

Ruby プログラミング 50 題

知識情報システム実習 担当 鈴木，時井，松村

概要

この問題集は，Ruby プログラミングの簡単な復習を意図している．

- グループで一題一題，全員が理解していることを確認しながら進めてほしい．
- プログラムだけが提示されているのは入力して実行せよという意味である．
- 全員が分かりきっていると判断したものはとばして（実行しなくても）構わない．
- 応用問題は，グループでの相談がおおいに必要だろう．
- なお，文字コードは UTF-8 である．プログラムに必須の最初の 1 行は省略してある．

1 print, \n

[Ruby-1]

```
print("Hello, Ruby.\n")
```

[Ruby-2] 前のプログラムを変更して，以下のように表示するプログラムにせよ．

```
Hello,  
Ruby.
```

[Ruby-3]

```
print("Hello, \"Ruby\".\n")
```

2 代入

[Ruby-4]

```
seisuu = 10  
print(seisuu, "\n")  
jissuu = 3.141592  
print(jissuu, "\n")
```

[Ruby-5]

```
mojiretsu = "Ruby"  
print(mojiretsu, "\n")
```

3 算術演算

[Ruby-6]

```
print(1 + 1, "\n")
print(2 - 3, "\n")
print(5 * 10, "\n")
print(100 / 4, "\n")
```

[Ruby-7]

```
print(1.0 + 2.0, "\n")
print(2.0 * 3.0, "\n")
print(5.0 - 8.0, "\n")
print(9.0 / 2.0, "\n")
```

[Ruby-8]

```
print(7 / 2, "\n")
print(7.0 / 2.0, "\n")
```

[Ruby-9]

```
i = 0
print(i, "\n")
i = i + 1
print(i, "\n")
i = i + 1
print(i, "\n")
```

4 文字列

[Ruby-10]

```
myString = "とっても楽しいRuby"
print(myString, "\n")
print(myString, "\n")
print("myString", "\n")
```

[Ruby-11]

```
print(12 + 12)
print("\n")
print("12" + "12")
print("\n")
print("12 + 12")
print("\n")
```

[Ruby-12]

```
when = "今日"
who = "私が"
what = "宙返りを"
where = "実習室で"
why = "寝坊したので"
how = "華麗に"

phrase1 = when + who + what + "した"
print(phrase1, " . \n")

phrase2 = when + who + why + where + how + what + "した"
print(phrase2, " . \n")
```

5 条件分岐

[Ruby-13]

```
a = 10
b = 10

if b > a
  print("b は a より大きい\n")
elsif b == a
  print("b は a と等しい\n")
else
  print("b は a より小さい\n")
end
```

[Ruby-14] 前のプログラムの a, b の値を変えて試してみよ .

[Ruby-15]

```
x = 2
if x >= 1 && x <= 10
  print("x は 1 から 10 の間にあります\n")
end
```

[Ruby-16] 前のプログラムで x の値をいろいろ変えて試してみよ .

6 繰り返し

[Ruby-17]

```
i = 0
print(i, "\n")
i = i + 1
print(i, "\n")
i = i + 1
print(i, "\n")
i = i + 1
print(i, "\n")
```

[Ruby-18]

```
i = 0
while i < 3
  print(i, "\n")
  i = i + 1
end
print(i, "\n")
```

[Ruby-19] 前のプログラムの繰り返しの数 (10) を変えてみよ .

[Ruby-20] 前のプログラムの + 1 の 1 を適当な数値に変えてみよ .

[Ruby-21]

```
i = 0
sum = 0
while i <= 10
  sum = sum + i
  print(sum, "\n")
  i = i + 1
end
```

[Ruby-22]

```
i = 0
j = 0
while i < 9
  j = 0
  while j < 9
    print(i * j, "\n")
    j = j + 1
  end
  i = i + 1
end
```

[Ruby-23]

```
tango = ["knowledge", "information", "system" , "library", "metadata"]

while true
  print("単語を入力してください\n")
  print("終わる時は . (ピリオド)を入力してください\n")
  a = gets.chomp
  if a == "."
    break
  end

  j = 0
  while j < tango.size
    if a == tango[j]
      print(a, "は辞書に登録されています\n")
      break
    end
    j = j + 1
  end
  if j == tango.size
    print(a, "を登録します\n")
    tango.push(a)
  end
end

tango.each{|value|
  print(value, "\n")
}
```

7 gets, chomp

[Ruby-24]

```
str = "Hi\n"
print(str, "!\n")

str1 = str.chomp
str2 = str.chomp.chomp
str3 = str.chop
str4 = str.chop.chop

print(str1, "!\n")
print(str2, "!\n")
print(str3, "!\n")
print(str4, "!\n")
```

[Ruby-25]

```
print("名前を入力してください.\n")
name = gets.chomp
print(name, "さん, こんにちは\n")
```

[Ruby-26] 前のプログラムで, `.chomp` を消して実行してみよ.

8 配列

[Ruby-27]

```
nums = [1, 2, 3]
strs = ["a", "b", "c"]

print(nums[0], "\n")
print(nums[1], "\n")
print(nums[2], "\n")

print(strs[0], "\n")
print(strs[1], "\n")
print(strs[2], "\n")
```

[Ruby-28]

```
name = Array.new(3, "")
print("名前を入力してください\n")
name[0] = gets.chomp

print("名前を入力してください\n")
name[1] = gets.chomp

print("名前を入力してください\n")
name[2] = gets.chomp

print(name[0], "さん、おはようございます!\n")
print(name[1], "さん、こんにちは!\n")
print(name[2], "さん、こんばんは!\n")
```

[Ruby-29]

```
fruit_list = Array.new(5, "")

i = 0
while i < 5
  print("<だもの名を入力してください> ")
  fruit = gets.chomp
  if fruit == "."
    break
  end

  fruit_list[i] = fruit
  i = i + 1
end

print("\n")
print("-----\n")
i = 0
while i < fruit_list.size
  print(fruit_list[i], "\n")
  i = i + 1
end
print("-----\n")
```

9 ハッシュ

[Ruby-30]

```
eng = {"かえる" => "frog", "はち" => "bee", "かも" => "duck",
      "せみ" => "cicada"}

print("選んでください (かえる、はち、かも、せみ) \n")
str = gets.chomp
print(eng[str], "\n")
```

[Ruby-31]

```
eng = {"かえる" => "frog", "はち" => "bee", "かも" => "duck",
      "せみ" => "cicada"}

eng.each{|key, value|
  print(key, ":", value, "\n")
}
```

[Ruby-32]

```
eng = Hash.new
print(eng.size, "\n")
print(eng.empty?, "\n")

eng["かえる"] = "frog"
eng["かも"] = "duck"
eng["せみ"] = "cicada"
print(eng.size, "\n")
print(eng.empty?, "\n")
eng.each{|key, value|
  print(key, ":", value, "\n")
}

eng.delete("かも")
eng.each{|key, value|
  print(key, ":", value, "\n")
}
print(eng.size, "\n")

eng.clear
print(eng.size, "\n")
```

10 メソッド定義

[Ruby-33]

```
def zeikomi(nedan)
  print(nedan * 1.05, "円\n")
end

zeikomi(300)
zeikomi(2600)
```

[Ruby-34]

```
def san_tan_gen(doushi)
  return doushi + "s"
end

print("動詞を入力してください> ")
doushi = gets.chomp
print("3人称単数現在の場合は, ", san_tan_gen(doushi), "としてください\n")
```

[Ruby-35]

```
def max(a, b)
  if a > b
    return a
  else
    return b
  end
end

print(max(2, 3), "\n")
```

[Ruby-36] 前のプログラムえ 2,3 を適当に変えてみよ .

11 正規表現

[Ruby-37]

```
print("文字列を入力してください。 \n")
str = gets.chomp

if /pen/ =~ str
  print("マッチしたで\n")
else
  print("マッチせえへん\n")
end
```

[Ruby-38] 前のプログラムで `pen` の部分を, `^pen`, `pen$`, `^pen$` にそれぞれ変更して実行し, 文字列 `pen`, `penalty`, `happen` を入力してみよ。

[Ruby-39] 前のプログラムで `pen` の部分を, 次のように変えてみよ。どのような入力で「マッチしたで」となるか？

1. `p.n`
2. `p*n`
3. `p+n`
4. `p?n`
5. `p(aei)n`
6. `^...$`
7. `^$`

12 ファイル入出力

[Ruby-40] 以下のプログラムを実行して, `foo.txt` の内容を確認せよ。

```
filename = "foo.txt"
out = open(filename, "w")
out.print("これはテストです。 \n")
out.close
```

[Ruby-41] 前のプログラムで, `out.print` をさらに数行追加してみよ。

[Ruby-42] 以下の出力と foo.txt の内容を比べよ .

```
filename = "foo.txt"
input = open(filename, "r")
line = input.gets
line.chomp!
print("入力内容は「", line, "」です。 \n")
input.close
```

[Ruby-43]

```
io = open("foo.txt", "r")
while true
  line = io.gets
  print(line)
  if line == nil
    break
  end
end
io.close
```

[Ruby-44]

```
io = open("foo.txt", "r")
while line = io.gets
  print(line)
end
io.close
```

[Ruby-45]

```
open("foo.txt", "r") {|io|
  while line = io.gets
    print(line)
  end
}
```

13 応用問題

[Ruby-46] 2 次方程式の解を求めるプログラムを作成せよ .

[Ruby-47] 画面に表示された文字列をキーボードから入力すると , 正解率に応じてメッセージを出力するプログラムを作成せよ . 表示する文字列はファイルに保存されているとする .

[Ruby-48] 子うさぎが 1 羽いる . 子うさぎは 1 年経つと大人のうさぎとなり , また 1 年経つと子うさぎを 1 羽産む . うさぎは死なないものとする . N 年後までの毎年 , 子うさぎと大人のうさぎ

がそれぞれ何羽になっているかを表示するプログラムを作成せよ。

[Ruby-49] 石取りゲームを作成せよ。石取りゲームとは、石の山から順番に石をとっていった最後の石を取った人が負けとなるゲームである。一度に取れる石の最大数は決まっている。ゲームはコンピュータ対人間の 2 人でやることを想定する。また、石の山にある石の数、一度に取れる石の最大数は適宜決定せよ。

[Ruby-50] ふたつのテキストファイルを読み込んで、同じファイルかどうかを判定するプログラムを作成せよ。

(以上)