

CS 683 Emerging Technologies
Fall Semester, 2006
Doc 16 Ruby Modules, Mixins, IO & Threads
Oct 16, 2006

Copyright ©, All rights reserved. 2006 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/opl.shtml>) license defines the copyright on this document.

References

Programming Ruby, The Pragmatic Programmers' Guide, Dave Thomas, ISBN 0-9745140-5-5

Some examples in this lecture are from the above text

Modules

Namespace

File: example.rb

```
module Example
  PI = 3.1415
  @@x = 1

  def Example.foo
    @@x += 1
  end

  def instance_method
    @@x += 1
  end

  class Bar
    def hello
      'Hello'
    end
  end
end
```

Module
class methods
instance methods
constants
class variables

```
require 'example'

puts Example.foo
puts Example::PI
a = Example::Bar.new
puts a.hello

begin
  puts Example.instance_method
rescue ArgumentError
  puts 'Error'
end
```

Output

```
2
3.1415
Hello
Error
```

Mixins

file: sampleMixin.rb

```
module SampleMixin

  def foo(x)
    x + 1
  end

  def bar
    @y = 0 if @y == nil
    @y += 1
  end

  def name_clash
    'Mixin'
  end

  def call_other
    other + 1
  end

  def other
  end
end
```

file: useMixin.rb

```
require 'sampleMixin'

class UseMixin
  include SampleMixin

  def name_clash
    'Class'
  end

  def y
    @y
  end

  def other
    5
  end
end
```

```
require 'test/unit'
require 'useMixin'
class TestMixin < Test::Unit::TestCase
  def test_mixin
    a = UseMixin.new
    assert_equal(a.foo(2), 3)
    assert_equal(a.name_clash, 'Class')
    assert_equal(a.bar, 1)
    assert_equal(a.bar, 2)
    assert_equal(a.y, 2)
    assert_equal(a.call_other, 6)

  end
end
```

IO

```
file = File.open('testFile')
while line = file.gets
  puts line
end
file.close
```

```
File.open('testFile') do |file|
  while line = file.gets
    puts line
  end
end
```

```
File.open('testFile') do |file|
  file.each_line {|line| puts line}
end
```

```
IO.foreach('testFile') {|line| puts line}
```

```
puts IO.readlines('testFile')
```

```
array = IO.readlines('testFile')
```

```
mode = 'a'
File.open('testFile', mode) do |file|
  file.print 'cat'
  file.puts 'dog'
  file.puts 5
end
```

```
File.open('testFile', mode) do |file|
  file << 'cat' << "dog\n" << 5 << "\n"
end
```

r	read-only, start at beginning of file
r+	read/write, start at beginning of file
w	write-only, start at beginning of file
w+	read/write, start at beginning of file
a	write-only, append to end of file
a+	read/write, start at end of file
b	Binary mode, Windows only, combine with above

Threads

```
require 'net/http'

pages = %w{ www.eli.sdsu.edu www.google.com slashdot.org}
threads = []

for page_to_fetch in pages
  threads << Thread.new(page_to_fetch) do |url|
    browser = Net::HTTP.new(url, 80)
    puts "Fetching #{url}"
    response = browser.get('/', nil)
    puts "Got #{url}: #{response.message}"
  end
end

threads.each {|thread| thread.join }
```

Fetching www.eli.sdsu.eduFetching www.google.comFetching slashdot.org

Got www.eli.sdsu.edu: OK

Got www.google.com: OK

Got slashdot.org: OK

Logging

```
require 'logger'
```

```
log = Logger.new("sample.log", 5, 10*1024)
```

```
log.level = Logger::WARN
```

```
log.debug('debug lowest level')
```

```
log.info('second level')
```

```
log.warn('third level')
```

```
log.error('fourth level')
```

```
log.fatal('getting serious')
```

```
log.unknown('always logged')
```

```
log.warn {'You can use block in put'}
```

```
log.warn('program foo part A') {'You can use both'}
```

Logging Output

```
# Logfile created on Tue Oct 17 14:38:55 PDT 2006 by logger.rb/1.5.2.4
W, [2006-10-17T14:38:55.375719 #1062] WARN -- : third level
E, [2006-10-17T14:38:55.375930 #1062] ERROR -- : fourth level
F, [2006-10-17T14:38:55.376042 #1062] FATAL -- : getting serious
A, [2006-10-17T14:38:55.376139 #1062] ANY -- : always loged
W, [2006-10-17T14:38:55.376249 #1062] WARN -- : You can use block in put
W, [2006-10-17T14:38:55.376354 #1062] WARN -- program foo part A: You can use both
```