

2013年11月9日

Ruby 初級者向けレッスン @KOF2013

ひがき @ Ruby 関西

Ruby とは

- まつもとゆきひろ氏が開発
- **Ruby on Rails**が有名
- スクリプト \iff コンパイル (Java)
- オブジェクト指向言語 \iff 関数型言語 (Haskell)
- クラスベース \iff プロトタイプベース (JavaScript)
- 軽量プログラミング言語

軽量プログラミング言語

Java の場合

```
class HelloWorld {  
    public static void main(String args[])  
        System.out.println("Hello World!");  
    }  
}
```

軽量プログラミング言語 (2)

Ruby の場合

```
puts "Hello World!"
```

Ruby は型のない言語?

```
i = 0
```

```
i.class          # => Fixnum
```

```
s = "hello"
```

```
s.class         # => String
```

主要な型 (クラス)

Fixnum	整数	Bignum	多倍長整数
Float	浮動小数点数	Class	クラス
String	文字列	Regex	正規表現
Array	配列	Hash	ハッシュ
Range	範囲	Exception	例外
Time	時刻	File	ファイル

などなど……

Rubyの変数には型がない

```
i = 0
```

```
i.class          # => Fixnum
```

```
i = "hello"
```

```
i.class          # => String
```

すべてがオブジェクト

```
0.next           # => 1
1.to_s           # => "1"

"hello".upcase   # => "HELLO"

2.times{puts "こんにちは"}
# >> こんにちは
# >> こんにちは
```


nil (NULL) もオブジェクト

```
nil.to_s          # => ""
nil.to_i          # => 0
nil.nil?          # => true
nil / 2           # => undefined method '/'
                  for nil:NilClass (NoMethodError)
```

真偽値 (条件式)

- false または nil は偽
- それ以外は真

true 1 0 "" [] { } Infinity NaN

繰り返し

```
a = [1, 2, 3]
a.each do |i|
  puts i
end
```

```
# >> 1
```

```
# >> 2
```

```
# >> 3
```

繰り返し

(2)

```
a = [1, 2, 3]
a.each{|i| puts i}
```

```
# >> 1
```

```
# >> 2
```

```
# >> 3
```

ブロック付きメソッドいろいろ

```
a = [0, 1, 2, 3]
```

```
a.map{|i| i * 2}          # => [0, 2, 4, 6]
```

```
a.select{|i| i.even?}    # => [0, 2]
```

```
a.inject{|s, i| s + i}   # => 6
```

```
a.find{|i| i.odd?}       # => 1
```

```
a.all?{|i| i.even?}     # => false
```

```
a.any?{|i| i.even?}     # => true
```

演習問題 1

- 1 から順番に数え上げる (数字を表示)
- 3 の倍数なら Fizz を表示する
- 5 の倍数なら Buzz を表示する
- 3 と 5 の倍数なら FizzBuzz を表示する

1, 2, Fizz, 4, Buzz, Fizz, 7, 8, Fizz, Buzz, 11,
Fizz, 13, 14, FizzBuzz, 16, 17, Fizz, 19, Buzz,
Fizz, 22, 23, Fizz, Buzz, 26, Fizz, 28, 29,
FizzBuzz

簡単なところから

- 1 から 30 まで順番に数え上げる (数字を表示)

```
numbers = 1..30  
numbers.each{|i| puts i}
```

```
numbers = (1..30).to_a  
puts numbers
```

3 の倍数なら Fizz を返すメソッド

```
def fizz_buzz(n)
  if n % 3 == 0
    "Fizz"
  else
    n
  end
end
```


メソッドを使ってみる

```
fizz_buzz(3)      # => "Fizz"
```

```
fizz_buzz(4)      # => 4
```

```
fizz_buzz(5)      # => 5
```

```
fizz_buzz(6)      # => "Fizz"
```

演習問題 2

ジャンケンの勝敗を判定しよう。

グーは 0、チョキは 1、パーは 2 とする。

勝ちなら 1、負けなら -1、引き分けなら 0 を返すメソッドを作ろう。

```
def judge(you, computer)
  ...
end
```

演習問題 1 — 解答例 1

```
(1..30).each do |i|  
  case  
  when i % 15 == 0 then puts "FizzBuzz"  
  when i % 5 == 0 then puts "Buzz"  
  when i % 3 == 0 then puts "Fizz"  
  else puts i  
  end  
end
```

演習問題 1 — 解答例 2

```
a = ["0".."30"]
```

```
re = /\d*$/
```

```
3.step(30, 3){|i| a[i].sub!(re, "Fizz")}
```

```
5.step(30, 5){|i| a[i].sub!(re, "Buzz")}
```

```
puts a[1..-1]
```

演習問題 1 — 解答例 3

```
(1..30).each do |i|  
  a = [[i, "Fizz"], ["Buzz", "FizzBuzz"]]  
  i5 = i % 5 == 0 ? 1 : 0  
  i3 = i % 3 == 0 ? 1 : 0  
  puts a[i5][i3]  
end
```

演習問題 2 — 解答例 1

```
def judge(you, computer)
  case [you, computer]
  when [0, 1], [1, 2], [2, 0] then 1
  when [0, 0], [1, 1], [2, 2] then 0
  when [0, 2], [1, 0], [2, 1] then -1
  end
end
```

演習問題 2 — 解答例 2

```
def judge(you, computer)
  case (computer - you) % 3
  when 0 then 0
  when 1 then 1
  when 2 then -1
  end
end
```

参考文献

- たのしい **Ruby** 第4版

ISBN 9784797372274

- プログラミング言語 **Ruby**

ISBN 9784873113944

- パーフェクト **Ruby**

ISBN 9784774158792

参考 URL

- **Ruby** 公式サイト

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

- **Ruby** 関西

`https://github.com/rubykansai/`

`workshops/wiki`

その他

- **この資料**

`https://k-of.jp/2013/sites/all/files/slides/RubyLessonKOF2013.pdf`

- **twitter**

@rubykansai

#rubykansai