

Ruby 初級者向けレッスン第 17 回 演習課題解答例

okkez @ Ruby 関西, チカホリ@小波ゼミ

2008 年 02 月 16 日

演習問題

以下の問題では for, while など使用禁止です。

九九の表

九九の表を作成しよう。

以下のような出力を得られるプログラムを作成してください。

```
| 1 2 3 4 5 6 7 8 9
--+-----
1| 1 2 3 4 5 6 7 8 9
2| 2 4 6 8 10 12 14 16 18
3| 3 6 9 12 15 18 21 24 27
4| 4 8 12 16 20 24 28 32 36
5| 5 10 15 20 25 30 35 40 45
6| 6 12 18 24 30 36 42 48 54
7| 7 14 21 28 35 42 49 56 63
8| 8 16 24 32 40 48 56 64 72
9| 9 18 27 36 45 54 63 72 81
```

解答例

```
01: #! /usr/bin/ruby
02:
03: def main
04:   arr = (1..9).map{|n| (1..9).map{|m| n * m }}
05:   arr.each_with_index do |line, idx|
06:     if idx == 0
07:       header = ' |' + (1..9).map{|v| '%3d' % v }.join(' ')
08:       header << "\n--+#{'-' * (header.size - 3)}"
09:       puts header
10:     end
```

```

11:     puts '%2d|' % (idx+1) + line.map{|v| '%3d' % v }.join('')
12:   end
13: end
14:
15: if __FILE__ == $0
16:   main
17: end

```

方針としてはわかりやすさ重視。必要以上に短いコードにしない。
map して *join* してるところは *inject* に置換可能。

line.4

まずは九九の表をさっくりと作成。

line.7

ヘッダー部分の出力。 *String#%* メソッドを使用。

100 マス計算

100 マス計算の問題と解答を作成しよう。

以下のような出力を得られるプログラムを作成してください。

```

+| 8 2 1 4 5 0 3 7 6 9
-----
8|
2|
0|
9|
4|
7|
6|
3|
5|
1|

+| 8 2 1 4 5 0 3 7 6 9
-----
8| 16 10 9 12 13 8 11 15 14 17
2| 10 4 3 6 7 2 5 9 8 11
0| 8 2 1 4 5 0 3 7 6 9
9| 17 11 10 13 14 9 12 16 15 18
4| 12 6 5 8 9 4 7 11 10 13
7| 15 9 8 11 12 7 10 14 13 16
6| 14 8 7 10 11 6 9 13 12 15

```

```
3| 11 5 4 7 8 3 6 10 9 12
5| 13 7 6 9 10 5 8 12 11 14
1| 9 3 2 5 6 1 4 8 7 10
```

- 計算の種類は左上の隅に出力する
- 計算の種類は四則 (加減乗除) とする (+, -, *, /)
- 除算の場合はゼロ割の可能性があるので考慮すること
- 出力される結果は毎回異なる (ランダム) ものとする

解答例

```
01: #! /usr/bin/ruby
02:
03: OPERATORS = %w[+ - * /]
04:
05: if RUBY_VERSION < '1.9.0'
06:   class Array
07:     def shuffle
08:       sort_by{ rand }
09:     end
10:
11:     def choice
12:       at(rand(size))
13:     end
14:   end
15: end
16:
17: def main
18:   base_array = (0..9).to_a
19:   ns = base_array.shuffle
20:   ms = base_array.shuffle
21:   answer = create_matrix(ns, ms, operator)
22:   question = answer.gsub(/([0-9]\|)([0-9X-]+$/m, '\\1')
23:   puts question
24:   puts ''
25:   puts answer
26: end
27:
28: def operator
29:   OPERATORS.choice
30: end
31:
```

```

32: def create_matrix(ns, ms, op)
33:   arr = ns.map{|n| ms.map{|m| calc(n, m, op)}}
34:   header = " #{op}|#{ms.map{|v| '%3d' % v }.join('')}"
35:   header << "\n--+#{'-' * (header.size - 3)}"
36:   body = []
37:   arr.each_with_index{|line,idx|
38:     body << " #{'%2d|' % ns[idx]}#{line.map{|v| '%3s' % v }.join('')}"
39:   }
40:   "#{header}\n#{body.join("\n")}"
41: end
42:
43: def calc(n, m, op)
44:   n.__send__(op, m)
45: rescue ZeroDivisionError
46:   'X'
47: end
48:
49: if __FILE__ == $0
50:   main
51: end

```

方針は九九の表で使った考え方を応用することと、1.8, 1.9 両対応にすること。

line.3

使用可能な演算子のリスト

line.5 - 10

1.8 用に Array クラスに shuffle, choice を追加。

line.21 - 22

先に答えを作成してから、問題を作成する。

line.43 - 47

二つの数値と一つの演算子を受け取って計算結果を返すメソッドを定義。ゼロ割を意識する
 とこうするのが良さそう。

色見本作成

rgb.txt を使用して HTML の色見本を作成してください。自分の PC に rgb.txt が入っている
 人はそれを使用してください。

なお、valid な html である必要はありません。

以下に rgb.txt の最初の数行を示します。

```

! $Xorg: rgb.txt,v 1.3 2000/08/17 19:54:00 cpqblld Exp $
255 250 250 snow

```

```
248 248 255 ghost white
248 248 255 GhostWhite
245 245 245 white smoke
```

解答例

```
01: #! /usr/bin/ruby
02:
03: class RGBLine
04:   attr_accessor :rgb, :name
05:
06:   def initialize(line)
07:     red, green, blue, @name = *line.strip.split(/\s+/, 4)
08:     @rgb = [red, green, blue]
09:   end
10:
11:   def html
12:     "<td style=\"background-color:#{hex_color}\"><span style=\"color:#{hex_reversed_color}\">"
13:   end
14:
15:   def hex_color
16:     "##{rgb.map{|v| '%02x' % v.to_i }.join('')}"
17:   end
18:
19:   def hex_reversed_color
20:     "##{rgb.map{|v| '%02x' % (255 - v.to_i) }.join('')}"
21:   end
22:
23:   def hash
24:     rgb.join('').hash
25:   end
26:
27:   def eql?(other)
28:     hash == other.hash
29:   end
30: end
31:
32: def main
33:   lines = File.readlines('rgb.txt').reject {|line| /^!/ =~ line }
34:   hash = Hash.new{|hash, key| hash[key] = [] }
35:   lines.map{|line| RGBLine.new(line) }.uniq.map{|v| v.html }.each_with_index do |c, idx|
36:     hash[idx.div(10)] << c
37:   end
```

```

38:   arr = hash.sort_by{|v| v[0] }.map{|v| v.last }
39:   puts "<table>"
40:   puts arr.map{|v| "<tr>#{v.join(',')}</tr>" }.join("\n")
41:   puts "</table>"
42: end
43:
44: if __FILE__ == $0
45:   main
46: end

```

方針はブラウザで表示確認出来る程度にちゃんとした html を出力すること。

line.3 - 30

クラスを作成。この様にクラスを作成しておくとメソッドを追加するだけで様々な出力フォーマットに対応することが可能になります。

line.7

String#split の第二引数を指定することで分割する上限を決めることができます。*String#split* で 4 つに分割したものを 4 つの変数に代入しています。

line.11 - 21

html に出力するために必要なメソッド。html 上で色は *#FFFFFF* のように表現されるので、そのような出力が得られるようにしています。また *hex_reversed_color* メソッドでは反対色 (補色) を計算しています。こうすることによってそこそこ見やすい出力になるように工夫しています。

line.23 - 29

hash と *eql?* を再定義しているのは、line.35 のメソッドチェーンの中で *Array#uniq* を使用しているためです。

line.34

ブロック付きの *Hash.new* でハッシュの初期値を決めるようにしています。

line.35 - 38

そのまま何も考えずに実装すると、横に長いテーブルになってしまいそうだったのである程度で改行するようにしてみました。html の方を工夫すればもう少しシンプルになるかもしれません。

line.39 - 41

ここでは手抜きで標準出力に結果を出力し、リダイレクトでファイルに書くことにしています。

ヒント

リファレンスマニュアルで以下のクラスやメソッドについて調べてみてください。

- Range

- `Array#map`
- `Array#collect`
- `Array#each`
- `Array#each_with_index`
- `Array#join`
- `Array#sort_by`
- `Kernel.#rand`
- `Kernel.#printf`
- `Kernel.#sprintf`
- `Kernel.#__send__`
- `String#%`
- `String#split`