

# プログラミング演習I

## – 第1回 プログラムに慣れる その1–

### 1 演習の進め方

教科書を参照しながら、演習をすすめよう。(演習\*)の下にプログラムが書かれている場合は、プログラムを実行し、実行結果を確かめよう。(演習\*)の下に文章が書かれている場合は、教科書等を参照し、調べよう。

各演習に対応するように、ファイル名を考えて付けて残しておくようにしよう。(例えば、ex1-1.rb など) 復習するときに便利である。

### 2 プログラムをつかってみよう

CGI プログラムを使ってみよう。

演習 1-1 <http://klis.tsukuba.ac.jp/klib/Subjects/ProgI/example2010/> にアクセスして、プログラムを実行してみよう。

プログラムをダウンロードし、実行してみよう。

演習 1-2 <http://klis.tsukuba.ac.jp/klib/Subjects/ProgI/example2010/> にアクセスして、プログラムをダウンロードして、実行してみよう。

#### 注意事項

プログラミング演習Iのテキスト内の\ (バックスラッシュ) は、キーボードの¥ (エンマーク) を使って入力する。

### 3 Rubyをとにかく使おう その1

ここでは、簡単なプログラムの入力と実行を繰り返してもらい、プログラミングに慣れることを目標とする。プログラムを1行毎に理解する前に、基本的な用語の確認とプログラムでよく使う命令文の確認を行おう。ここでは、全ての事項を完全に覚える必要はない。(もちろん、きちんと理解するに越したことはない。)まずは、全体の雰囲気をつかんでほしい。

どのようなプログラムを記述すると、下記に示したことを行うことができるのだろうか？  
まずは、演習 1-3 から点線内に示されているプログラムを実行し、また教科書を使って調べながらすすめていこう。

プログラミング演習 I 前半で行う内容の一部

文字列を表示するには？

改行するには？

日本語を表示するには？

演算を行うには？

3 足す 2

3 割る 2

変数って何？

キーボードを使って値を入力するには？

入力した値を保存しておきたい

文字列を入力するには？

数値を入力するには？

型変換って何？

文字列 9999 を 数値 9999 に変換するには？

小数を扱うには？

教科書

p.8- プログラムの解説

#### 演習 1-3

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

#### 演習 1-4

文字列オブジェクトとは何か

\n は、何を表すか

print メソッドとは何か

引数とは何か

教科書

p.10- 文字列 (記述方法によって、なぜ結果が異なるのか説明できるようにしよう。)

演習 1-5

```
-----  
print("Hello, \nRuby\n!\n")  
-----
```

演習 1-6

```
-----  
print("Hello, \"Ruby\\\"!\n")  
-----
```

演習 1-7

```
-----  
print("Hello \\ Ruby!\n")  
-----
```

演習 1-8

```
-----  
print('Hello, \nRuby\n!\n')  
-----
```

演習 1-9

演習 1-5 から演習 1-8 の表示結果が異なる理由を説明しよう。

演習 1-10 以下のプログラム中の「文章を入力してみよう」と書かれている部分を書き換えよう。

```
-----  
print("文章を入力してみよう")  
-----
```

教科書

p.12- メソッドの呼び出し

演習 1-11

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

#### 演習 1-12

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

#### 演習 1-13

```
-----  
print("プログラミング", "演習", "が", "はじまってしまった。", "\n")  
-----
```

#### 教科書

p.13- puts メソッド ( print と puts の違いを確かめよう )

#### 演習 1-14

```
-----  
puts("Hello, Ruby.")  
puts("Hello, Ruby.")  
puts("Hello, Ruby,")  
-----
```

#### 演習 1-15

```
-----  
print("Hello, Ruby.")  
print("Hello, Ruby.")  
print("Hello, Ruby.")  
-----
```

#### 教科書

p.15- 日本語の表示

#### 演習 1-16

```
-----  
print("いづれの御時にか女御更衣あまたさぶらひたまいけるなかに\n")  
print("いとやむごとなき際にはあらぬがすぐれて時めきたまふありけり\n")  
-----
```

演習 1-17 ruby -Ks スクリプトファイル名 とするのはどのような場合が調べよう。

教科書

p.175- 算術演算

演習 1-18 算術演算を行うときに使う、+ - / \* の意味を調べよう。

教科書

p.19 - 四則演算 ( 四則演算を行ってみよう。結果が同じになるか確かめよう。 )

演習 1-19

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

演習 1-20

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

演習 1-21

```
-----  
print(1 + 2, "\n")  
print(2 * 3, "\n")  
print(5 - 8, "\n")  
print(9 / 2, "\n")  
-----
```

演習 1-22 演習 1-20 と演習 1-21 の計算結果が異なる理由を調べよう。  
(ヒント：整数と浮動小数点数の違いは?)

教科書

p.20- 数学的な関数

#### 演習 1-23

```
-----  
print("sin(3.1415) = ", Math.sin(3.1415), "\n")  
print("sqrt(10000) = ", Math.sqrt(10000), "\n")  
-----
```

教科書

p.177- Math モジュール

演習 1-24 Math モジュールのメソッドにはどのようなものがあるか調べよう。

演習 1-25 1 行目と 2 行目の表示される値は同じ? 異なる場合はその理由を考えよう。

```
-----  
print(5 * (12 - 8) - 15, "\n")  
print(5 * 12 - 8 - 15, "\n")  
-----
```

演習 1-26 文字列の足し算と掛け算を行うと、どのような結果が得られるか試そう。

```
-----  
print("I like" + " apple pie.", "\n")  
-----
```

#### 演習 1-27

```
-----  
print("blink " * 4, "\n")  
-----
```

#### 演習 1-28

```
-----  
print("Hello," + " Ruby! " * 4, "\n")  
-----
```

### 演習 1-29

```
-----  
print(("Hello," + " Ruby! ") * 4, "\n")  
-----
```

教科書

p.21- 変数

演習 1-30 「変数名 = オブジェクト」は何を意味するのか調べよう。

教科書

p.67- 変数名のつけ方

演習 1-31 変数名のつけ方を調べよう。また、プログラミング演習版コーディング規約（演習ページ内）を参照して、コーディング規約を確かめておこう。

演習 1-32 変数が示すオブジェクトが変わることを確認しよう。

```
-----  
my_String = "楽しい Ruby"  
print(my_String, "\n")
```

```
my_String = "愉快的 Ruby"  
print(my_String, "\n")
```

```
my_String = "面白い Ruby"  
print(my_String, "\n")  
-----
```

演習 1-33 2,3,4 行目の表示結果は同じ？ 異なる場合はその理由を考えよう。

```
-----  
myString = "楽しい Ruby"  
print(myString, "\n")  
print(myString, "\n")  
print("myString", "\n")  
-----
```

#### 演習 1-34

```
-----  
myString = ""  
print(myString, "\n")  
print("myString", "\n")  
-----
```

演習 1-35 変数を使って、文字列を表示するプログラムを作成してみよう。

教科書

p.179 数値型の変換

#### 演習 1-36

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

#### 演習 1-37

```
-----  
print("整数を入力してください\n")  
x = gets.chomp  
print("整数を入力してください\n")  
y = gets.chomp  
print(x + y, "\n")  
-----
```

演習 1-38 gets と chomp について調べよう。



演習 1-39 以下のプログラムはエラーになる。なぜだろうか？

```
-----  
var1 = 2  
var2 = "5"  
print(var1 + var2, "\n")  
-----
```

演習 1-40 演習 1-39 との違いを説明せよ。

```
-----  
var1 = 2  
var2 = "5"  
print(var1.to_s + var2, "\n")  
-----
```

演習 1-41

```
-----  
var1 = 2  
var2 = "5"  
print(var1 + var2.to_i, "\n")  
-----
```

演習 1-42 演習 1-37 を数値の計算にするには、プログラムをどのように修正したらよいか考え、試そう。

演習 1-43

```
-----  
print("15".to_f, "\n")  
print("99.999".to_f, "\n")  
print("99.999".to_i, "\n")  
-----
```

演習 1-44 p.179 「数値型の変換」を参照し、型変換とは何か調べよう。

演習 1-45 読みやすいプログラムを書く方法を調べよう。

p.23 「コメントを書く」を参照し、コメントの書き方を調べよう。

p.114 「読みやすいプログラムを書こう」を参照し、改行や空白の使い方を調べよう。

プログラミング演習版 Ruby コーディング規約（演習ページ内）を参照し、プログラムを記述する際のポイントを確認しよう。