Secrets of the standard library

Paul Battley

http://po-ru.com/

@threedaymonk

Forwardable

Forwardable

(expose another object's methods)

```ruby
class Event < ActiveRecord::Base
  belongs_to :opportunity
end
```

```ruby
class Event < ActiveRecord::Base
  belongs_to :opportunity
end

opportunity.organization  #  :-)
event.opportunity  #  :-)
event.opportunity.organization  #  :-(
```

```ruby
class Event < ActiveRecord::Base
  belongs_to :opportunity

  def organization
    opportunity.organization
  end
end

event.organization
```

```ruby
require "forwardable"

class Event < ActiveRecord::Base
  extend Forwardable
  belongs_to :opportunity

  def_delegators :opportunity, :organization
# , ... more methods ...
end

event.organization
```

```ruby
require "forwardable"

class Event < ActiveRecord::Base
  extend Forwardable
  belongs_to :opportunity

  def_delegator :opportunity, :organization,
:opportunity_organization
end

event.opportunity_organization
```

```ruby
class Opportunity < ActiveRecord::Base
  validates_presence_of :name
  alias title name
end
```

```ruby
class Opportunity < ActiveRecord::Base
  validates_presence_of :name
  alias title name
end

# => NameError: undefined method `name' for
class `Opportunity'
```

```ruby
class Opportunity < ActiveRecord::Base
  validates_presence_of :name
  alias_method :title, :name
end
```

```ruby
class Opportunity < ActiveRecord::Base
  validates_presence_of :name
  alias_method :title, :name
end

# => in `alias_method':NameError: undefined
method `name' for class `Opportunity'
```

```ruby
require "forwardable"

class Opportunity < ActiveRecord::Base
  extend Forwardable
  validates_presence_of :name
  def_delegator :self, :name, :title
end
```

```ruby
require "forwardable"

class Opportunity < ActiveRecord::Base
  extend Forwardable
  validates_presence_of :name
  def_delegator :self, :name, :title
end

opportunity.title
```

open

open

(handle input and output)

A file

```ruby
open("somefile.txt", "w") do |f|
  f.puts "Hello"
end
```

A URI

```
require "open-uri"
sample = open("http://google.co.uk"){ |f|
f.read(1024) }
```

A program!

```
open("|less", "w") do |f|
  80.times do |i|
    f.puts "Line #{i}"
  end
end
```

A bit more clever

```
if STDOUT.tty?
  $stdout = open("|less", "w")
end

# output stuff

$stdout.close # <-- don't forget!
```

open3

open3

(interact with other processes)

```ruby
require "open3"

output = Open3.popen3("php"){ |stdin,
stdout, stderr|
  stdin.puts "<? echo 'hello cruel world';
?>"
  stdin.close
  stdout.read
}

# => "hello cruel world"
```

Array#assoc and rassoc

Array#assoc and rassoc

(look up arrays of arrays)

```
data = [["cat",  "kitten", "feline"],
        ["dog",  "puppy",  "canine"],
        ["bear", "cub",    "ursine"]]
```

```ruby
data = [["cat",  "kitten", "feline"],
        ["dog",  "puppy",  "canine"],
        ["bear", "cub",    "ursine"]]

data.assoc("dog")
# => ["dog", "puppy", "canine"]
```

```ruby
data = [["cat",  "kitten", "feline"],
        ["dog",  "puppy",  "canine"],
        ["bear", "cub",    "ursine"]]

data.assoc("dog")
# => ["dog", "puppy", "canine"]

data.rassoc("kitten")
# => ["cat",  "kitten", "feline"]
```

Array#pack and String#unpack

Array#pack and String#unpack

(convert data to and from binary representations)

QT/MPEG4 atom header

```
length, name = @io.read(8).unpack("Na4")

header = [1234, "moov"].pack("Na4")
# => "\000\000\004\322moov"
```

UTF-8

```
codepoints = "string".unpack("U*")
# => [115, 116, 114, 105, 110, 103]

em_dash = [0x2014].pack("U")
# => "—"
```

Set

Set

(mathematical sets)

```ruby
require "set"

a = Set.new([:x, :y, :z])
b = Set.new([:w, :x])

a | b # => #<Set: {:w, :y, :z, :x}>

a ^ b # => #<Set: {:w, :y, :z}>

a & b # => #<Set: {:x}>
```

```ruby
require "set"

vowel_sequences = Set.new
words = %w[ baboon sheep goat stoat weasel ]

words.each do |word|
  word.scan(/[aeiou]+/).each do |sequence|
    vowel_sequences << sequence
  end
end

vowel_sequences
# => #<Set: {"ee", "oo", "a", "e", "ea",
"oa"}>
```

```ruby
require "set"

words = %w[ baboon sheep goat stoat weasel ]

vowel_sequences = words.inject(Set.new){
|set, word|
  set.merge(word.scan(/[aeiou]+/))
}

vowel_sequences
# => #<Set: {"ee", "oo", "a", "e", "ea",
"oa"}>
```

Soap

Soap

(yuk)

```ruby
require "soap/wsdlDriver"
soap =
SOAP::WSDLDriverFactory.new(wsdl_url).create_rpc
result = soap.someRemoteMethod("paramName" => 99
```

pp

pp

(pretty printing)

```
data = [{:name => "stoat",  :family =>
"Mustelidae"},
        {:name => "weasel", :family =>
"Mustelidae"},
        {:name => "skunk",  :family =>
"Mephitidae"}]
```

```
p data
[{:name=>"stoat", :family=>"Mustelidae"},
{:name=>"weasel", :family=>"Mustelidae"},
{:name=>"skunk", :family=>"Mephitidae"}]
```

```
require "pp"
pp data
[{:name=>"stoat", :family=>"Mustelidae"},
 {:name=>"weasel", :family=>"Mustelidae"},
 {:name=>"skunk", :family=>"Mephitidae"}]
```

終

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