**Ruby language**

<http://www.ruby-lang.org/>

**Documentation**

<http://ruby-doc.org/>

<http://www.ruby-lang.org/en/documentation/>

**Recommended (Additional Resource):**

*Programming Ruby 1.9 & 2.0, The Pragmatic Programmers’ Guide*

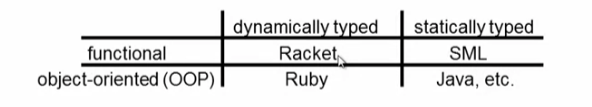
**Why Ruby?**

* Pure object-oriented
  + All values are objects (even numbers)
* Class-based
  + Every object has a class that determines behavior
    - Java, unlike Javascript
    - Mixins (neither Java interfaces nor C++ multiple inheritance)
* Dynamically typed
* Convenient reflection
  + Run-time inspection of objects
* Very dynamic
  + Can change classes during execution
* Blocks and libraries encourage lots of closure idioms
* Syntax, scoping rules, semantics of a “scripting language”
  + Variables “spring to life” on use
  + Very flexible arrays

**Not our focus but is Ruby:**

* Lots of support for string manipulation and regular expressions
* Popular for server-side web app
  + Ruby on Rails
* Often many ways to do the same thing
  + More of a “why not add that too?” approach

**Where Ruby fits**



Note: Racket also has classes and objects when you want them

* In Ruby, everything uses them (at least implicitly)

Historical Note:

* *Smalltalk*
  + Also dynamically typed
  + Class-based
  + Pure OOP language
  + Smaller just as powerful
  + But this is less modern and useful than Ruby

Dynamically typed OOP helps identify OOP’s essence by not having to discuss types

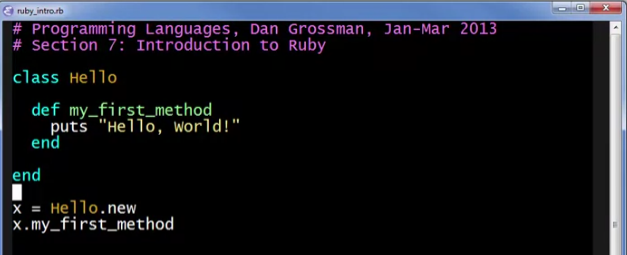
**A note on the homework**

* About understanding and extending an *existing* program in an unfamiliar language
  + Good practice
  + Quite different feel than previous homeworks
  + Read code: determine what you do and do not need to understand

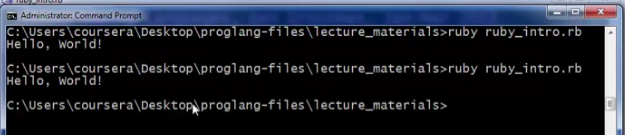
Homework requires the Tk graphics library to be installed

**Small Ruby Program**

*@emacs*



@terminal/command prompt



Running the file: ruby <filename>

Accessing the REPL: irb



load <filename> -> this is like ‘use’ in ML