

# プログラミング1

## (第4回) 関数の利用2、ループ処理 (while文)

### 1. Chapter 4.1.1の補足2

#### 1. 関数とローカル変数

関数内の変数と、関数外の変数は**スコープ**が異なることに注意。

### 2. Chapter 3.1の補足

#### 1. Iteration, looping (反復処理)

#### 2. ループ処理の例、実行例

繰り返させたい処理を**ブロック**で指定し、**繰り返し条件**を設定。

#### 3. 3種類の処理流れ制御

### 3. 演習

処理の流れは**逐次・条件分岐・反復処理**の3タイプのみ。

### 4. 宿題

#### 基本道具

- ・型 (int, float, str, bool)
- ・演算 (数値・文字列・比較・論理)
- ・フロー制御 (if, while, for)
- ・関数定義 (def)

講義ページ: <http://ie>

# 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. 演習・課題への取り組み方、レポートの書き方のデモ
2. 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
3. Chapter 3.4 A Few Words About Using Floats
  1. 浮動小数点数の取り扱い
4. 演習
  1. 演習1~4: 初めてのペア・プログラミング
  2. 演習5: 数当てゲーム1 (大小ヒント付き) を実装してみよう
5. 宿題

講義ページ: <http://ie.u-ryukyu.ac.jp/~tnal/2017/prog1/>

# 演習・課題への取り組み方、 レポートの書き方のデモ

# 取り組み方の例

- 問題を分割する。
  - 分割して分かるところから手を付ける(土台を作る)
  - 分からないところは、更に分割できないか考える
    - それでも分からないなら、分割の仕方や、分割した問題の解き方を訪ねてみよう
    - 「分割の仕方」=問題解決手段の一つ
- 個々のサブ問題を個別に解く。
  - これ以上分割できない⇨最小の部品なら教科書・授業で習ってるはず
    - ->該当部分を復習
      - 該当部分が分からないなら、該当部分の探し方を尋ねてみよう。
      - 教科書・授業の復習が足りてないかも。
- それらの組み合わせ方を考える。

# 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. 演習・課題への取り組み方、レポートの書き方のデモ
2. 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
3. Chapter 3.4 A Few Words About Using Floats
  1. 浮動小数点数の取り扱い
4. 演習
  1. 演習1~4: 初めてのペア・プログラミング
  2. 演習5: 数当てゲーム1 (大小ヒント付き) を実装してみよう
5. 宿題

KISS原則を踏まえて  
小さく作ろう。分割の  
仕方を考えよう。

# 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までの全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'

# 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. 演習・課題への取り組み方、レポートの書き方のデモ
2. 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
3. Chapter 3.4 A Few Words About Using Floats
  1. 浮動小数点数の取り扱い
4. 演習
  1. 演習1~4: 初めてのペア・プログラミング
  2. 演習5: 数当てゲーム1 (大小ヒント付き) を実装してみよう
5. 宿題

シーケンス集合に対する反復処理を書けるようになる。





# 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. 演習・課題への取り組み方、レポートの書き方のデモ
2. 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
3. Chapter 3.4 A Few Words About Using Floats
  1. 浮動小数点数の取り扱い
4. 演習
  1. 演習1~4: 初めてのペア・プログラミング
  2. 演習5: 数当てゲーム1 (大小ヒント付き) を実装してみよう
5. 宿題

小数を使う際には丸め  
誤差と桁あふれに注意。

講義ページ: <http://ie.u-ryukyu.ac.jp/~tnal/2017/prog1/>

# 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	

# 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. 演習・課題への取り組み方、レポートの書き方のデモ

KISS原則を踏まえて小さく作ろう。分割の仕方を考えよう。

2. 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

シーケンス集合に対する反復処理を書けるようになる。

3. Chapter 3.4 A Few Words About Using Floats

1. 浮動小数点数の取り扱い

小数を使う際には丸め誤差と桁あふれに注意。

4. 演習

1. 演習1~4: 初めてのペア・プログラミング
2. 演習5: 数当てゲーム1 (大小ヒント付き) を実装してみよう

5. 宿題

講義ページ: <http://ie.u-ryukyu.ac.jp/~tnal/2017/prog1/>

# 演習

演習1～4: 初めてのペア・プログラミング

演習5: 数当てゲーム1 (大小ヒント付き)  
を実装してみよう

# 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, progate

# 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>