

Tham Khảo Nhanh Ruby

Object, block, iterator, regex và I/O file thiết yếu

Cơ Bản

Hello World

```
puts "Hello, World!"
print "no newline"
p [1, 2, 3] # inspect output: [1, 2, 3]
```

Chạy Ruby

```
ruby script.rb # run a file
ruby -e 'puts "hi"' # run inline
irb # interactive REPL
```

Biến

| | |
|-----------------|-----------------------------------|
| name | Biến cục bộ |
| @name | Biến thể hiện (instance variable) |
| @@count | Biến lớp (class variable) |
| \$debug | Biến toàn cục (global variable) |
| MAX_SIZE | Hằng số (chữ hoa theo quy ước) |

Kiểu Dữ Liệu

| | |
|---------------|-------------|
| 42.class | # Integer |
| 3.14.class | # Float |
| "hello".class | # String |
| true.class | # TrueClass |
| nil.class | # NilClass |
| :symbol.class | # Symbol |

String

Cơ Bản về String

```
name = "World"
puts "Hello, #{name}!" # interpolation (double quotes)
puts 'No #{interpolation}' # literal (single quotes)
multi = <<-HEREDOC
  indented heredoc
HEREDOC
```

Phương Thức String

| | |
|----------------------------|-----------------------------|
| .length / .size | Số lượng ký tự |
| .upcase / .downcase | Chuyển đổi chữ hoa/thường |
| .strip | Xóa khoảng trắng đầu/cuối |
| .split(' ', '') | Tách thành mảng |
| .gsub(/pat/, 'rep') | Thay thế toàn bộ |
| .include?('sub') | Kiểm tra có chuỗi con không |
| .start_with?('pre') | Kiểm tra tiền tố |
| .chars / .bytes | Mảng ký tự / bytes |
| .to_i / .to_f | Chuyển sang integer / float |
| .freeze | Làm string bất biến |

Array & Hash

Array

```
arr = [1, "two", :three]
arr << 4 # push (append)
arr[0] # 1
arr[-1] # 4 (last element)
arr[1..2] # ["two", :three] (slice)
```

Phương Thức Array

| | |
|--|----------------------------|
| .push / .pop | Thêm/xóa từ cuối |
| .shift / .unshift | Xóa/thêm từ đầu |
| .flatten | Làm phẳng mảng lồng nhau |
| .compact | Xóa giá trị nil |
| .uniq | Xóa phần tử trùng lặp |
| .sort / .reverse | Sắp xếp / đảo ngược |
| .map { x x * 2 } | Biến đổi từng phần tử |
| .select { x x > 0 } | Lọc phần tử |
| .reduce(0) { sum, x sum + x } | Tích lũy thành giá trị đơn |

Hash

```
user = { name: "Alice", age: 30 } # symbol keys
old = { "key" => "value" } # string keys
user[:name] # "Alice"
user[:email] = "a@b.com" # add pair
user.fetch(:name, "default") # with default
```

Phương Thức Hash

| | |
|----------------------------------|--------------------------|
| .keys / .values | Mảng key / giá trị |
| .each { k, v } | Duyệt từng cặp key-value |
| .merge(other) | Hợp nhất hai hash |
| .key?(k) / .value?(v) | Kiểm tra tồn tại |
| .select { k, v } | Lọc các cặp |
| .transform_values { v } | Biến đổi tất cả giá trị |

Luồng Điều Khiển

Điều Kiện

```
if score >= 90 then "A"
elsif score >= 80 then "B"
else "C"
end
puts "adult" if age >= 18 # inline if
puts "minor" unless age >= 18 # inline unless
```

Case / When

```
case status
when :ok then puts "success"
when :error then puts "failed"
when 400..499 then puts "client error"
else puts "unknown"
end
```

Vòng Lặp

```
5.times { |i| puts i }
(1..10).each { |n| puts n }
while condition do end
until condition do end
loop { break if done }
```

Ternary & Logic

```
status = age >= 18 ? "adult" : "minor"
name = input || "default" # or-assign
name ||= "fallback" # same effect
```

Phương Thức

Định Nghĩa Phương Thức

```
def greet(name, greeting = "Hello")
  "#{greeting}, #{name}!"
end
greet("Alice") # "Hello, Alice!"
greet("Bob", "Hi") # "Hi, Bob!"
```

Giá Trị Trả Về

```
def add(a, b)
  a + b # last expression is implicit return
end
def divide(a, b)
  return nil if b == 0
  a.to_f / b
end
```

Tham Số Keyword & Splat

```
def connect(host:, port: 80, **opts)
  puts "#{host}:#{port} #{opts}"
end
def log(*messages)
  messages.each { |m| puts m }
end
```

Quy Ước Phương Thức

| | |
|--------------------|---|
| method? | Trả về boolean (vị tự) |
| method! | Thay đổi receiver tại chỗ (bang method) |
| self.method | Định nghĩa class method |

Lớp

Định Nghĩa Lớp

```
class User
  attr_accessor :name, :email
  def initialize(name, email)
    @name = name
    @email = email
  end
end
```

Kế Thừa

```
class Admin < User
  def initialize(name, email, level)
    super(name, email)
    @level = level
  end
end
```

Kiểm Soát Truy Cập

| | |
|----------------------|----------------------------------|
| public | Mặc định; truy cập từ bất kỳ đâu |
| private | Chỉ truy cập trong lớp |
| protected | Truy cập trong lớp và lớp con |
| attr_reader | Tạo phương thức getter |
| attr_writer | Tạo phương thức setter |
| attr_accessor | Tạo cả getter và setter |

Module

Mixin

```
module Greetable
  def greet
    "Hello, I'm #{name}"
  end
end
class User; include Greetable; end
```

Namespace

```
module Payment
  class Processor
    def charge(amount) end
  end
end
p = Payment::Processor.new
```

Tham Khảo Nhanh Ruby

Include vs Extend

| | |
|------------------------|---|
| include ModName | Thêm làm instance method |
| extend ModName | Thêm làm class method |
| prepend ModName | Chèn trước lớp trong chuỗi tra cứu method |

Block & Iterator

Cú Pháp Block

```
[1, 2, 3].each { |n| puts n }      # single-line block
[1, 2, 3].each do |n|
  puts n                          # multi-line block
end
```

Yield

```
def with_logging
  puts "start"
  result = yield
  puts "end"
  result
end
with_logging { expensive_operation }
```

Proc & Lambda

```
square = Proc.new { |x| x ** 2 }
square.call(5)          # 25
double = ->(x) { x * 2 } # lambda
double.call(3)         # 6
[1, 2, 3].map(&square)  # [1, 4, 9]
```

Iterator Phổ Biến

| | |
|-------------------------------|----------------------------------|
| .each | Duyệt từng phần tử |
| .map / .collect | Biến đổi từng phần tử |
| .select / .filter | Giữ các phần tử khớp |
| .reject | Xóa các phần tử khớp |
| .reduce / .inject | Tích lũy thành giá trị đơn |
| .each_with_index | Duyệt kèm chỉ số |
| .flat_map | Map và làm phẳng một cấp |
| .any? / .all? / .none? | Kiểm tra boolean trên collection |

Regexp

Khớp Mẫu

```
"hello 42" =~ /\d+/      # 6 (match position)
"hello" =~ /\d+/        # nil (no match)
"hello".match?(/ell/)   # true
md = "age: 30".match(/(\d+)/)
md[1]                   # "30"
```

Mẫu Phổ Biến

| | |
|------------------|----------------------------|
| /^start/ | Khớp từ đầu dòng |
| /end\$/ | Khớp ở cuối dòng |
| /\d+/ | Một hoặc nhiều chữ số |
| /\w+/ | Ký tự từ |
| /\s+/ | Khoảng trắng |
| /[a-z]+/i | Không phân biệt hoa/thường |
| /(group)/ | Nhóm bắt |

Thay Thế

```
"hello world".sub(/world/, "Ruby") # first match
"aabba".gsub(/a/, "x")             # all matches: "xxbbx"
"foo bar".gsub(/(\w+)/) { $1.upcase } # "FOO BAR"
```

I/O File

Đọc & Ghi

```
content = File.read("data.txt")
lines = File.readlines("data.txt", chomp: true)
File.write("out.txt", "hello\n")
File.open("log.txt", "a") { |f| f.puts "entry" }
```

Thao Tác File

| | |
|--------------------------------|---------------------------------|
| File.exist?(path) | Kiểm tra file tồn tại |
| File.directory?(path) | Kiểm tra đường dẫn là thư mục |
| File.basename(path) | Tên file không có thư mục |
| File.extname(path) | Phần mở rộng file |
| File.size(path) | Kích thước file tính bằng bytes |
| File.delete(path) | Xóa file |
| Dir.glob('*.*rb') | Tìm file khớp mẫu |
| FileUtils.mkdir_p(path) | Tạo thư mục theo đệ quy |

CSV & JSON

```
require "json"
data = JSON.parse(File.read("data.json"))
File.write("out.json", JSON.pretty_generate(data))
require "csv"
CSV.foreach("data.csv", headers: true) { |row| puts row["name"] }
```