Ruby Explorations I

Mark Keane...CSI...UCD

De Basics

Part I: History, Environment and Progs


```
RubyLect2n.progs — -bash — 105×29
MacBook-Air-2:RubyLect2n.progs user$
MacBook-Air-2:RubyLect2n.progs user$ ruby -v
ruby 2.3.1p112 (2016-04-26 revision 54768) [x86_64-darwin15]
MacBook-Air-2:RubyLect2n.progs user$ ruby test.rb
hi there big boy !
MacBook-Air-2:RubyLect2n.progs user$ irb -v
irb 0.9.6(09/06/30)
MacBook-Air-2:RubyLect2n.progs user$ irb
irb(main):001:0> puts $LOAD_PATH
/opt/local/lib/ruby2.3/gems/2.3.0/gems/did_you_mean-1.0.0/lib
/opt/local/lib/ruby2.3/site_ruby/2.3.0
/opt/local/lib/ruby2.3/site_ruby/2.3.0/x86_64-darwin15
/opt/local/lib/ruby2.3/site_ruby
/opt/local/lib/ruby2.3/vendor_ruby/2.3.0
/opt/local/lib/ruby2.3/vendor_ruby/2.3.0/x86_64-darwin15
/opt/local/lib/ruby2.3/vendor_ruby
/opt/local/lib/ruby2.3/2.3.0
/opt/local/lib/ruby2.3/2.3.0/x86_64-darwin15
=> nil
irb(main):002:0> ruby test.rb
ArgumentError: wrong number of arguments (given 0, expected 2..3)
        from (irb):2:in `test'
        from (irb):2
        from /opt/local/bin/irb:11:in `<main>'
irb(main):003:0> quit
MacBook-Air-2:RubyLect2n.progs user$ ruby test.rb
hi there big boy !
MacBook-Air-2:RubyLect2n.progs user$
MacBook-Air-2:RubyLect2n.progs user$
```

Done !

http://www.ruby-lang.org/en/downloads/

http://rubyosx.rubyforge.org one-click installer for OSX

directions for Windows, MacOSX (bundled) and Linux (see Thomas , 2009, chapter 1)

get Ruby 2.3.1 (latest stable version)

Most basic, use a text editor to write files, a cshell to run files and navigate directories

ruby finename.rb [runs the file called *filename*]

irb [gets you the interactive version]

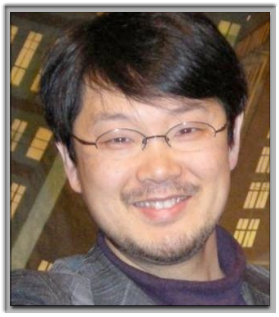
http://www.ruby-doc.org/core/

Ruby History

created by Yukihiro Matsumoto in 1993

a language balancing functional and imperative programming

Tubular Bells of programming...



Flanagan, D & Matsumoto, Y. (2008).
The Ruby Programming Language. O'Reilly.

Aside on Progamming

imperative pg: computation is statements that change a program state; an algorithm with explicit steps or procedures (e.g. C, BASIC)

declarative pg: logic of computation without flow of control, what the program should achieve not how it achieves (e.g. Prolog)

functional pg: computation as the evaluation of mathematical functions avoiding state and mutable data (e.g. Scheme, Lisp)

Ruby Precursors

imperative languages were first (Fortran, Pascal, Basic, Algol, Ada, C)

extended to have objects in OOP (***Smalltalk***-80, C++, Python, ***Perl***, ***Eiffel***); real-world analogy

into the modern era with Java and Ruby...

but, with another line from functional programming (many ***LISP***-like features)

http://www.ruby-doc.org/core/

Me

Chair of CS since 1998

BA (Psych, UCD, '82),

PhD (Psych, TCD, '87)

Fellow TCD (1994)

SFI (2004-07)

VP Innovation (2007-09)

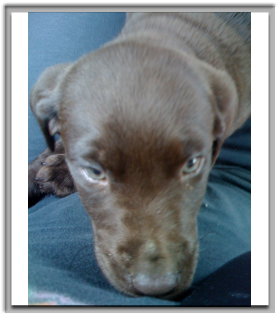
Big Shot (-present)

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

My Puppy

this is my puppy

her name is ...



Ruby is Object-Oriented

is a true OOP lang, methods are invoked on objects

syntax used DOT (.) operator

these egs are the purest cases

```
>> 43.class
=> Fixnum

>>"string".length
=> 6

>>"stri".length
=> 4

>> 5.to_s
"5"
```

Ruby is Object-Oriented

43, **"string"**, **"stri"** and **5** are all objects

class, **length**, **to_s** are all methods

everything is an *expression*; made up of an *object* and a *method*

```
>> 43.class
=> Fixnum

>>"string".length
=> 6

>>"stri".length
=> 4

>> 5.to_s
"5"
```

every *expression* evaluates-to/returns a value

Method Syntax I: DOT

underscores not hypens

DOT means the *message* "to_s" is being *sent to* the number 5, 5 is the *receiver* of the *message*

OR, the number 5 is being asked "*convert yourself to a string*"

```
>> 5.to_s
=> "5"
>> "5".to_i
=> 5
>> "mark".class
=> String
>> 5.class
=> Fixnum
```

fns in lowercase

classes start with CAPS

Syntax II: Parameter Args

class and ***to_s*** are like methods with a single, object arg using DOT

other methods are written with arg-style

...or better use arg in brackets

safer to assume brackets

extrapolate **String**

```
>> 5.to_s
=> "5"
>> Float 5
=> 5.0
>> 5.Float
NoMethodError:...
>> Float(5)
=> 5.0
>> String("5")
=> "5"
```

kernel methods syntax

Syntax III: Dot + Parameters

something? is predicate

ask object "dna" the message does it include "na"

can have many args

```
  0 1 2
  m a g
```

note, positions are counted off from zero

```
>> "dna".include?("na")
=> true
>> "dna".include("we")
=> NoMethodError:...
>> "dna".include?("we")
=> false
>> "mag".insert(3, "oooo")
=> "magoooo"
>> "mag".insert(2, "ooo")
=> "maooog"
```

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

Say Hello to print...

$stdout.puts("hello")

we will ignore ***print*** and use ***puts*** and ***p***

whoaaaah Nelly !

puts sends args to screen with <cr>

...then it returns a value , namely ***nil***

```
>> puts "hello"
hello
=> nil
>> puts("hello", "mo")
hello
mo
=> nil
>> p "hello"
"hello"
=> "hello"
```

kernel methods syntax

try this with no parenth..

will cause bugs

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

Variables I:Local

= is used for variable assignment; **not** to be confused with **==** equality

will cause bugs

```
>> a = 1
=> 1
>> b = 2
=> 2
>> a + b
=> 3
>> c = 2
=> 2
>> c == b
=> true
>> c = a
=> 1
>> c
=> 1
>> c += 4
=> 5
```

+ operator has own syntax

a+(b)

lowercase indicates local

Variables II:Local

= is used for variable assignment; **not** to be confused with **==** equality

will cause bugs

what's odd about **+**

```
>> name = "mark"
=> "mark"
>> surname = "keane"
=> "keane"
>> name + surname
=> "markkeane"
>> name + " " + surname
=> "mark keane"
>> name2 = "mark"
=> "mark"
>> name2 == name
=> true
```

lowercase indicates local

Variables: Global

capital indicates global

best avoided but if you have to...

access all areas...

used for Constants

```
>> Currency = “dollars”
=> “dollars”
>> Solid = 56
=> 56
>> Solid = 45
(irb):42: warning: already initialized constant Solid
(irb):38: warning: previous definition of Solid was here
=> 45
>> Solid
=> 45
```

The Usual Types

underscores not hypens

Strings

Numbers

Arrays

Hash Tables

Your own objs

symbols used as hash keys

```
>> “mark”.instance_of?(String)
=> true
>> “mark”.instance_of?(Array)
=> false
>> [“a”,”b”].instance_of?(Array)
=> true
>> [“a”,”b”,”c”].length
=> 3
>> :mark.class
=> Symbol
```

redo with symbols

Its functional ...with an F

Every function call returns a value, of last evaluated expression

Results in most common bug

```
NoMethodError:
undefined method
`any_fun' for
nil:NilClass
```

```
>> puts “hi mark”
hi mark
=> nil
>> a = “foo”
=> “foo”
>> b = (puts a)
foo
=> nil
>> a
=> “foo”
>> b
=> nil
```

why?

what would **p** do

Defining a function

underscores not hypens

def is used to define a function

we are doing it in **irb** which is awkward

next, we will do it in a file which is nicer

see how we include parameters

some **nil**s go to limbo

```
>> def hail_the_king
>> puts “king mark”
>> end
=> :hail_any_king
>> hail_the_king
king mark
=> nil
>> def hail_any_king(me)
>> puts “hail”
>> puts me
>> end
=> :hail_any_king
>> hail_any_king(“sam”)
hail
sam
=> nil
```

parameter for fn

me is local to fn scope

From irb to ruby

watch quotes

Create a file in a text editor of your choice

start console

run file using ruby

but double check what /usr/bin/ruby is?

```
def hail_the_king
    puts “hail king mark”
end
```

king.rb

```
def hail_the_king
    puts “hail king mark”
end

hail_the_king
```

king.rb

```
$ ruby king.rb
$
$ ruby king.rb
hail king mark
$
```

where's nil?

REM Cycle

Edit file

Save file

Re-run file using Ruby...

will cause bugs

```
def hail_the_king
    puts “hail king marko”
end

hail_the_king
```

king.rb

```
$ ruby king.rb
$
$ ruby king.rb
hail king marko
$
```

A.D. Hacker “I am changing my file, saving it but I am getting same wrong answer when I run it ?”

Oh…now I see !

```
def hail_the_king
    puts "hail king marko"
end

print $LOAD_PATH
hail_the_king
```

king.rb

```
$ ruby king.rb
$["/opt/local/lib/ruby2.3/gems/2.3.0/gems/
did_you_mean-1.0.0/lib", "/opt/local/lib/ruby2.3/
site_ruby/2.3.0", "/opt/local/lib/ruby2.3/
site_ruby/2.3.0/x86_64-darwin15", "/opt/local/lib/
ruby2.3/site_ruby", "/opt/local/lib/ruby2.3/
vendor_ruby/2.3.0", "/opt/local/lib/ruby2.3/
vendor_ruby/2.3.0/x86_64-darwin15", "/opt/local/
lib/ruby2.3/vendor_ruby", "/opt/local/lib/
ruby2.3/2.3.0", "/opt/local/lib/ruby2.3/2.3.0/
x86_64-darwin15"]Versions/2.0/usr/lib/ruby/2.0.0/
x86_64-darwin15", "/System/Library/Frameworks/
Ruby.framework/Versions/2.0/usr/lib/ruby/2.0.0/
universal-darwin15"-darwin15", "/System/Library/
Frameworks/Ruby.framework/Versions/2.0/usr/lib/
```

Teasers

What **ruby** does **irb** use?

```
markkean% irb -v
irb 0.9.6(09/06/30)

markkean% irb1.8 -v
irb 0.9.5(05/04/13)

markkean% irb conf
/opt/local/lib/ruby2.3/2.3.0/irb/magic-file.rb:8:in `initialize': No such file or directory @ rb_sysopen - conf
(Errno::ENOENT)
    from /opt/local/lib/ruby2.3/2.3.0/irb/magic-file.rb:8:in `open'
    from /opt/local/lib/ruby2.3/2.3.0/irb/magic-file.rb:8:in `open'
    from /opt/local/lib/ruby2.3/2.3.0/irb/input-method.rb:101:in `initialize'
    from /opt/local/lib/ruby2.3/2.3.0/irb/context.rb:85:in `new'
    from /opt/local/lib/ruby2.3/2.3.0/irb/context.rb:85:in `initialize'
    from /opt/local/lib/ruby2.3/2.3.0/irb.rb:426:in `new'
    from /opt/local/lib/ruby2.3/2.3.0/irb.rb:426:in `initialize'
    from /opt/local/lib/ruby2.3/2.3.0/irb.rb:381:in `new'
    from /opt/local/lib/ruby2.3/2.3.0/irb.rb:381:in `start'
    from /opt/local/bin/irb:11:in `<main>'

markkean% irb1.9 conf
/opt/local/lib/ruby1.9/1.9.1/irb/magic-file.rb:7:in `initialize': No such file or directory - conf (Errno::ENOENT)
        from /opt/local/lib/ruby1.9/1.9.1/irb/magic-file.rb:7:in `open'
        from /opt/local/lib/ruby1.9/1.9.1/irb/magic-file.rb:7:in `open'
        from /opt/local/lib/ruby1.9/1.9.1/irb/input-method.rb:77:in `initialize'
        from /opt/local/lib/ruby1.9/1.9.1/irb/context.rb:79:in `new'
        from /opt/local/lib/ruby1.9/1.9.1/irb/context.rb:79:in `initialize'
        from /opt/local/lib/ruby1.9/1.9.1/irb.rb:91:in `new'
        from /opt/local/lib/ruby1.9/1.9.1/irb.rb:91:in `initialize'
        from /opt/local/lib/ruby1.9/1.9.1/irb.rb:56:in `new'
        from /opt/local/lib/ruby1.9/1.9.1/irb.rb:56:in `start'
        from /opt/local/bin/irb1.9:12:in `<main>'

markkean%
```

uses different path variables

Geek Crap I: Search Paths

Move along, there is nothing to see here....

Ruby uses many environmental variables to store search paths and the like... under the bonnet

MacBook-Air-2$ ruby -e 'puts $LOAD_PATH'

```
/opt/local/lib/ruby2.3/gems/2.3.0/gems/did_you_mean-1.0.0/lib
/opt/local/lib/ruby2.3/site_ruby/2.3.0
/opt/local/lib/ruby2.3/site_ruby/2.3.0/x86_64-darwin15
/opt/local/lib/ruby2.3/site_ruby
/opt/local/lib/ruby2.3/vendor_ruby/2.3.0
/opt/local/lib/ruby2.3/vendor_ruby/2.3.0/x86_64-darwin15
/opt/local/lib/ruby2.3/vendor_ruby
/opt/local/lib/ruby2.3/2.3.0
/opt/local/lib/ruby2.3/2.3.0/x86_64-darwin15
```

REM:

Geek Crap II: Search Paths

BUT...if I check my other version of Ruby

```
[Mouseking6-3:~] markkean% ruby1.8 -e 'puts $LOAD_PATH'

/Library/Ruby/Site/1.8
/Library/Ruby/Site/1.8/powerpc-darwin9.0
/Library/Ruby/Site/1.8/universal-darwin9.0
/Library/Ruby/Site
/System/Library/Frameworks/Ruby.framework/Versions/1.8/usr/lib/ruby/1.8
/System/Library/Frameworks/Ruby.framework/Versions/1.8/usr/lib/ruby/1.8/powerpc-darwin9.0
/System/Library/Frameworks/Ruby.framework/Versions/1.8/usr/lib/ruby/1.8/universal-darwin9.0
```

REM:

The story so far ...

function syntax

variables

equality

basic functions

defining functions

Using ***irb***

Using ***ruby***

Now, U Try It ...Prac 1

start practical here and work through

take big problems to class hour later in week

practical needs to be turned in by end of class hour; with your name on it

practical will form part of the continuous assessment of the course