



Метапрогра ммирование, файлы

Методы для метапрограммирования,
операции для работы с файлами



Метод :send

```
1 class Ship
2   attr_reader :damage, :armor
3
4   def initialize(damage:, armor:)
5     @damage = damage
6     @armor = armor
7   end
8 end
9
10 class DamageUpgrade
11   def initialize(ship, up_damage)
12     @ship = ship
13     @up_damage = up_damage
14   end
15
16   def damage
17     @up_damage + @ship.damage
18   end
19
20   def armor
21     @ship.armor
22   end
23 end
24
25 def ship_specs(ship)
26   puts "Ship damage: #{ship.damage}, armor: #{ship.armor}"
27 end
28
29 ship = Ship.send(:new, damage: 1, armor: 1)
30 upgraded_ship = DamageUpgrade.send(:new, Ship.new(damage: 1, armor: 1), 9)
31
32 # send(:ship_specs) #=> wrong number of arguments (given 0, expected 1) (ArgumentError)
33
34 send(:ship_specs, ship) #=> Ship damage: 1, armor: 1
35 send(:ship_specs, upgraded_ship) #=> Ship damage: 10, armor: 1
```

```
3.1.1 :001 > 1.send(:even?)
=> false
3.1.1 :002 > Math.send(:atan)
(irb):2:in `atan': wrong number of arguments
from (irb):2:in `<main>'
from /Users/alex/.rvm/rubies/ruby-3.1.1/bin/irb:12:in `load'
from /Users/alex/.rvm/rubies/ruby-3.1.1/bin/irb:12:in `load'
from /Users/alex/.rvm/rubies/ruby-3.1.1/bin/irb:12:in `load'
3.1.1 :003 > Math.send(:atan, 1)
=> 0.7853981633974483
```

```
3.1.1 :001 > Object.send(:new)
=> #<Object:0x000000010312a080>
3.1.1 :002 > BasicObject.send(:new)
(Object doesn't support #inspect)
=> attr_reader :damage, :armor
3.1.1 :003 > Object.send(:new).send(:to_s)
=> "#<Object:0x0000000106f66a88>"
3.1.1 :004 > BasicObject.send(:new).send(:to_s)
(irb):4:in `<main>': undefined method `send' for
from /Users/alex/.rvm/rubies/ruby-3.1.1/bin/irb:12:in `load'
from /Users/alex/.rvm/rubies/ruby-3.1.1/bin/irb:12:in `load'
from /Users/alex/.rvm/rubies/ruby-3.1.1/bin/irb:12:in `load'
3.1.1 :005 > false.send(:!)
=> true
3.1.1 :006 > nil.send(:!).send(:!) == !!nil
=> true
3.1.1 :007 > -10.send(:+@)
=> -10
3.1.1 :008 > -10.send(:-@)
=> 10
```

Метод :public_send

```
10 class DamageUpgrade
11   def initialize(ship, up_damage)
12     @ship = ship
13     @up_damage = up_damage
14   end
15
16   def damage
17     @up_damage + original_damage
18   end
19
20   def armor
21     @ship.armor
22   end
23
24   private
25
26   def original_damage
27     @ship.damage
28   end
29 end
30
31 def ship_specs(ship)
32   puts "Ship damage: #{ship.damage}, armor: #{ship.armor}"
33 end
34
35 ship = DamageUpgrade.new(Ship.send(:new, damage: 1, armor: 1), 9)
36
37 puts ship.send(:damage) #=> 10
38 puts ship.send(:original_damage) #=> 1
39
40 puts ship.public_send(:damage) #=> 10
41 puts ship.public_send(:original_damage) #=> private method `original_damage' called
```

Метод :respond_to?

```
1 class Duck
2   def name
3     'duck'
4   end
5
6   def quack
7     'quack'
8   end
9 end
10
11 class Goose
12   def honk
13     'honk'
14   end
15 end
16
17 def make_noise(quackable_object)
18   return unless quackable_object.respond_to?(:quack)
19
20   puts "#{quackable_object.name} makes #{quackable_object.quack}"
21 end
22
23 duck = Duck.new
24 goose = Goose.new
25
26 make_noise(duck) #=> duck makes quack
27 make_noise(goose) # nothing will return
```

```
→ Documents irb
3.1.1 :001 > Object.respond_to?(:new)
=> true
3.1.1 :002 > BasicObject.respond_to?(:new)
=> true
3.1.1 :003 > Object.new.respond_to?(:to_s)
=> true
3.1.1 :004 > BasicObject.new.respond_to?(:to_s)
(irb):4:in `<main>': undefined method `respond_to?'
          from /Users/alex/.rvm/rubies/ruby-3.1.1/lib
          from /Users/alex/.rvm/rubies/ruby-3.1.1/bin
```

```
3.1.1 :001 > class A, include_all = f
3.1.1 :002 >   def a1
3.1.1 :003 >   end
3.1.1 :004 >   if obj responds to the given n
3.1.1 :005 >   protected
3.1.1 :006 >
3.1.1 :007 >   def a2
3.1.1 :008 >   end
3.1.1 :009 >
3.1.1 :010 >   private respond_to_m
3.1.1 :011 >
3.1.1 :012 >   def a3 parameter is given a
3.1.1 :013 >   end
3.1.1 :014 >   end
=> :a3
3.1.1 :015 > A.new.respond_to?(:a1) include_a
=> true
3.1.1 :016 > A.new.respond_to?(:a2)
=> false
3.1.1 :017 > A.new.respond_to?(:a3)
=> false
3.1.1 :018 > A.new.respond_to?(:a3, true)
=> true
3.1.1 :019 > A.new.respond_to?(:a4, true)
=> false
```

Метод :method_missing

```
1  class RandomClass
2  end
3
4  # RandomClass.new.some_method #=> undefined method `some_method'
5
6  class RandomClass
7    def method_missing(method_name, *args, &block) # args and block are optional
8      | puts "Tried to call #{method_name} with args: #{args} and block: #{block_given?}"
9    end
10 end
11
12 RandomClass.new.some_method(1, a: 2) { |a| a * 2 }
13 #=> Tried to call some_method with args: [1, {:a=>2}] and block: true
14 RandomClass.new.some_method
15 #=> Tried to call some_method with args: [] and block: false
16
17 class NestedRandomClass < RandomClass
18 end
19
20 NestedRandomClass.new.some_method
21 #=> Tried to call some_method with args: [] and block: false
22
23 class NestedRandomClass < RandomClass
24   private
25
26   def some_private_method
27     | puts 'It is a private method'
28   end
29
30   def method_missing(method_name)
31     | if private_methods.include?(method_name)
32       |   send(method_name)
33       |   return
34     end
35
36     super
37   end
38 end
39
40 NestedRandomClass.new.some_private_method #=> It is a private method
```

Использования :method_missing

- для делегирования

```
1 class Ship
2   attr_reader :damage, :armor
3
4   def initialize(damage:, armor:)
5     @damage = damage
6     @armor = armor
7   end
8 end
9
10 class DamageUpgrade
11   def initialize(ship, up_damage)
12     @ship = ship
13     @up_damage = up_damage
14   end
15
16   def damage
17     @up_damage + @ship.damage
18   end
19
20   private
21
22   def method_missing(method_name)
23     @ship.public_send(method_name)
24   end
25 end
26
27 puts DamageUpgrade.new(Ship.new(damage: 2, armor: 2), 4).armor #=> 2
```

```
1 require 'delegate'
2
3 class Ship
4   attr_reader :damage, :armor
5
6   def initialize(damage:, armor:)
7     @damage = damage
8     @armor = armor
9   end
10 end
11
12 class DamageUpgrade < SimpleDelegator
13   def initialize(ship, up_damage)
14     @ship = ship
15     @up_damage = up_damage
16     super(ship)
17   end
18
19   def damage
20     @up_damage + @ship.damage
21   end
22 end
23
24 puts DamageUpgrade.new(Ship.new(damage: 2, armor: 2), 4).armor #=> 2
```

Использования :method_missing

- для создания domain-specific language (DSL)

```
1 class StringBuilder
2   attr_reader :result
3
4   def initialize(&block)
5     @result = ''
6
7     block.call(self)
8   end
9
10  private
11
12  def method_missing(method_name, *args)
13    @result += args.empty? ? "#{method_name}; " : "#{method_name}: #{args.first}; "
14  end
15 end
16
17 builder = StringBuilder.new do |str|
18   str.this_is_string('yes')
19   str.it_is_magic_string('yes')
20   str.simple_key
21 end
22
23 puts builder.result #=> this_is_string: yes; it_is_magic_string: yes; simple_key;
```

```
class Person
  # ... Class Definition ... #
  def to_builder
    Jbuilder.new do |person|
      person(self, :name, :age)
    end
  end
end

class Company
  # ... Class Definition ... #
  def to_builder
    Jbuilder.new do |company|
      company.name name
      company.president president.to_builder
    end
  end
end

company = Company.new('Doodle Corp', Person.new('John Stobs', 58))
company.to_builder.target!

# => {"name":"Doodle Corp","president":{"name":"John Stobs","age":58}}
```

* gem <https://github.com/rails/jbuilder>

Файлы. Чтение из файла

```
1  file = File.open('test_file')
2  p file.read #=> "line 1 in this file\nline 2 in this file\nline 3 in this file\n\n"
3
4  file.close
5  # file.read #=> closed stream (IOError)
6
7  file = File.new('test_file', 'r')
8  p file.read #=> "line 1 in this file\nline 2 in this file\nline 3 in this file\n\n"
9
10 # p file.readline #=> end of file reached (EOFError)
11 file.rewind
12 p file.readline #=> "line 1 in this file\n"
13 p file.readline #=> "line 2 in this file\n"
14
15 file.rewind
16 p file.readlines #=> ["line 1 in this file\n", "line 2 in this file\n", "line 3 in this file\n", "\n"]
17 file.close
18
19 # Alternative to open and close methods
20 p File.read('test_file').class #=> String
21
22 File.foreach('test_file') { |line| p line } #=> "line 1 in this file\n", "line 2 in this file\n", ...
23
24 p File.binread('test_file', 6, 20) #=> "line 2"
25
26 p File.read('test_file_2') #=> "Линия 1 файла\nЛиния 2 файла\nЛиния 3 файла\n\n"
27 p File.read('test_file_2', 2) #=> "\xD0\x9B"
28 p File.read('test_file_2', 2).force_encoding('UTF-8') #=> Л
```


Файлы. Запись в файл

```
1  File.open('test_file', 'w') { |file| file.write("#{Time.now} - new line\n") }
2
3  p File.read('test_file') #=> "2022-08-08 13:45:36 +0300 - new line\n"
4
5  File.write('test_file', "#{Time.now} - new second line\n", mode: 'a')
6
7  p File.read('test_file') #=> "2022-08-08 13:45:36 +0300 - new line\n2022-08-08 13:45:36 +0300 - new second line\n"
8
9  # "r"  Read-only, starts at beginning of file  (default mode).
10
11 # "r+" Read-write, starts at beginning of file.
12
13 # "w"  Write-only, truncates existing file
14 #      to zero length or creates a new file for writing.
15
16 # "w+" Read-write, truncates existing file to zero length
17 #      or creates a new file for reading and writing.
18
19 # "a"  Write-only, each write call appends data at end of file.
20 #      Creates a new file for writing if file does not exist.
21
22 # "a+" Read-write, each write call appends data at end of file.
23 #      Creates a new file for reading and writing if file does
24 #      not exist.
```

ПОТОК ВВОДА/ВЫВОДА

```
1  p $stdout.class, $stdout.tty? #=> IO, true # (teletypewriter - console)
2
3  $stdout.puts 'Print smth' #=> Print smth
4  $stdout << "Pring smth too\n" #=> Print smth too
5
6  # $stdout.gets #=> not opened for reading (IOError)
7
8  $stdin.gets(3) # gets 3 characters from input stream
9
10 # $stdin << 'fd' #=> not opened for writing (IOError)
11
12 p File.superclass #=> IO
13
14 read_file = File.open('test_file', 'r')
15 p read_file.tty? #=> false
16
17 p read_file.gets(7) #=> "2022-08"
18 read_file.close
19
20 write_file = File.open('test_file', 'a+')
21 p write_file.gets(7) #=> "2022-08"
22
23 write_file.puts 'Print smth'
24 write_file << "Print smth too \n"
25
26 write_file.rewind
27 p write_file.readlines
28 #=> ["2022-08-08 16:45:36 +0400 - new line\n", "2022-08-08 16:45:36 +0400 - new second line\n", "Print smth\n", "Print smth too \n"]
```

Свой tty поток

```
3.1.1 :002 > $stdout.close
/Users/alex/.rvm/rubies/ruby-3.1.1/lib/ruby/3.1.0/irb.rb:681:in `write': closed stream (IOError)
    from /Users/alex/.rvm/rubies/ruby-3.1.1/lib/ruby/3.1.0/irb.rb:681:in `puts'
    from /Users/alex/.rvm/rubies/ruby-3.1.1/lib/ruby/3.1.0/irb.rb:681:in `puts'
    from /Users/alex/.rvm/rubies/ruby-3.1.1/lib/ruby/3.1.0/irb.rb:681:in `handle_exception'
    from /Users/alex/.rvm/rubies/ruby-3.1.1/lib/ruby/3.1.0/irb.rb:596:in `block (2 levels) in
    from /Users/alex/.rvm/rubies/ruby-3.1.1/lib/ruby/3.1.0/irb.rb:770:in `signal_status'
    from /Users/alex/.rvm/rubies/ruby-3.1.1/lib/ruby/3.1.0/irb.rb:558:in `block in eval_input'
    from /Users/alex/.rvm/rubies/ruby-3.1.1/lib/ruby/3.1.0/irb/ruby-lex.rb:268:in `block (2 l
    from /Users/alex/.rvm/rubies/ruby-3.1.1/lib/ruby/3.1.0/irb/ruby-lex.rb:250:in `loop'
    from /Users/alex/.rvm/rubies/ruby-3.1.1/lib/ruby/3.1.0/irb/ruby-lex.rb:250:in `block in e
    from /Users/alex/.rvm/rubies/ruby-3.1.1/lib/ruby/3.1.0/irb/ruby-lex.rb:249:in `catch'
    from /Users/alex/.rvm/rubies/ruby-3.1.1/lib/ruby/3.1.0/irb/ruby-lex.rb:249:in `each_top_l
    from /Users/alex/.rvm/rubies/ruby-3.1.1/lib/ruby/3.1.0/irb.rb:557:in `eval_input'
    from /Users/alex/.rvm/rubies/ruby-3.1.1/lib/ruby/3.1.0/irb.rb:491:in `block in run'
    from /Users/alex/.rvm/rubies/ruby-3.1.1/lib/ruby/3.1.0/irb.rb:490:in `catch'
    from /Users/alex/.rvm/rubies/ruby-3.1.1/lib/ruby/3.1.0/irb.rb:490:in `run'
    from /Users/alex/.rvm/rubies/ruby-3.1.1/lib/ruby/3.1.0/irb.rb:419:in `start'
    from /Users/alex/.rvm/rubies/ruby-3.1.1/lib/ruby/gems/3.1.0/gems/irb-1.4.1/exe/irb:11:in
    from /Users/alex/.rvm/rubies/ruby-3.1.1/bin/irb:25:in `load'
    from /Users/alex/.rvm/rubies/ruby-3.1.1/bin/irb:25:in `'
→ Documents
```

```
1 file = File.new('test_file', 'r')
2 p file.tty? #=> false
3
4 io = File.new('/dev/tty', 'w')
5 p io.tty? #=> true
6
7 io.puts 'This is console' #=> This is console
8
9 io.close
10 file.close
```

Полезные ссылки

<https://betterprogramming.pub/metaprogramming-with-ruby-send-and-public-send-methods-f85aa8f91366> – Примеры работы методов `send` и `public_send`;

<https://medium.com/podiihq/method-missing-in-ruby-af4c6edd5130> – Описание работы метода `method_missing`;

<https://www.rubyguides.com/2015/05/working-with-files-ruby/> – Примеры работы с файлом и полезные методы для работы с файлом;

<https://ruby-doc.org/core-2.5.0/IO.html> – Официальная документация по классу `IO`.

Конец! Спасибо!