

Learning Ruby - 14

Grabbag IV - the *fourth* installment of the *Trilogy*

Creating Iterators

- Adding iterator support (that is, **.each**, **.collect**, **.inject**, etc., etc.) is easy
- Simply mixin the **Enumerable** class into your class
- Use **include Enumerable** near the top of your class code
- Create an **each** method that makes sense for your class

Ethernet.each

```
class Ethernet
  include Enumerable

  def each
    yield @dest
    yield @src
    yield @type_len
    yield @payload
  end # of each.

  ...

end # of Ethernet.
```

Slurping Files

```
string_of_lines = IO::read( "/etc/passwd" )
```

```
array_of_lines = IO::readlines( "/etc/passwd" )
```

Processing Command-line Args

```
# Remember: all arguments are passed in as strings!
```

```
ARGV.each { |arg| puts arg }
```

```
option1, option2 = ARGV.collect { |arg| arg }
```

```
if ARGV.length == 0  
  fail "Please supply some parameters!"  
end # of if.
```

Environments and Configurations

```
ENV.keys.each { |key| print "#{key} -> ", ENV[key], "\n" }
```

```
ENV.collect() do |key, value|  
  print "#{key} --> #{value} \n"  
end # of do
```

```
require 'rbconfig'  
include Config
```

```
CONFIG.keys.sort.each do |key|  
  print "#{key} -> ", CONFIG[key], "\n"  
end # of do.
```

Where Are My Modules?

```
ruby -e 'puts $:'
```

```
/usr/local/lib/site_ruby/1.8  
/usr/local/lib/site_ruby/1.8/i486-linux  
/usr/local/lib/site_ruby/1.8/i386-linux  
/usr/local/lib/site_ruby  
/usr/lib/ruby/1.8  
/usr/lib/ruby/1.8/i486-linux  
/usr/lib/ruby/1.8/i386-linux
```

```
.
```

More ... Ruby So Far

Look for ways to do cool stuff with Ruby.

For example:

- A project proposal for 4th Year:

Create a Ruby extension that allows for the manipulation of the *Torque Game Engine* from within any Ruby program.