gem2.0 list** and **gem list**

if NO good, if YES you probably need to clean up your commands

Handy Gem Commands

rubygems used to be a separate gem in 1.8, now comes with 2.0

in 2.0 you never **require rubygems**

when you use a gem in a file you say **require 'csv'**

```
$ gem update --system
Latest version currently installed.
Aborting.
```

```
$ gem update gems
Updating installed gems
Nothing to update
```

```
$ irb
>> Gem.path
=> ["/Users/user/.gem/ruby/2.3.0", "/
opt/local/lib/ruby2.3/gems/2.3.0"]
>> require 'csv'
=> true
>> require 'rails'
LoadError: no such file to load...
```

```
$ ruby -e 'p Gem.path'
....same as above....
$ ruby -e "p require 'k_means'"
```

rubygems.org

Problem with Gems

ruby

dir-a	gem1	gem2
dir-b	gem2	gem3
dir-c	gem5	gem6

irb

gem

Problem with Gems

ruby

dir-a	gem1	gem2
dir-b	gem2	gem3
dir-c	gem5	gem6

irb

gem

they may not be looking in the same directories...

Problem with Gems

ruby2.0
ruby1.8

dir-a	gem1	gem2
dir-b	gem2	gem3
dir-c	gem5	gem6

irb1.8
irb2.0

they may multiple versions of them...

gem2.0
gem1.8

..not looking in the same directories...

Where are my gems...

```
 $ gem env
RubyGems Environment:
  - RUBYGEMS VERSION: 2.6.6
  - RUBY VERSION: 2.3.1 (2016-04-26 patchlevel 112) [x86_64-darwin15]
  - INSTALLATION DIRECTORY: /opt/local/lib/ruby2.3/gems/2.3.0
  - USER INSTALLATION DIRECTORY: /Users/user/.gem/ruby/2.3.0
  - RUBY EXECUTABLE: /opt/local/bin/ruby2.3
  - EXECUTABLE DIRECTORY: /opt/local/bin
  - SPEC CACHE DIRECTORY: /Users/user/.gem/specs
  - SYSTEM CONFIGURATION DIRECTORY: /opt/local/etc
  - RUBYGEMS PLATFORMS:
    - ruby
    - x86_64-darwin-15
  - GEM PATHS:
     - /opt/local/lib/ruby2.3/gems/2.3.0
     - /Users/user/.gem/ruby/2.3.0
  - GEM CONFIGURATION:
     - :update_sources => true
     - :verbose => true
     - :backtrace => false
     - :bulk_threshold => 1000
  - REMOTE SOURCES:
     - https://rubygems.org/
  - SHELL PATH:
     - /opt/local/bin
```

Where are my gems...

```
 $ gem env
RubyGems Environment:
  - RUBYGEMS VERSION: 2.6.6
  - RUBY VERSION: 2.3.1 (2016-04-26 patchlevel 112) [x86_64-darwin15]
  - INSTALLATION DIRECTORY: /opt/local/lib/ruby2.3/gems/2.3.0
  - USER INSTALLATION DIRECTORY: /Users/user/.gem/ruby/2.3.0
  - RUBY EXECUTABLE: /opt/local/bin/ruby2.3
  - EXECUTABLE DIRECTORY: /opt/local/bin
  - SPEC CACHE DIRECTORY: /Users/user/.gem/specs
  - SYSTEM CONFIGURATION DIRECTORY: /opt/local/etc
  - RUBYGEMS PLATFORMS:
    - ruby
    - x86_64-darwin-15
  - GEM PATHS:
     - /opt/local/lib/ruby2.3/gems/2.3.0
     - /Users/user/.gem/ruby/2.3.0
  - GEM CONFIGURATION:
     - :update_sources => true
     - :verbose => true
     - :backtrace => false
     - :bulk_threshold => 1000
  - REMOTE SOURCES:
     - https://rubygems.org/
  - SHELL PATH:
     - /opt/local/bin
```

version of **ruby** associated with **gem**

where gems are installed

where gem searches for gems

if have you multiple versions of gem loaded this gets tricky; so, look at command cleaning

Where are my other gems...

```
% gem1.8 env
RubyGems Environment:
  - RUBYGEMS VERSION: 1.3.6
  - RUBY VERSION: 1.8.6 (2009-06-08 patchlevel 369) [universal-darwin9.0]
  - INSTALLATION DIRECTORY: /Library/Ruby/Gems/1.8
  - RUBY EXECUTABLE: /System/Library/Frameworks/Ruby.framework/Versions
                    /1.8/usr/bin/ruby
  - EXECUTABLE DIRECTORY: /usr/bin
  - RUBYGEMS PLATFORMS:
    - ruby
    - universal-darwin-9
  - GEM PATHS:
     - /Library/Ruby/Gems/1.8
     - /Users/markkean/.gem/ruby/1.8
     - /System/Library/Frameworks/Ruby.framework/Versions/1.8/usr/lib/ruby/ge
  - GEM CONFIGURATION:
     - :update_sources => true
     - :verbose => true
     - :benchmark => false
     - :backtrace => false
     - :bulk_threshold => 1000
  - REMOTE SOURCES:
     - http://rubygems.org/
```

if have you multiple versions of gem loaded this gets tricky; so, look at command cleaning

Command Cleaning...

```
$ cd /usr/bin

$ ls -la /usr/bin | grep irb
lrwxr-xr-x    1 root   irb -> irb1.8
lrwxr-xr-x    1 root   irb1.8 -> ../../System/Library/
Frameworks/Ruby.framework/Versions/Current/usr/bin/irb

$ rm irb
rm: irb: Permission denied
$ sudo rm irb


$ sudo ln -s /opt/local/bin/irb2.0 irb

$ ls -la /usr/bin | grep irb
lrwxr-xr-x    1 root   irb -> /opt/local/bin/irb2.0
lrwxr-xr-x    1 root   irb1.8 -> ../../System/Library/
Frameworks/Ruby.framework/Versions/Current/usr/bin/irb
```

Command Cleaning...

```
$ cd /usr/bin

$ ls -la /usr/bin | grep irb
lrwxr-xr-x    1 root   irb -> irb1.8
lrwxr-xr-x    1 root   irb1.8 -> ../../System/Library/
Frameworks/Ruby.framework/Versions/Current/usr/bin/irb

$ rm irb
rm: irb: Permission denied
$ sudo rm irb


$ sudo ln -s /opt/local/bin/irb2.0 irb

$ ls -la /usr/bin | grep irb
lrwxr-xr-x    1 root   irb -> /opt/local/bin/irb2.0
lrwxr-xr-x    1 root   irb1.8 -> ../../System/Library/
Frameworks/Ruby.framework/Versions/Current/usr/bin/irb
```

do all this for **irb**, **gem** and **ruby**

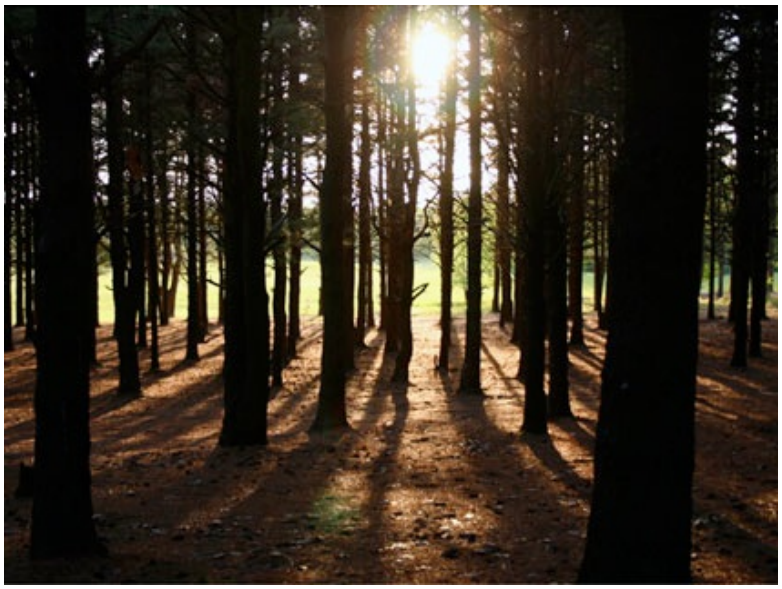
Command Cleaning...

if you don't then you could be using **irb1.8** to test for a gem loaded by **gem2.0** before using a program run with **ruby1.8**

oh...and did I say they all may be searching in different places

do all this for **irb**, **gem** and **ruby**

We are still in them...



What's this !

we have cleaned our commands

we have loaded a gem to the right place (**k_means**)

then, we look for it with **require**

```
$ gem env
- INSTALLATION DIRECTORY:
/opt/local/lib/ruby/gems/2.0.0
...

$ gem list
*** LOCAL GEMS ***
k_means (0.0.7)
fastercsv (1.5.4 ruby)
...

$ irb
>> require 'k_means'
LoadError: no such file to load -- k_means
 from (irb):1:in `require'
 from (irb):1
 from /usr/bin/irb:12:in `<main>'

>> require 'fastercsv'
=> true
```

What's this !

```
$ gem env
- INSTALLATION DIRECTORY:
/opt/local/lib/ruby/gems/1.9.1
```

```
$ irb
>> Gem.path
=> ["/opt/local/lib/ruby1.9/gems/1.9.1"]

$LOAD_PATH.each {|pth| p pth}
"/opt/local/lib/ruby1.9/gems/1.9.1/gems/fastercsv-1.5.4/lib"
"/opt/local/lib/ruby1.9/gems/1.9.1/gems/rubygems-update-1.3.7/bin"
"/opt/local/lib/ruby1.9/site_ruby/1.9.1"
"/opt/local/lib/ruby1.9/site_ruby/1.9.1/powerpc-darwin9"
"/opt/local/lib/ruby1.9/site_ruby"
"/opt/local/lib/ruby1.9/vendor_ruby/1.9.1"
"/opt/local/lib/ruby1.9/vendor_ruby/1.9.1/powerpc-darwin9"
"/opt/local/lib/ruby1.9/vendor_ruby"
"/opt/local/lib/ruby1.9/1.9.1"
"/opt/local/lib/ruby1.9/1.9.1/powerpc-darwin9"
```

What's this !

```
$ gem env
- INSTALLATION DIRECTORY:
/opt/local/lib/ruby/gems/1.9.1
```

```
$ irb
>> Gem.path
=> ["/opt/local/lib/ruby1.9/gems/1.9.1"]

$LOAD_PATH.each {|pth| p pth}
"/opt/local/lib/ruby1.9/gems/1.9.1/gems/fastercsv-1.5.4/lib"
"/opt/local/lib/ruby1.9/gems/1.9.1/gems/rubygems-update-1.3.7/bin"
"/opt/local/lib/ruby1.9/site_ruby/1.9.1"
"/opt/local/lib/ruby1.9/site_ruby/1.9.1/powerpc-darwin9"
"/opt/local/lib/ruby1.9/site_ruby"
"/opt/local/lib/ruby1.9/vendor_ruby/1.9.1"
"/opt/local/lib/ruby1.9/vendor_ruby/1.9.1/powerpc-darwin9"
"/opt/local/lib/ruby1.9/vendor_ruby"
"/opt/local/lib/ruby1.9/1.9.1"
"/opt/local/lib/ruby1.9/1.9.1/powerpc-darwin9"
```

Let's Confirm that..

```
$ ls /opt/local/lib/ruby/gems/1.9.1/gems

ai4r-1.9  distance_measures-0.0.5  highline-1.6.2
net-scp-1.0.4    net-ssh-2.2.1      rake-0.9.2
capistrano-2.8.0  fastercsv-1.5.4     k_means-0.0.7
net-sftp-2.0.5  net-ssh-gateway-1.1.0  rubygems-update-1.8.10

>> ls /opt/local/lib/ruby1.9/gems/1.9.1
fastercsv-1.5.4     rubygems-update-1.3.7
```

Solutions?

We could change/modify **irb/ruby Gem.path** within **ruby** or **irb** (file by file)

Could set an ENV variable GEM_HOME or GEM_PATH in our .bash_profile or .profile

NB; latter will change gems for all your rubys

```
export GEM_HOME='/opt/local/lib/ruby/gems/2.0.0'
export GEM_PATH='/opt/local/lib/ruby/gems/2.0.0'
export PATH=/opt/local/bin:opt/local/sbin:$PATH
```

.bash_profile

Second Main Problem

Gems have dependencies to other gems; load one and several appear (**k_means** loads **distance_measures**)

Gems have dependencies to other applications in your system; in Mac Xcode has a bunch of tools that may be used (4.1 not-equal 7.1)

if your directories and paths are all not clean then you may encounter horrible problems (e.g, loading from source AND package man. system)

Also means that gems compiled under one ruby version and OS may not run with new ruby and OS

Part E:

Getting out of the house...

Gems, Ruby & the Web

We are going to look at getting out of the house in two main ways:

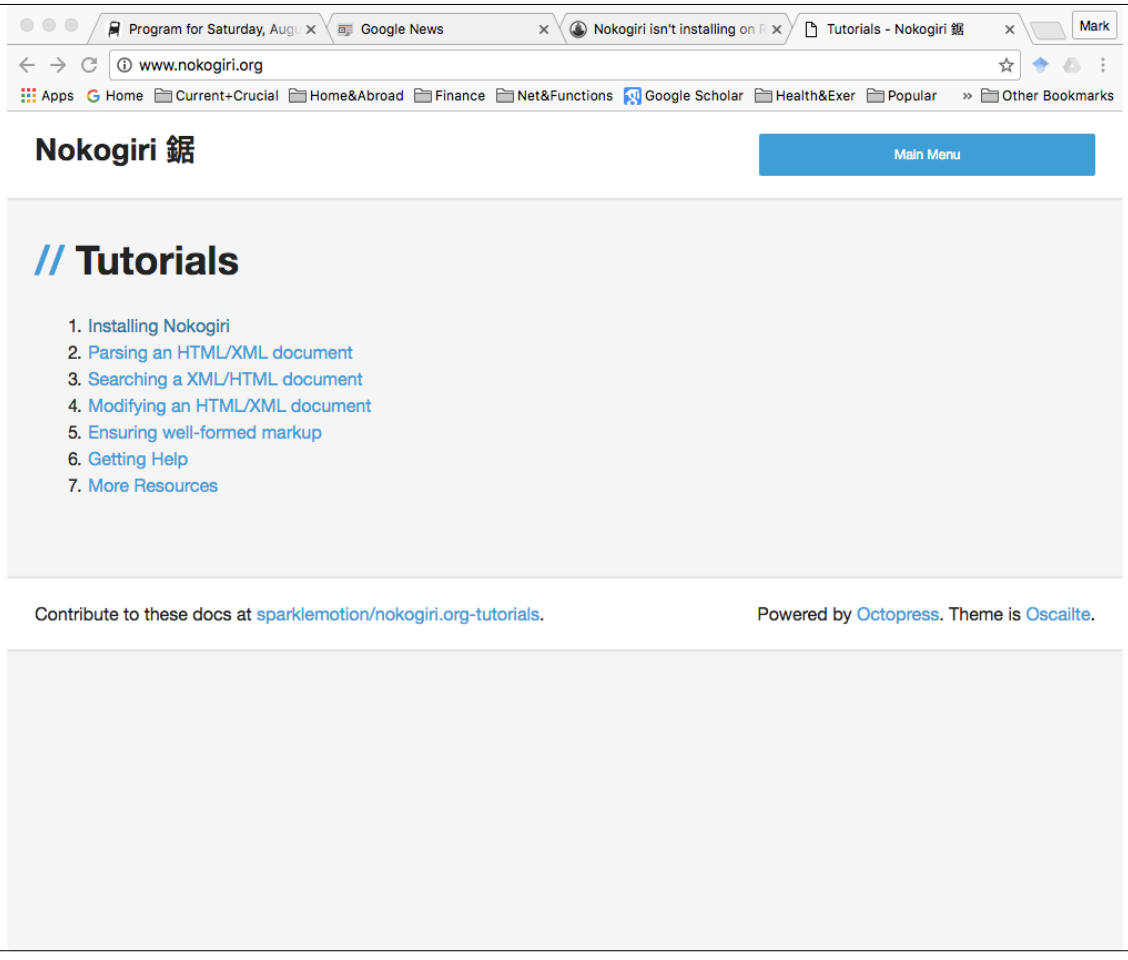
1) in doing some basic web-page parsing and scraping using Nokogiri

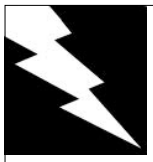
2) in showing how with a few simple lines of code you can become a spammer

we also now start to use gems

Nokogiri for 2.3.1

```
MacBook-Air-4:~ user$ sudo gem install nokogiri -- --use-system-libraries
Fetching: mini_portile2-2.1.0.gem (100%)
Successfully installed mini_portile2-2.1.0
Fetching: pkg-config-1.1.7.gem (100%)
Successfully installed pkg-config-1.1.7
Fetching: nokogiri-1.6.8.gem (100%)
Building native extensions with: '--use-system-libraries'
This could take a while...
Successfully installed nokogiri-1.6.8
Parsing documentation for mini_portile2-2.1.0
Installing ri documentation for mini_portile2-2.1.0
Parsing documentation for pkg-config-1.1.7
Installing ri documentation for pkg-config-1.1.7
Parsing documentation for nokogiri-1.6.8
Installing ri documentation for nokogiri-1.6.8
Done installing documentation for mini_portile2, pkg-config, nokogiri after 3
seconds
3 gems installed
MacBook-Air-4:~ user$
```





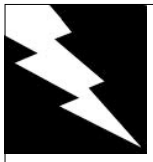
Nokogiri

is a library for parsing html and xml

basically, turns doc into an hierarchy of embedded elements

from that you can pull out what you want into arrays of elements (uses Xpath)

http://nokogiri.org/



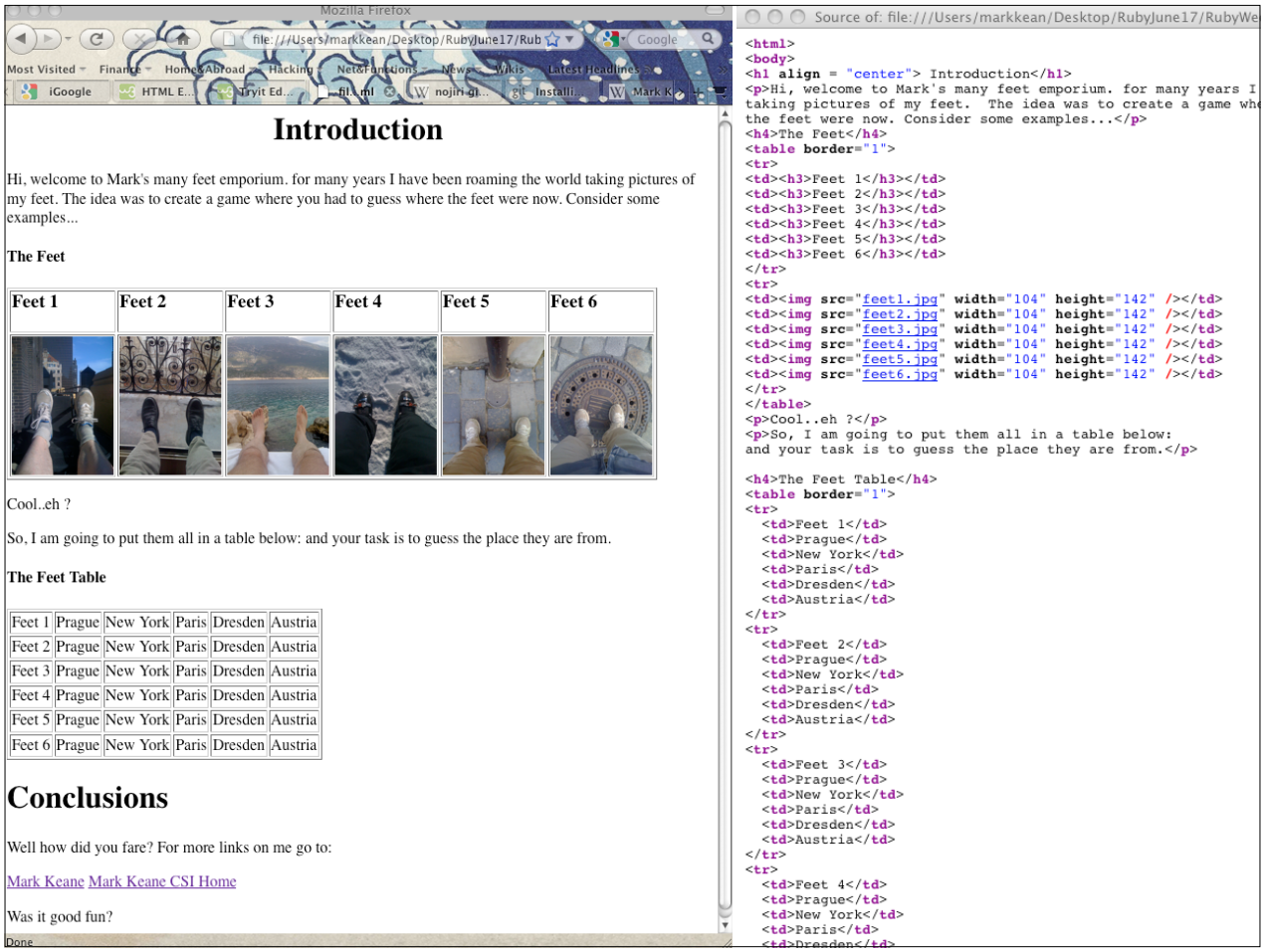
Nokogiri

There is always great fun with Nokogiri every year…but it is learning fun…

Notably, it depends on a number of earlier libraries that can cause problems…follow instructions carefully

Also, on Macs, I found installing the gem in RubyMine worked better than doing it via the CommandLine

http://nokogiri.org/



Nokogiri Parse

```
require 'nokogiri'
require 'pp'
file = File.open("feet/feet.html")
doc = Nokogiri::HTML(file)
```

```
#(Document:0x95150 {
  name = "document",
  children = [
    #(DTD:0x936f2 { name = "html" }),
    #(Element:0x930d0 {
      name = "html",
      children = [
        #(Element:0x92a04 {
          name = "body",
          children = [
            #(Text "\n"),
            #(Element:0x92234 {
              name = "h1",
              attributes = [
                #(Attr:0x91f50 { name = "align", value = "center" })],
              children = [ #(Text " Introduction")]
              }),
            #(Text "\n"),
            #(Element:0x90ad8 {
              name = "p",
              children = [
                    #(Text "Hi, welcome to Mark's many feet emporium. for many years I have been roaming the world\ntaking pictures of my feet.  The idea was to create a game where you had to guess where \nthe feet were now. Consider some examples...")]
              }),

...
```

Simple Nokogiri I

```
require 'nokogiri'

doc = Nokogiri::HTML(File.open("feet/feet.html"))
elements = doc.xpath("/html/body//p")
p elements
```

dochac.rb

[#<Nokogiri::XML::Element:0x3d23b8 name="p" children=[#<Nokogiri::XML::Text:0x3d1bfc "Hi, welcome to Mark's many feet emporium. for many years I have been roaming the world\ntaking pictures of my feet. The idea was to create a game where you had to guess where \nthe feet wer now. Consider some examples...">]>, #<Nokogiri::XML::Element:0x3d181e name="p" children=[#<Nokogiri::XML::Text:0x3d11f2 "Cool..eh ?">]>, #<Nokogiri::XML::Element:0x3d0860 name="p" children=[#<Nokogiri::XML::Text:0x3cf4ba "So, I am going to put them all in a table below:\nand your task is to guess the place they are from.">]>, #<Nokogiri::XML::Element: 0x3ce6fa name="p" children=[#<Nokogiri::XML::Text:0x3cdcc8 "Well how did you fare? For more links on me go to:">]>, #<Nokogiri::XML::Element:0x3cd7a0 name="p" children=[#<Nokogiri::XML::Text:0x3cd304 "Was it good fun?">]>]

an array with all the p elements

Simple Nokogiri II

```
require 'nokogiri'

doc = Nokogiri::HTML(File.open("feet/feet.html"))
elements = doc.xpath("//p")
p elements.length
p elements.last
p elements.last.inner_html
p elements.last.inner_text
```

dochac.rb

```
$ ruby dochac.rb
 5
<Nokogiri::XML::Element:0x3cd7a0 name="p"
children=[#<Nokogiri::XML::Text:0x3cd304 "Was it good
fun?">]>
 "Was it good fun?"
 "Was it good fun?"
```

nokogiri element

its inner html

its inner text

Simple Nokogiri III

```
require 'nokogiri'

doc = Nokogiri::HTML(File.open("feet/feet.html"))
elements4 = doc.xpath("//img")
p elements4
```

dochac.rb

[#<Nokogiri::XML::Element:0x3c2a6c name="img" attributes=[#<Nokogiri::XML::Attr:0x3c2a30 name="src" value="feet1.jpg">, #<Nokogiri::XML::Attr:0x3c2a26 name="width" value="104">, #<Nokogiri::XML::Attr:0x3c2a1c name="height" value="142">]>, #<Nokogiri::XML::Element:0x3ca564 name="img" attributes=[#<Nokogiri::XML::Attr:0x3ca514 name="src" value="feet2.jpg">, #<Nokogiri::XML::Attr:0x3ca4f6 name="width" value="104">, #<Nokogiri::XML::Attr:0x3ca4ec name="height" value="142">]>, #<Nokogiri::XML::Element:0x1ff630 name="img" attributes=[#<Nokogiri::XML::Attr:0x1ff27a name="src" value="feet3.jpg">, #<Nokogiri::XML::Attr:0x1ff23e name="width" value="104">, #<Nokogiri::XML::Attr:0x1ff13a name="height" value="142">]>, #<Nokogiri::XML::Element:0x1fcff2 name="img" attributes=[#<Nokogiri::XML::Attr:0x1fcfa2 name="src" value="feet4.jpg">, #<Nokogiri::XML::Attr:0x1fcf98 name="width" value="104">, #<Nokogiri::XML::Attr:0x1fcef8 name="height" value="142">]>, #<Nokogiri::XML::Element:0x1f61e8 name="img" attributes=[#<Nokogiri::XML::Attr:0x1f58d8 name="src" value="feet5.jpg">, #<Nokogiri::XML::Attr:0x1f58b0 name="width" value="104">, #<Nokogiri::XML::Attr:0x1f586a name="height" value="142">]>, #<Nokogiri::XML::Element:0x94f70 name="img" attributes=[#<Nokogiri::XML::Attr:0x94e76 name="src" value="feet6.jpg">, #<Nokogiri::XML::Attr:0x94e08 name="width" value="104">, #<Nokogiri::XML::Attr:0x94dea name="height" value="142">]>]

Simple Nokogiri IV

```
require 'nokogiri'

doc = Nokogiri::HTML(File.open("feet/feet.html"))
elements = doc.xpath("//td")

elements = doc.search("//td")
p elements[7]
p elements[7].inner_html
p elements[7].inner_text
```

dochac.rb

#<Nokogiri::XML::Element:0x3caa32 name="td" children=[#<Nokogiri::XML::Element:0x3ca564 name="img" attributes=[#<Nokogiri::XML::Attr:0x3ca514 name="src" value="feet2.jpg">, #<Nokogiri::XML::Attr:0x3ca4f6 name="width" value="104">, #<Nokogiri::XML::Attr:0x3ca4ec name="height" value="142">]>]>

nokogiri element

"<img src=\"feet2.jpg\" width=\"104\" height=\"142\">"

its inner html

""

its inner text

Simple Nokogiri V

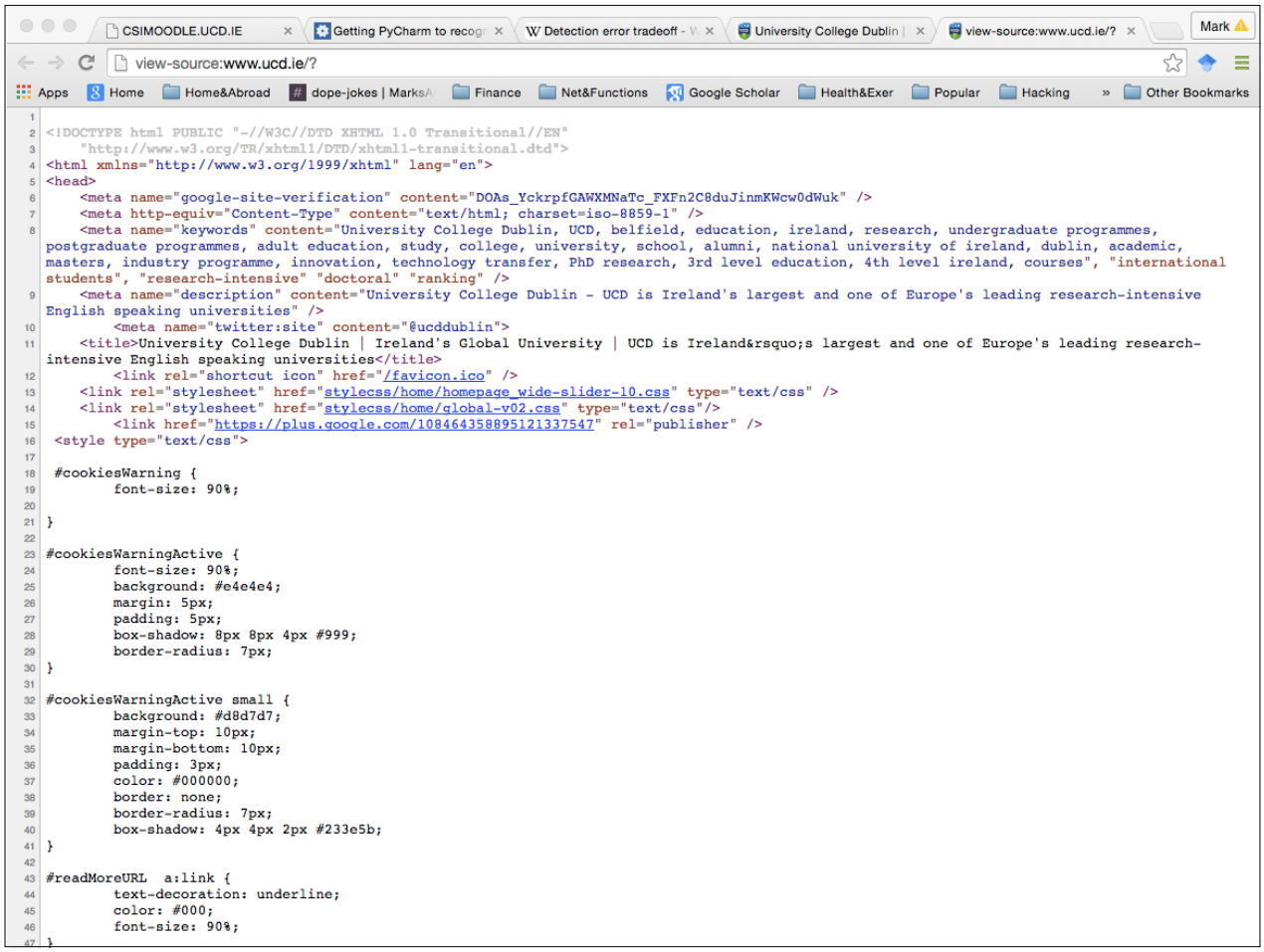
```
require 'nokogiri'

elements4 = doc.xpath("//img")

def search_for_image_names(parse_a)
  parse_a.search("//td").each do |td_element|
    imgs = td_element.search("img")
    if imgs.any?
        then imgs.each {|image| puts "Found Image called #{image[:src]}"} end
   end
end

search_for_image_names(elements4)
```

we can search for sub-elements

get attributes of sub-elements

dochac.rb

Found Image called feet1.jpg
Found Image called feet2.jpg
Found Image called feet3.jpg
Found Image called feet4.jpg
Found Image called feet5.jpg
Found Image called feet6.jpg




```
require 'nokogiri'
require 'open-uri'
require 'pp'
doc = Nokogiri::HTML(open("http://www.ucd.ie")
pp doc
```

ruby dochacweb.rb
/usr/bin/ruby -e $stdout.sync=true;$stderr.sync=true;load($0=ARGV.shift) /Users/user/Dropbox/X_Teaching/Ruby 2013-15/A_Lects&Pracs.2015/RubyWeek7 (Oct 22th).14/RubyLect7.progs/dochacweb.rb
#(Document:0x3fc77d0f89ec {
name = "document",
children = [
#(DTD:0x3fc77d0f8154 { name = "html" }),
#(Element:0x3fc77d0fdadc {
name = "html",
attributes = [
#(Attr:0x3fc77d0fd6f4 {
name = "xmlns",
value = "http://www.w3.org/1999/xhtml"
}),
#(Attr:0x3fc77d0fd6e0 { name = "lang", value = "en" })],
children = [
#(Text "\n"),
#(Element:0x3fc77d101970 {
name = "head",
children = [
#(Text "\n "),
#(Element:0x3fc77d100e94 {
name = "meta",
attributes = [
...

Nokogiri Parse

Simple Nokogiri I (on Web)

```
require 'nokogiri'
require 'open-uri'
require 'pp'

doc = Nokogiri::HTML(open("http://www.ucd.ie"))
elements = doc.xpath("//a[@href]")
p elements.length
p elements[50]
item = elements[50]["href"]
p item
```

dochacweb.rb

$ruby dochacweb.rb — no of a-href tags — nokogiri element

```
70
#<Nokogiri::XML::Element:0x3ff1a855e860 name="a" attributes=[#<Nokogiri::XML::Attr:0x3ff1a855e7e8
name="href" value="news/2015/08AUG15/200815-ucd-hosts-world-rugby-womens-sevents-qualifier-
series.html">] children=[#<Nokogiri::XML::Element:0x3ff1a855e324 name="img" attributes=[#<Nokogiri::XML::Attr:
0x3ff1a855e2c0 name="src" value="news/2015/08AUG15/200815-ucd-hosts-world-rugby-womens-sevents-
qualifier-series-fp.jpg">, #<Nokogiri::XML::Attr:0x3ff1a855e2ac name="alt" value="UCD hosts World Rugby
Women's Sevens qualifier series">]>, #<Nokogiri::XML::Text:0x3ff1a8563cac "UCD hosts World Rugby Women's
Sevens qualifier series">]>
"news/2015/08AUG15/200815-ucd-hosts-world-rugby-womens-sevents-qualifier-series.html"
```

value of href attribute

Simple Nokogiri I (on Web)

```
require 'nokogiri'
require 'open-uri'
require 'pp'

doc = Nokogiri::HTML(open("http://www.ucd.ie"))
elements = doc.xpath("//a[@href]")
p elements.length
p elements[50]
item = elements[50]["href"]
p item
```

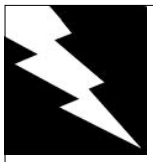
use Nokogiri:XML iff XML doc

dochacweb.rb

ruby dochacweb.rb — no of a-href tags — nokogiri element

```
111
#<Nokogiri::XML::Element:0x3f825c name="a" attributes=[#<Nokogiri::XML::Attr:
0x3f81bc name="href" value="/wiki/University_of_London">,
#<Nokogiri::XML::Attr:0x3f819e name="title" value="University of London">]
children=[#<Nokogiri::XML::Text:0x3f7d5c "University of London">]>
"/wiki/University_of_London"
```

value of href attribute



Sending an Email

first install actionmailer

$ gem install actionmailer

...it gets sort of busy

…20 gem installed

$ gem install rails

$ gem install rack

Sending an Email

```
require 'action_mailer'

ActionMailer::Base.smtp_settings = {   :address => 'smtp.gmail.com',
                                       :port => 25,
                                       :domain => 'ucd.ie',
                                       :user_name => 'youreUCDusernamehere',
                                       :password => 'yourpassword',
                                       :authentication  => :plain }

class SimpleMailer < ActionMailer::Base
  def simple_message(recipient)
   mail(:from => 'youremailaddresshere@ucd.ie',
        :to => recipient,
        :subject => 'tester',
        :body => 'wow!, just sent you this from a program')
  end
end

email = SimpleMailer.simple_message('youremailaddresshere@ucd.ie')
puts email
email.deliver
```

mailer.rb

Please Go Home Now
& Have a Rest....