Ruby初級者向けレッスン第 22 回 演習問題解答例

okkez @ Ruby 関西, モリ@小波ゼミ

2008 年 07 月 26 日

演習問題

いろいろな計算をしてみよう

通常モードと bc モードで計算結果の違いを見てみましょう。

1. 1 + 1
2. 10 / 4
3. 100 ** 10000
4. sqrt(16 / 36)
5. BMI を計算してみましょう。BMI = 体重 (kg) / 身長 (m) ** 2
6. 1 年は何時間でしょう？
7. 10 年は何分でしょう？
8. わたしが生まれてから 9 億 3400 万秒経っているとしたら今何歳でしょう？

解答例

irb -m で起動した場合と irb という風にオプションなしで起動した場合で動作が違うことを確認できれば OK です。解答例は省略します。

irb のオプションをいろいろといじってみよう

リファレンスマニュアルを調べてみましょう。

1. 動作中、一時的に結果表示を消してみよう

```
irb(main):007:0> 1+1
=> 2                              # 計算結果が表示される
irb(main):008:0> ***********      # 何かする
irb(main):009:0> 1+1              # もう一度計算する
irb(main):010:0>                  # 何も表示されない
```

1. 動作中にプロンプトの一部を変更してみよう

全体を変更する場合は以下のようにしますが、一部だけ変更するにはどうするでしょう？

```
irb(main):010:0> conf.prompt_mode = :SIMPLE
>> conf.prompt_mode = :DEFAULT
irb(main):012:0>
```

解答例

1. conf.echo = false
2. conf.prompt_i などを変更すれば良いです。詳細はリファレンスマニュアルを参照してください。

メソッドの動作確認をしてみよう

各メソッドの使い方はリファレンスマニュアルなどで確認してください。

1. Array#zip
2. Kernel#rand, Kernel#srand
3. Array#map, Array#select などにブロックを渡さない場合にどんなオブジェクトが返されるか調べてみましょう。
4. require 'yaml' して、 Kernel#y にいろいろなオブジェクトを渡してみよう。

解答例

実際に動かしてみればわかるので省略します。

作ってみよう

irb で動作確認しながら作ってみましょう。

1. 計算 100 もどきを作りましょう。

仕様

- コンピュータが計算問題を次々に出題します
- 回答はキーボードから入力します
- 出題される計算問題は難易度を選択出来ます (最初に選ぶ)
  - かんたん：一桁同士の四則演算 (割り算なし) のみ

  - むずかしい : 四則演算 (二桁の計算もあり、割り切れない除算はなし、答えに小数は出ない)

- 合計で 100 問出題します
- 10 問ごとに何問終了したか出力します
- 最後に以下のものを出力します
  - 正解数、不正解数
  - 全問回答するのにかかった時間

実行イメージ

```
計算 100 開始
難易度を選択してください。 1:かんたん 2:むずかしい >
1:かんたん を開始します。Press Enter >
1 + 1 = [回答入力する]
正解|不正解を表示する
1 + 2 = [回答入力する]
正解|不正解を表示する
....
1 + 2 = [回答入力する]
正解|不正解を表示する
10 問突破 #=> 10 問ごとに何問終わったか出力する
....
1 + 2 = [回答入力する]
正解|不正解を表示する
100 問突破
100 問終了しました
正解    : xx 問
不正解 : xx 問
タイム : xx 分 xx 秒
```

解答例 1

出題する可能性のある問題を列挙してから、出題開始するパターン。

```
01: #!/usr/bin/ruby
02:
03: require 'readline'
04:
05: MODE = {
06:   1 => ['1:かんたん', [:+, :-, :*]],
07:   2 => ['2:むずかしい', [:+, :-, :*, :/]]
```

```
08: }
09:
10: def answer(question)
11:   answer = nil
12:   loop do
13:     answer = Readline.readline("#{question} = ")
14:     if /\A-?\d+/ =~ answer
15:        return answer = answer.to_i
16:      else
17:        puts '数値を入力してください。'
18:      end
19:   end
20: end
21:
22: def init_calc_table(mode)
23:   hash = {:+ => [], :- => [], :* => [], :/ => [] }
24:   (0...(10 ** mode)).to_a.product((0..(10 ** mode)).to_a).each do |l, r|
25:     MODE[mode][1].each do |op|
26:       hash[op] << ["#{l} #{op} #{r}",
27:                    l.__send__(op, r)] unless op == :/ and (r == 0 or l % r != 0)
28:     end
29:   end
30:   return hash
31: end
32:
33: def show_result(questions, answers, start_time, end_time)
34:   right, wrong =questions.zip(answers).partition{|q, a| q.last == a }
35:   duration = (end_time - start_time).to_i
36:   min, sec = duration.divmod(60)
37:   puts "100 問終了しました"
38:   puts "正解   : #{right.size} 問"
39:   puts "不正解 : #{wrong.size} 問"
40:   puts "タイム : #{min} 分 #{sec} 秒"
41: end
42:
43: def main
44:   puts '計算 100 開始'
45:   mode = nil
46:   loop do
47:     mode = Readline.readline('難易度を選択してください。 1:かんたん 2:むずかし
い > ')
48:     if /1|2/ =~ mode
49:       mode = mode.to_i
```

```
50:       break
51:     end
52:   end
53:   calc_table = init_calc_table(mode)
54:   operators = MODE[mode][1]
55:   print "#{MODE[mode][0]} を開始します。 Press Enter. >"
56:   STDIN.gets
57:   start_time = Time.now
58:   answers = []
59:   questions = []
60:   100.times do |n|
61:     questions << calc_table[operators.choice].choice
62:     answers << answer(questions.last[0])
63:     puts questions.last[1] == answers.last ? '正解' : '不正解'
64:     puts "#{n + 1} 問突破" if (n + 1) % 10 == 0
65:   end
66:   end_time = Time.now
67:   show_result(questions, answers, start_time, end_time)
68: end
69:
70: if __FILE__ == $0
71:   main
72: end
```

- readline ライブラリを使用して、打ち間違えたときも修正出来るようにしています。
  - readline ライブラリについてはリファレンスマニュアルを参照してください。
- 「むずかしい」の場合で最大 40000(= 100 * 100 * 4) 通りしか問題は出題されないのであらかじめ計算してみました。
- Ruby1.8.7 以降で追加されたメソッドを使用しています。
- ユーザの入力を求める部分で不正な値を入力された場合は再入力を求めるようにしています。

解答例 2

毎回問題を生成するパターン。

```
01: #!/usr/bin/ruby
02:
03: require 'readline'
04:
05: MODE = {
06:   1 => ['1:かんたん',  [:+, :-, :*]],
```

```
07:   2 => ['2:むずかしい', [:+, :-, :*, :/]]
08: }
09:
10: def question(mode, op)
11:   begin
12:     left = rand(10 ** mode)
13:     right = rand(10 ** mode)
14:   end while op == :/ and (right == 0 or left.__send__(:%, right) != 0)
15:   formula = "#{left} #{op} #{right}"
16:   answer = left.__send__(op, right)
17:   [formula, answer]
18: end
19:
20: def answer(formula, num)
21:   loop do
22:     buf = Readline.readline("#{formula} = ").chomp
23:     return num == buf.to_i if /\A-?\d+/ =~ buf
24:     puts '数値を入力してください。'
25:   end
26: end
27:
28: def show_result(right_count, wrong_count, start_time, end_time)
29:   min, sec = (end_time - start_time).divmod(60)
30:   puts <<EOD
31: 100 問終了しました
32: 正解   : #{right_count} 問
33: 不正解 : #{wrong_count} 問
34: タイム : #{min} 分 #{sec} 秒
35: EOD
36: end
37:
38: def main
39:   puts '計算 100 開始'
40:   mode = nil
41:   loop do
42:     mode = Readline.readline('難易度を選択してください。 1:かんたん 2:むずかし
い > ')
43:     if /1|2/ =~ mode
44:       mode = mode.to_i
45:       break
46:     end
47:   end
48:   operators = MODE[mode][1]
```

```
49:   print "#{MODE[mode][0]} をはじめます Press Enter > "
50:   STDIN.gets
51:   right_count = wrong_count = 0
52:   start_time = Time.now.to_i
53:   100.times do |n|
54:     formula, num = question(mode, operators.choice)
55:     if answer(formula, num)
56:       puts '正解'
57:       right_count += 1
58:     else
59:       puts '不正解'
60:       wrong_count += 1
61:     end
62:     puts "#{n + 1} 問突破" if (n + 1) % 10 == 0
63:   end
64:   end_time = Time.now.to_i
65:   show_result(right_count, wrong_count, start_time, end_time)
66: end
67:
68: if __FILE__ == $0
69:   main
70: end
```

- 問題を生成する部分で後置の while を使用しています。
- 結果を表示する部分でヒアドキュメントを使用しています。
- Ruby1.8.7 以降で追加されたメソッドを使用しています。
- ユーザの入力を求める部分で不正な値を入力された場合は再入力を求めるようにしています。