

2018年12月1日

Ruby 初級者向けレッスン 68回
— クラス —

ひがき @ **Ruby** 関西

クラスとは

- オブジェクトの雛形
- 属性と振舞いを定義
 - カプセル化 (データ・実装の隠蔽)
- **Ruby** はクラスベースのオブジェクト指向言語

クラスを作ってみよう

```
class Person; end
```

```
obj = Person.new
```

```
# => #<Person:0x007fb1a184a198>
```

```
obj.class          # => Person
```

```
Person.superclass  # => Object
```

属性を持たせてみよう

```
class Person
  def initialize(name)
    @name = name
  end
end
```

```
matz = Person.new('matz')
# => #<Person:0x007fb1a1848438
      @name="matz">
```

属性にアクセスしてみよう

- インスタンス変数は外から見えない

```
class Person
  attr_reader :name
end
```

```
matz.name          # => "matz"
```

変数・定数のおさらい

ローカル変数	p erson
インスタンス変数	@ person
クラス変数	@@ person
グローバル変数	\$ person
定数	P erson

属性を増やしてみよう

```
class Person
  def initialize(name, born = nil)
    @name, @born = name, born
  end
  attr_accessor :born
end
```

```
matz.methods.map(&:to_s).grep(/born/)
# => ["born", "born="]
```

属性を増やしてみよう (2)

```
matz.born = Time.local(1965, 4, 14)
dhh = Person.new('dhh', DateTime.new(
  1979, 10, 15, 0, 0, 0, "+01:00"))
```

```
matz.born
# => 1965-04-14 00:00:00 +0900
```

```
dhh.born.to_time
# => 1979-10-15 00:00:00 +0100
```


メソッドを作ってみよう

```
class Person
  def age
    (Time.now.strftime('%Y%m%d').to_i -
     @born.strftime('%Y%m%d').to_i) /
    1_00_00
  end
end

matz.age      # => 53
dhh.age       # => 39
```

メソッドを上書きしてみよう

```
matz.to_s    # => "#<Person:0x007fb1a184843"
```

```
class Person
  def to_s
    "#{@name}({age})"
  end
end
```

```
matz.to_s    # => "matz(53)"
```

```
dhh.to_s     # => "dhh(39)"
```

インスタンスを比較してみると…

```
person = Marshal.load(Marshal.dump matz)  
# => matz(53)
```

```
person == dhh    # => false
```

```
person == matz   # => false # おかしい
```

```
person.method(:==)
```

```
# => #< Method: Person(BasicObject)#== >
```

順序を決めよう

```
class Person
  include Comparable
  def <=> o
    @name <=> o.name
  end
end

person == dhh      # => false
person == matz     # => true
matz > dhh         # => true
```

Array#sort してみよう

```
people = [matz, dhh]
```

```
people.sort # => [dhh(39), matz(53)]
```

年齢順に sort してみよう

若い順

```
people.sort_by(&:age)  
# => [dhh(39), matz(53)]
```

年功序列

```
people.sort{|a, b| b.age <=> a.age}  
# => [matz(53), dhh(39)]
```

```
people.sort_by{|p| -p.age}  
people.sort_by(&:born.to_time)
```

Hash のキーにしてみると……

```
h = {matz => "Ruby", dhh => "Rails"}
```

```
h[matz]      # => "Ruby"
```

```
h[dhh]       # => "Rails"
```

```
key = Marshal.load(Marshal.dump matz)
```

```
key == matz  # => true
```

```
h[key]       # => nil # おかしい
```

hash 値を計算しよう

```
class Person
```

```
  def hash
```

```
    [@name, @born].hash
```

```
  end
```

```
end
```

```
matz.hash      # => -2004926503290248919
```

```
dhh.hash       # =>  2713023373120596011
```

```
key.hash       # => -2004926503290248919
```


eq1? を上書きしよう

```
class Person
  def eq1? o
    [@name, @born].eq1? [o.name, o.born]
  end
end
```

```
key.eq1? matz      # => true
key.eq1? dhh       # => false
```

Hash にアクセスしてみよう

```
h = {matz => "Ruby", dhh => "Rails"}
```

```
h[matz]      # => "Ruby"
```

```
h[dhh]       # => "Rails"
```

```
h[key]       # => "Ruby"
```

等値性のおさらい

`==` 内容が等しいか?

`===` case 式で使用

`eql?` Hash クラスが使用

`equal?` 同一オブジェクトか?

アクセス制御してみよう

- public
- protected
- private

```
class Person
  protected :born
end
```

```
matz.born
```

```
# ~> protected method 'born' called for  
matz(49):Person (NoMethodError)
```

Struct でクラスを作る

```
Person = Struct.new(:name, :born)
```

```
matz = Person.new('matz')
```

```
dhh = Person.new('dhh',  
                 Time.local(1979, 10, 15))
```

```
person = Marshal.load(Marshal.dump matz)
```

```
person == matz # => true
```

```
person == dhh  # => false
```

Struct なら等値性も

```
h = {matz => "Ruby", dhh => "Rails"}
```

```
key = Marshal.load(Marshal.dump matz)
```

```
h[matz]      # => "Ruby"
```

```
h[dhh]       # => "Rails"
```

```
h[key]       # => "Ruby"
```

Struct でもメソッドを

```
class Person
  def age
    (Time.now.strftime('%Y%m%d').to_i -
     born.strftime('%Y%m%d').to_i) /
    1_00_00
  end
end

dhh.age      # => 35
```

今日 話さなかったこと

- 委譲と継承
- モジュール
- クラス変数・クラスメソッド
- Range の始点と終点

まとめ

- クラスの作り方
 - 属性・アクセサ
 - メソッド
- オブジェクトの等値性

演習問題 0

今日のレッスンで分からなかったこと、疑問に思ったことをグループで話し合ってみよう。

演習問題 1

属性として身長と体重を追加しよう。体重は秘密にしよう。

演習問題 2

BMI を計算するメソッドを追加しよう。

$$BMI = \frac{w}{t^2}$$

$$w = \text{体重 [kg]}$$

$$t = \text{身長 [m]}$$

演習問題 3

Person#<=> を書き直そう。

その妥当な仕様は？

```
p0 = Person.new('matz')
```

```
p1 = Person.new('Matz',  
                Time.local(1965, 4, 14))
```

```
p0 <=> p1    # => ?
```

演習問題 4

Person#=== を定義しよう。

その妥当な仕様は？

```
p0 = Person.new('matz')
```

```
p1 = Person.new('Matz',  
                Time.local(1965, 4, 14))
```

```
p0 === p1    # => ?
```

自己紹介

- 名前 (ニックネーム)
- 普段の仕事・研究内容・代表作
- Ruby 歴・コンピュータ歴
- 勉強会に来た目的
- などなど

参考

- 公式サイト

`https://www.ruby-lang.org/`

- るりま

`http://docs.ruby-lang.org/ja/`

- 解答例

`https://github.com/higaki/`

`learn_ruby_kansai_84`