

基礎プログラミングI

第2回 変数・制御構造

メディア情報コース
平居 悠（ひらい ゆたか）

到達目標

計算機上での情報の取り扱い方の基礎の理解

1. コンピュータに指示を送る基本を理解する。
2. 文字列や数値を基本とした値の概念とそれを格納する変数の仕組みを理解する。
3. 変数一つで集合を表す概念を理解する。
4. 2進数を基本としたコンピュータの内部表現について基本を理解する。
5. 現実社会の簡単な問題を扱うプログラムを作成できるようになる。

前回の復習

- プログラミングとは何だろうか？
 - プログラミング言語の文法に従って処理手順を書く工程
- なぜプログラミングを学ぶのか？
 - 世の中を支えるコンピュータが動く仕組みを学べる。論理構成力、状況判断力などが身に付く。
- Rubyで“Hello, world!”と画面に表示させるにはどのように書けば良いか？
 - `print “Hello, world!\n”`とする。

今回

第1回	4月9日	プログラミングの基礎
第2回	4月16日	変数・制御構造
第3回	4月23日	メソッド、値の型変換
第4回	4月30日	確認テスト
第5回	5月14日	集合処理1（配列）
第6回	5月21日	集合処理2（CSVとデータ処理）
第7回	5月28日	集合処理3（ハッシュ）
第8回	6月11日	正規表現
第9回	6月18日	計算機の内部表現
第10回	6月25日	スタイルとデバッグ、実用的なプログラム
第11回	7月2日	作品の公開
第12回	7月9日	チーム課題作成
第13回	7月16日	チーム課題発表
第14回	日程未定	期末試験

今回の目標

値と変数を理解し入出力を利用するプログラムを作成できるようになる。

導入課題

チーム内で2人もしくは3人組を作り、以下から1つ問題を選んで協力して解答せよ。解答はs4基礎プロI (G, H)の「#02 導入課題」に指示通り書き込む。ただし、チーム内のペアごとに異なる問題を解答すること。書き込みは全員行うこと。

- 1. 数値と文字列の違いを説明せよ。**
- 2. 制御構造とは何か。**
- 3. $x > 0$ の時、Hello!と表示するif文を書け。**

タイピング練習スケジュール

- 第1回 ホームポジション
- 第2回 ローマ字
- 第3回 英語初級
- 第4回 日本国憲法 (trr試験)
- 第5回 ホームポジション
- 第6回 ローマ字
- 第7回 英語初級
- 第8回 日本国憲法 (trr試験)
- 第9回 ホームポジション
- 第10回 ローマ字
- 第11回 英語初級
- 第12回 日本国憲法 (trr試験、合格スコア150)

trr起動方法

1. ブラウザを起動し、<https://www.koeki-prj.org/trr/>に繋ぐ。
2. 学籍番号（Cは大文字、省略なし8桁）を入力する。
3. Koeki MAILに届いたパスコードをPasscode: 欄に入力する。

ホームポジション

左手でタイプするキー

右手でタイプするキー



左手の人差指から小指までの
ホームポジション

両手の親指の
ホームポジション

右手の人差指から小指までの
ホームポジション

足し算プログラム

右のプログラム
accum.rbを入力し、
実行せよ。ただし、
全ての行で[Tab]キー
を押して正しいイン
デント位置に揃える
こと。

```
#!/usr/koeki/bin/ruby  
# -*- coding: utf-8 -*-
```

```
sum = 0  
while true  
  STDERR.print "足したい数(終了は q): "  
  line = gets.chomp  
  if line == "q" then  
    break  
  end  
  sum += line.to_i  
  printf("これまでの合計は %d です\n", sum)  
end  
printf("合計は %d です\n", sum)
```

足し算プログラム

実行結果とプログラムを見比べ、以下の点について自分なりの予想を立てよ。予想をs4基礎プロIG, H「#2 授業内課題」に書き込むこと。

1. プログラム中何度も繰り返し処理されているのはどの範囲か？
2. キーボードからデータを読み込んでいるのはどの部分か？
3. breakは何をするものか？
4. breakのところが処理されるのはどういうときか？
5. sumという単語は何を意味しているか？
6. +=という記号は何を意味しているか？
7. printfは何をするものか？
8. printfのところにある%dは実行結果に出てきていないが、どうなってしまったのか？

Rubyで利用できる演算

算術演算：代表的な演算子

演算子	意味
+	加算
*	乗算
-	減算
/	除算
%	剰余
**	べき乗

算術演算：演算子の優先順位

基本的に数学と同じ。

高	↔	低
**	*	+
	/	-
	%	

算術演算：演算子の優先順位

例：

$$1 + 2 * 3 ** 2 = 1 + 2 * 9 = \mathbf{19}$$

括弧 ()があれば先に括弧内が計算される。

$$((1 + 2) * 3) ** 2 = (3 * 3) ** 2 = 9 ** 2 = \mathbf{81}$$

算術演算 : 数式との対応

数学	Rubyの式
$1 + 2 \times 3$	<code>1 + 2 * 3</code>
$(1 + 2) \div 3$	<code>(1 + 2) / 3</code>
$\frac{x + y + z}{3}$	<code>(x + y + z) / 3</code>

算術演算：代入を伴う演算

代入演算子	意味
=	通常代入 $x=5$
+=	加算代入 $x+=5$ で x が 5 増える
=	乗算代入 $x=5$ で x が 5 倍になる
--	減算代入 $x-=5$ で x が 5 減る
/	除算代入 $x/=5$ で x が $1/5$ になる
%	剰余代入 $x%=5$ は $x = x\%5$ と同じ
**	べき乗代入 $x**=5$ は $x = x**5$ と同じ

算術演算：代入を伴う演算の例

x=10のとき、代入演算結果は以下の通り。

代入演算	その後のxの値
$x += 1$	11
$x *= 2$	20
$x -= 1$	9
$x /= 2$	5
$x \% = 3$	1
$x ** = 2$	100

論理演算

制御構造の判定条件式等に利用する演算子のことを**論理演算子**という。

比較演算子	意味
==	左辺と右辺が等しいか
<	左辺が右辺より小さいか
<=	左辺が右辺以下か
>	左辺が右辺より大きいか
>=	左辺が右辺以上か
&&	「かつ」
	「または」
!	否定
not	否定

論理演算の例

answer変数の値が“yes”と等
しかつたら「ほげほげ」

```
if answer == “yes” then
```

```
  ほげほげ
```

```
end
```

論理演算の例

xが12以下かつ1以上なら
「ほげほげ」

```
if x <= 12 && x >= 1 then
```

```
    ほげほげ
```

```
end
```

論理演算の例

yが“はい”または“yes”と等しい場合は「yesだってさ」そうでなければ「yesじゃないってさ」と出力

```
if y == “はい” || y == “yes” then
    printf(“yesだってさ\n”)
else
    printf(“yesじゃないってさ\n”)
end
```

課題

`accum.rb`のように`q`を入力するまで数値を読み込んで、それが偶数なら「偶数です」、そうでなければ「奇数です」と出力するプログラム`evenodd.rb`を作成せよ。

作成できたら挙手をし、TAに作り方と意味を説明する。

レポート提出方法

TAから評価を受けた後、s4基礎プロIG, Hの「#2 提出課題 (課題2)」に以下の通り書き込む。**締め切り：本日中**

レポート本体は

1. 作成したプログラム
2. プログラムの実行結果
3. プログラムの各部の説明（今週覚えたことを重点的に5行程度）
4. 「quiz ruby-2」のスコア
5. 感想
6. `evenodd.rb`プログラムのファイル添付

という構成にする。今回のプログラムは誰が書いても同様のものになるので、「プログラムの各部の説明」を自分が理解した上で**自分の言葉**でしっかりと書く。

今回の目標

値と変数を理解し入出力を利用するプログラムを作成できるようになる。

まとめ

値：

プログラムの中でやり取りされるデータ

変数：

値を入れる箱のようなもの

制御構造：

プログラムの実行順序などを変える仕組み

次回

第1回	4月9日	プログラミングの基礎
第2回	4月16日	変数・制御構造
第3回	4月23日	メソッド、値の型変換
第4回	4月30日	確認テスト
第5回	5月14日	集合処理1（配列）
第6回	5月21日	集合処理2（CSVとデータ処理）
第7回	5月28日	集合処理3（ハッシュ）
第8回	6月11日	正規表現
第9回	6月18日	計算機の内部表現
第10回	6月25日	スタイルとデバッグ、実用的なプログラム
第11回	7月2日	作品の公開
第12回	7月9日	チーム課題作成
第13回	7月16日	チーム課題発表
第14回	日程未定	期末試験

事前課題

1. 第3回解説動画(<https://youtu.be/p2YHv7YtdNQ>)を視聴する。
2. 第3回授業資料値の型変換(<http://roy.e.koeki-u.ac.jp/~yuuji/2026/pf1/04/index.html>)を読む。
3. ターミナルでquiz ruby-3を3回以上行い、最も早くクリアした記録をs4基礎プロIG, Hの「#03 事前課題 (ruby-3)」に貼り付ける。

締切：4月22日 (水)