Ruby Basics

Ruby History

- Invented by Yukihiro “Matz” Matsumoto
- Version 1.0 came out in 1996 (Japan)
- September 2000 – first English language book “Programming Ruby” printed (PickAxe)
- Latest version is up to 2.0
- Made famous by Rails around 2005

Ruby (20K feet view)

- Dynamic

- Object-oriented
  - Object-possessed, almost everything is an object

- **Elegant** and **expressive**
  - Terse at times, but very readable (~~Perl~~)
- Influenced by Perl, Smalltalk, Eiffel and Lisp

...Java...

```java
public class Print3Times {
  public static void main(String[] args){
    for(int i = 0; i < 3; i++) {
      System.out.println("Hello World!");
    }
  }
}
```

…Ruby…

```ruby
3.times do
  puts "Hello World"
end
```

Resources

- http://www.ruby-doc.org/
- Can also use “PickAxe” (Part IV)!!!

The Pragmatic Programmers
Programming Ruby 1.9
The Pragmatic Programmers' Guide
Dave Thomas
The Facets of Ruby Series

Ruby Basics

- 2 space indentation for each nested level is (strongly) encouraged
  - No tabs
- # is used for comments
  - Use comments in stingy moderation – the code itself should tell the story instead

```
# this is a comment
puts 5 # so is this
```

Printing to console

- *puts* - Standard Ruby method to print strings to console (as in **put s**tring)
  - Adds a new line after the printed string
  - Similar to *System.out.println()* in Java
  - Used for most of the examples
- *p* - Prints out internal representation of an object
  - Debugger-style output

```
puts 5 # => 5
puts "something else" # => something else
```

Naming conventions

- Variables
  - Lowercase or `snake_case` if multiple words
- Constants
  - Either `ALL_CAPS` or `FirstCap`
- Classes (and Modules)
  - `CamelCase`

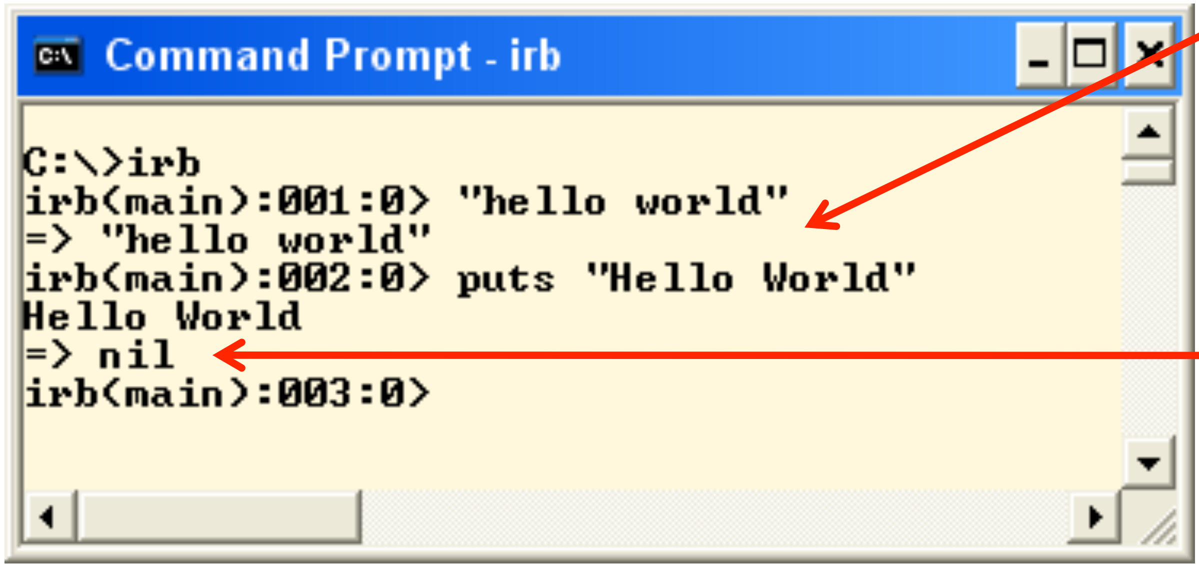
Lose the semicolons

- Leave semicolons off at the end of the line
- Can cram several statements in with a semicolon in between
  - Usually highly discouraged

```
a = 3 # semicolons not needed
a = 2; b = 3 # sometimes used
```

IRB – Interactive Ruby

- Console-based interactive Ruby interpreter
  - REPL (Read Evaluate Print Loop)
- Comes with a ruby installation
- Lets you try stuff out (quickly!)



```
C:\>irb
irb(main):001:0> "hello world"
=> "hello world"
irb(main):002:0> puts "Hello World"
Hello World
=> nil
irb(main):003:0>
```

Anything evaluates to something – no need to assign to a variable

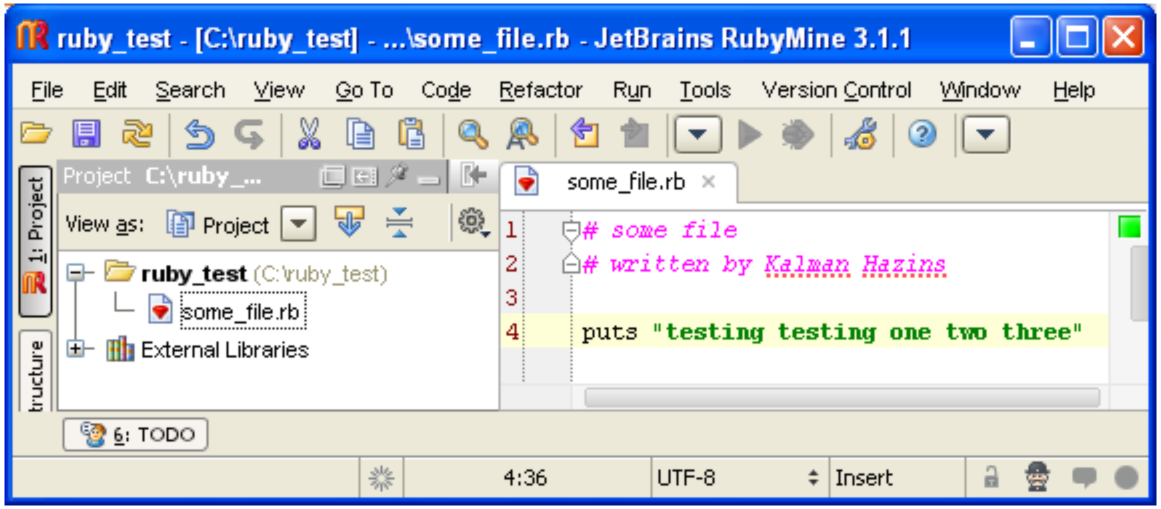
puts returns nil

nil is an object!

- In many languages, `nil` (`null`) means no object
- In Ruby – nil is an object that happens to represent nothing

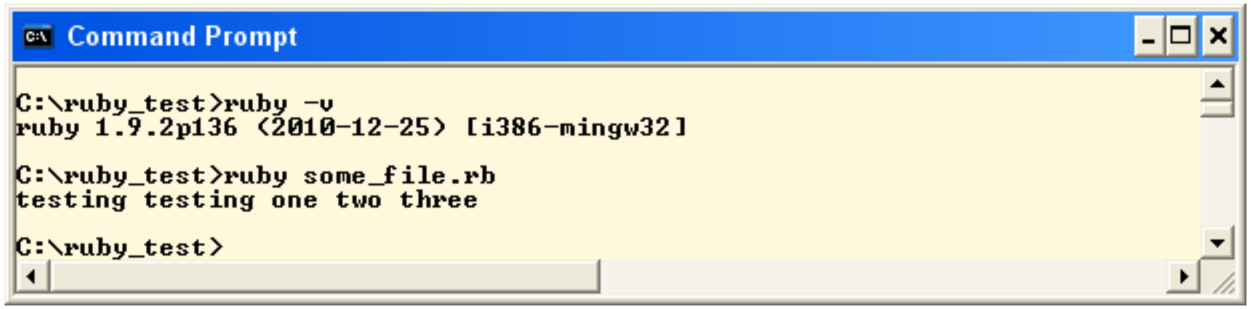
```
puts nil.class # => NilClass
puts nil.nil? # => true
```

Running ruby


ruby_test - [C:\ruby_test] - ...\some_file.rb - JetBrains RubyMine 3.1.1
File Edit Search View Go To Code Refactor Run Tools Version Control Window Help
Project C:\ruby_...
View as: Project
ruby_test (C:\ruby_test)
some_file.rb
External Libraries
6: TODO
4:36 UTF-8 Insert

```ruby
# some file
# written by Kalman Hazins

puts "testing testing one two three"
```



```
C:\ruby_test>ruby -v
ruby 1.9.2p136 (2010-12-25) [i386-mingw32]

C:\ruby_test>ruby some_file.rb
testing testing one two three

C:\ruby_test>
```