

2012年7月21日

---

**Ruby** 初級者向けレッスン 43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 (属性・振舞いの隠蔽)
- **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:0x10102718>
```

```
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:0x10138598 @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	<code>person</code>
インスタンス変数	<code>@person</code>
クラス変数	<code>@@person</code>
グローバル変数	<code>\$person</code>
定数	<code>Person</code>

# 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',
                 Time.local(1979, 10, 15))
```

```
matz.born
```

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

```
dhh.born
```

```
# => 1979-10-15 00:00:00 +0900
```

# 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) /
    10000
  end
end

matz.age      # => 47
dhh.age       # => 32
```

# メソッドを上書きしてみよう

---

```
matz.to_s # => "#<Person:0x10138598>"
```

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

```
matz.to_s # => "matz(47)"
```

```
dhh.to_s # => "dhh(32)"
```

# インスタンスを比較してみると…

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

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

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

# 順序を決めよう

---

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

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

# Array#sort してみよう

---

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

```
people.sort # => [dhh(32), matz(47)]
```

# 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
    code = 17
    code = 37 * code + @name.hash
    code = 37 * code + @born.hash
  end
end

matz.hash      # => -22068619118
dhh.hash       # => 14923733106
```

# eql? を上書きしよう

```
class Person
  def eql? o
    return false unless @name.eql? o.name
    return false unless @born.eql? o.born
    true
  end
end

key.eql? matz      # => true
key.eql? 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 ma
```

# 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

---

クラス Person には、いくつかのバグがある。  
それを見つけて出して修正しよう。

# 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