

CS第1 演習ガイド

本日の予定

1. 準備
2. プログラムを走らせてみる
3. レポート課題2の解答作成

1. 準備

1. ログインする.
2. 講義のウェブページから プログラム をダウンロードする.
 - Downloads(ダウンロード)フォルダに kadai2.zip 置かれる
3. **Terminal** を動かす (TSUBAME と直接対話する窓口).
 - 2.1. **cd** Downloads Downloadsというフォルダに入る.
 - 2.2 **cd** kadai2 kadai2というフォルダに入る. .

2. プログラムを走らせてみる

Ruby で書かれたプログラムを実行する:

現在, `kadai2` の部屋にいるはず. (そうでないときはそこに移動)

3. `ls` その部屋にあるファイルを表示させる.

4. そこにあるプログラムをいくつか実行してみる.

Ruby で書かれたプログラムを実行するための命令 `ruby` を用いる.

4.1 `ruby max.rb` たとえば `max.rb` を実行してみる.

(もし時間があれば)

4.2 `ruby abcPrint.rb` 完成させて実行してみる.

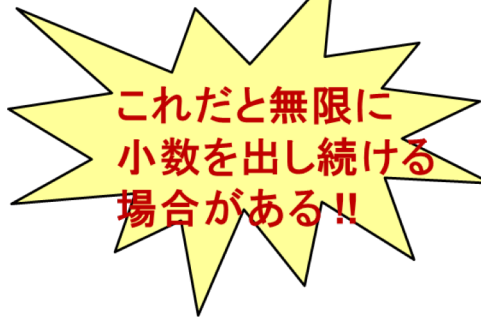
CS第1 レポート課題2

課題 循環小数の循環を止めよ！

配列は同じようなデータを統一的に処理するには便利な道具だが、それ以外にも賢い使い方がいくつかある。その例を考えてみよう。

```
puts("分母 d を下さい")
d = gets().to_i
puts("1 / d を求めます")
stop = 0; leng = 0; x = 1
while stop != 1
  x = x * 10
  q = x / d
  leng = leng + 1
  print(leng, ":", q, "¥n")
  sleep(0.5) # ゆっくり表示するため
  x = x % d
  if x == 0
    stop = 1
  end
end
end
```

junkan.rb



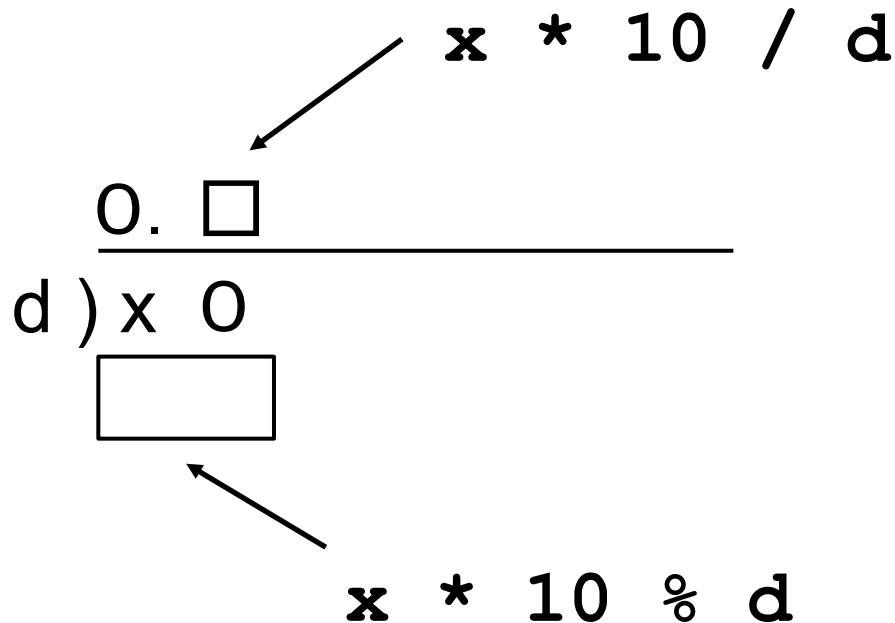
これだと無限に
小数を出し続ける
場合がある!!

プログラムの要求仕様

- ・ 循環小数になっても計算が止まる
- ・ 繰り返しが出る直前、直後くらいまでに止める

プログラムの考え方

- x / d を小数として表示する. ただし, $x < d$



- 変数 stop: プログラムを止めたいときに値を1(0以外)にする

注意

n桁目とm桁目の数が同じでもそこで循環していない場合もある

```
$ ruby junkan.rb  
分母 d を下さい  
17  
1 / 17 を求めます  
1:0  
2:5  
3:8  
4:8  
5:2  
6:3  
7:5  
8:2  
9:9  
10:4
```

```
11:1  
12:1  
13:7  
14:6  
15:4  
16:7  
17:0  
18:5  
19:8
```

Ruby: 出力

【入出力】

`puts(a, b, "hello", c)` ← 変数 `a`, `b` の値, 文字列 `hello`, 変数 `c` の値を改行しながら画面に表示する

`print(a, b, "hello", c, "¥n")` ← 上と同様. ただし改行はしない. 空白も空けない. 従って, 最後には改行記号を画面に出すことで改行させる.

※ プログラムを書くときには, 「¥」の代わりに「バックスラッシュ」をタイプする.

```
1 print( a, b, "hello", c, "\n")
```

※ Mac ではバックスラッシュは Option キーと ¥ キーを同時に押すとタイプできる.

※ Windowsでは, 「¥」と「バックスラッシュ」が同一の文字コードを持つので, 「¥」のままで問題ありません.

CS第1 レポート課題2

提出方法

- OCW-i から提出: 10月24日 レポート課題2
- 提出期限: 10月31日 午前10:40
 - 10月31日 午後9:00までのレポートは, 採点しますが, 大きく減点します.
- 提出物
 - レポート(PDFファイル)
 - ファイル名 report.pdf
 - ファイル形式がPDFでない場合は減点します.
 - プログラム: junkan.rb

CS第1 レポート課題2

提出すべきものと採点基準(満点 15)

1. レポート

- どうやって循環を見つけるのかの説明. A4 で 1 枚程度で十分だが, 読み手は, そのやり方を知らないと思って書くこと(配点 10)
- 工夫した点(加点対象とします, 配点 ~ 5)
 - 循環する部分がわかるようにしよう

2. プログラムのファイル(配点 5; ただし, プログラム自体は読まないで 1できちんと説明すること)

Rubyの配列

【配列】	初期設定	<code>aa = [0, 0, -5, 4]</code> <code>aa = Array.new(4)</code> ← 要素数 4 の配列生成し aa とする <code>aa = Array.new(4, 0)</code> ← 各要素の初期値が 0
	指定方法	<code>aa[i]</code> = aa の <i>i</i> 番目 (添え字 <i>i</i> は 0 から)
	コマンド	<code>aa.length</code> = 配列 aa の長さ (=要素数) ※「添え字」は「インデックス」(index) ともいう.

まとめ: Terminal command

命令	使用例	意味
mkdir	mkdir kadai2	kadai2 というフォルダ(部屋)を作る
cd	cd kadai2	kadai2 というお部屋に入る
	cd ..	上の(大きな)部屋に戻る
	cd ../..	上の上の部屋に戻る
ls	ls	その部屋にあるファイルを表示する
cat	cat foo.rb	foo.rb の中身を表示する
rm	rm foo.rb	foo.rb を消す(戻らないので注意)

まとめ: Terminal command

使用例	意味
<code>mv ~/Downloads/sub.rb .</code>	Downloadsフォルダにあるsub.rb を現在いるフォルダ(.)に移動
<code>mv ~/Downloads/*.rb .</code>	Downloadsフォルダにある～.rb というファイルを全て現在いるフォルダ(.)に移動
<code>open -a coteditor sub.rb</code>	cotEditor で sub.rb を開く