

CS第1 演習ガイド

1. ログインする.
2. Terminal を動かす
3. 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 day4.zip が展開される
4. Terminal で day4 をCS1 に移動
 - `cd Documents/CS1`
 - `mv ~/Downloads/day4 ./`
 - `cd day4`

CotEditor でファイルを開くには

```
$ open -a coteditor ファイル名
```

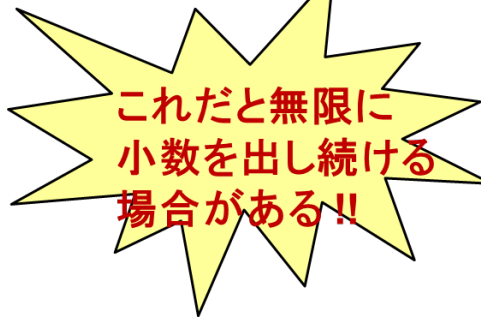
CS第1 レポート課題2

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。

```
print("分母 d を下さい")
d = int(input())
print("1 / ", d, " を求めます")
stop = False
leng = 0
x = 1
while not stop:
    x = x * 10
    q = x // d
    leng = leng + 1
    print(leng, ":", q)
    time.sleep(0.5)
    x = x % d
    if x == 0:
        stop = True
```

junkan.py



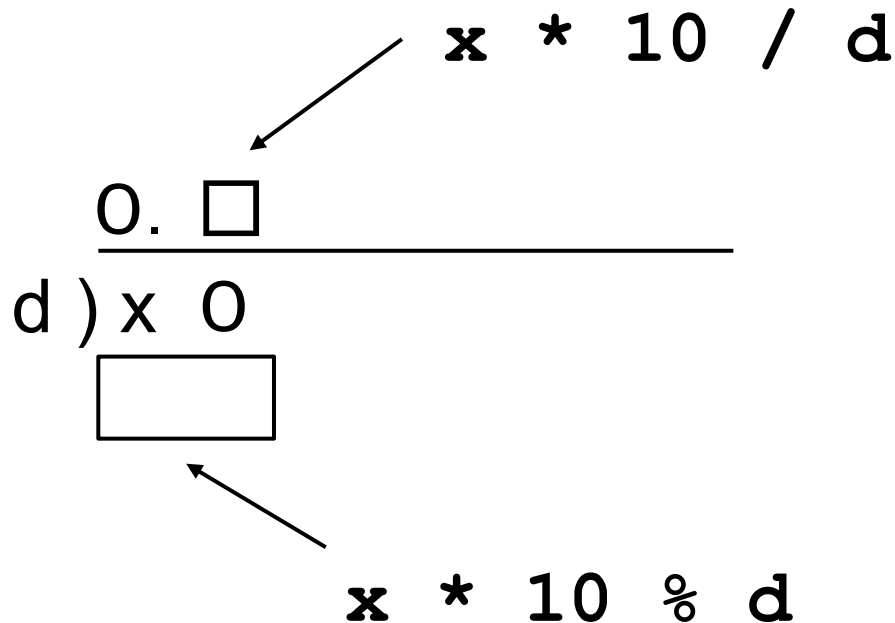
これだと無限に
小数を出し続ける
場合がある!!

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 **True** 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

```
$ python junkan.py
```

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

20 : 8

配列：初期化（大きさが固定でない場合）

```
$ python
...
>>> [3, 1, 4, 1, 5, 9, 2]
[3, 1, 4, 1, 5, 9, 2]
>>> [0] * 10
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
>>> x = 5
>>> [3] * x
[3, 3, 3, 3, 3]
```

各要素が 0 で, 長さ 10 の配列

各要素が 3 で, 長さ x の配列

CS第1 レポート課題2

提出方法

- OCW-i から提出: 10月23日 レポート課題2
- 提出期限: 10月30日 午前10:40
 - 10月30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 **kadai2.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.py**

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)

Pythonの配列

【配列】	配列の作成	[0, 0, -5, 4]	
		[0] * 4	各要素の初期値が 0, 要素数が4の配列
	要素の参照	aa[i]	aa の i 番目 (添え字 i は 0 から)
	大きさ	len(aa)	配列 aa の長さ (= 要素数)
※「添え字」は「インデックス」(index) ともいう.			