

# プログラミング1

## (第5回) ループ処理(for文)、range()関数とリストによるシーケンス 集合表現

1. Chapter 3.2 For Loops
  1. もう一つのループ処理
  2. シーケンス集合とコード例
2. Chapter 3.4 A Few Words About Using Floats
  1. 浮動小数点数の取り扱い
3. 演習
  1. 演習1~4: 初めてのレポート
  2. 演習5: if文, 関数の利用
  3. 演習6: while文
4. 宿題

講義ページ: <http://ie.u-ryukyu.ac.jp/~tnal/2016/prog1/>

# Chapter 3.2, 3.4の補足

3.2 For Loops

3.4 A Few Words About Using Floats

## 3.2 For Loops (反復) \* 2つ目の反復制御

### range ()関数

range(stop): 0～stop-1までのint型オブジェクトを生成  
range(start, stop): start～stop-1

```
x = 4  
for i in range(0, x):  
    print(i)
```

### for文

「in ～」として指定されたシーケンス集合に対して、  
(1)1つずつ要素を取り出し、  
(2)その要素を対象としてブロックを実行(反復処理)。  
(3)全要素に対して(2)を実行し終わったらfor文を終了。

シーケンス集合の例: str, range, list

# シーケンス集合の例1 (str型オブジェクト)

```
# コード1  
for c in 'abc':  
    print(c)
```

```
# 結果1
```

```
a  
b  
c
```

```
# str型オブジェクト(復習)
```

```
>>> len('abc')  
3  
>>> 'abc'[0]  
'a'  
>>> 'abc'[1]  
'b'  
>>> 'abc'[2]  
'c'  
>>> enemy = 'naltoma'  
>>> enemy[0]  
'n'
```

# シーケンス集合の例2 (range型オブジェクト)

## # コード2

```
index = 0
```

```
word = 'abc'
```

```
for index in range(len(word)):
```

```
    print('word[{0}] = {1}'.format(index, word[index]))
```

## # 実行結果2

```
word[0] = a
```

```
word[1] = b
```

```
word[2] = c
```

### str.format()メソッド

- 対象文字列は{}で区切られた置換フィールドを含む。
- 置換フィールドを何で置き換えるかを引数で指定。
- e.g.,  
'{0}' and '{1}'.format('spam', 'eggs')  
'pi = {0:.3f}'.format(3.14159)

{ }: brace

参考: str.format()メソッド

チュートリアル: <http://docs.python.jp/3/tutorial/inputoutput.html>

# シーケンス集合の例3 (list型オブジェクト)

## # コード3

```
items = ['I did it all', 4, 'love']  
index = 0  
for i in items:  
    print('items[{0}] = {1}'.format(index, i))  
    index += 1
```

## # 結果3

```
items[0] = I did it all  
items[1] = 4  
items[2] = love
```

### list (リスト) 型オブジェクト

- 順序付けられたオブジェクト集合。
- 「リスト名」= 集合名
- 「リスト名[インデックス]」= 指定した順番のオブジェクト。
- e.g., items[0] = 'I did it all'







# Reserved words, 予約語

<https://goo.gl/rEzdAN>

- 一覧 (赤丸は今回出てきた予約語)

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

# 演習

演習1～4: 初めてのレポート

演習5: if文, 関数の利用

演習6: while文の利用

# 補足1

- ペアプログラミングを始める前に
  - 記入漏れ
    - 「実施日」と「報告者」
  - 前回の復習確認
    - 「何をやったっけ？」
    - 「これはこうやれば良いんだっけ？」

# 補足2

- ペアプログラミングのやり方

- 7ステップ

- 作業を決める
- 最初の目標を決める
- パートナーを頼りにし、支えてやる
  - driver: 仕事を終わらせることに専念
  - observer: 横から観察し、疑問・改善・簡潔化など大局的な問題について考える
- 喋る
  - 「一人で悩む」のは十秒程度に留める
  - 一緒に相談しながら考える練習
- お互い何をやっているか把握する
  - 頻繁に同期をとる
- 喜ぶ
- 交代する

分業ではない(observer=観察しながら気づいたことをコメント)

二人で2,3分考えても分からない場合には、手を上げて質問しよう

# 演習

- 演習1～演習4: 初めてのレポート
  - <https://ie.u-ryukyu.ac.jp/~tnal/2016/prog1/week2-ex.html>
  - レポートの作成手順は授業ページを参照
- 演習5: if文と関数
  - <https://ie.u-ryukyu.ac.jp/~tnal/2016/prog1/week3-ex.html>
- 演習6: while文
  - <https://ie.u-ryukyu.ac.jp/~tnal/2016/prog1/week4-ex.html>
- ペア・プログラミング
  - <https://ie.u-ryukyu.ac.jp/~tnal/2016/prog1/week2-pair-programming.html>
  - driver, observer (navigator)

# 宿題

- 復習: **適宜**(これまでの内容)
- 予習: 教科書読み
  - 4章
    - 4 Functions, scoping, and abstraction (冒頭1ページ)
    - 4.1 Functions and Scoping
      - (4.1.1 Function Definitions) #終了済み
      - 4.1.2 Keyword Arguments and Default Values
      - 4.1.3 Scoping
    - 4.2 Specifications
    - (4.3 Recursion) スキップ
    - (4.4 Global Variables) スキップ
    - 4.5 Modules
- 復習・予習(オススメ): paiza
  - Python入門編1:プログラミングを学ぶ(全9回)
    - <https://paiza.jp/works/python3/primer>
  - プログラミングスキルチェック \*レベル設定のある課題集
    - <https://paiza.jp/challenges/info>

# 参考文献

- 教科書: Introduction to Computation and Programming Using Python, Revised And Expanded Edition
- Python 3.5.1 documentation, <https://docs.python.org/3.5/index.html>
- str.formatメソッド, <http://docs.python.jp/3/tutorial/inputoutput.html>
- 【5分で覚えるIT基礎の基礎】ゼロから学ぶ2進数 第4回  
2進数で小数を表す方法, <http://itpro.nikkeibp.co.jp/members/ITPro/ITBASIC/20020624/1/?rt=nocnt>
- 倍精度浮動小数点数, <https://ja.wikipedia.org/wiki/倍精度浮動小数点数>