

数値計算基礎 / 貯留層工学入門 講座

関数プログラミング

CONTENTS

1	Python 関数
2	外部自作モジュール
3	演習
4	Tips

CONTENTS

1	Python 関数
2	外部自作モジュール
3	演習
4	Tips

1. Python関数

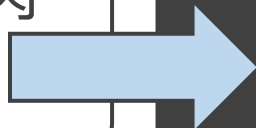
関数とは

◆ 同じタイプの入力に対して同じ処理を実行

入力（引数）

○：鰯，鯛

×：大根，麦茶，豚肉

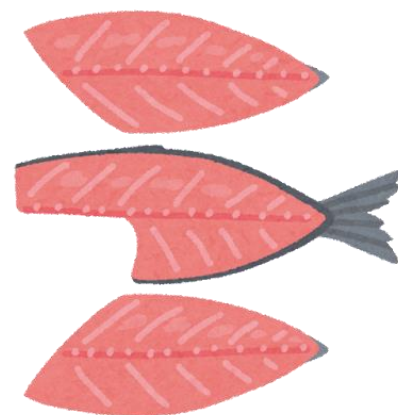


関数

SanmaiOroshi (Sakana)



出力（戻り値）



1. Python関数

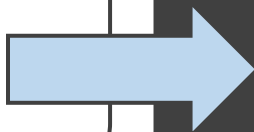
関数とは

◆ 同じタイプの入力に対して同じ処理を実行

入力（引数）

○：鰻，大根，豚肉

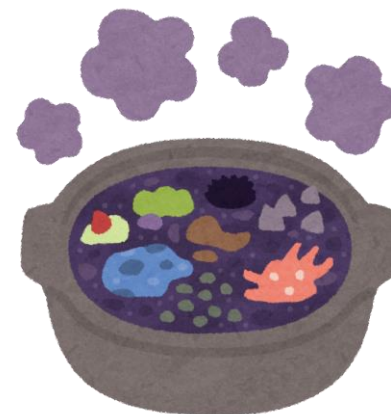
×：鉄



関数
Nabe (Food)



出力（戻り値）



1. Python関数

Pythonにおける関数 (Quick Review)

◆ 定義は次の通り

```
def 関数名(引数1, 引数2, ...):  
    処理  
    return 戻り値1, 戻り値2
```

◆ ただし, 戻り値はオプション (なくてもよい)

```
def Keisan(a, b):  
    kaerichi = a + b  
    return kaerichi
```

```
Num1 = 10;
```

```
Num2 = 5;
```

```
Goukei = Keisan(Num1, Num2);
```

1. Python関数

Pythonにおける関数 (Quick Review)

◆ 定義は次の通り

```
def 関数名(引数1, 引数2, ...):  
    処理  
    return 戻り値1, 戻り値2
```

◆ ただし, 戻り値はオプション (なくてもよい)

```
def Keisan(a, b):  
    kaerichi = a + b  
    print(kaerichi)  
    return
```

```
Num1 = 10;
```

```
Num2 = 5;
```

```
Keisan(Num1, Num2);
```

CONTENTS

1

Python 関数

2

外部自作モジュール

3

演習

4

Tips

2. 外部自作モジュール

使い方

- ◆ 関数は必ずしも呼び出し元のコードと同一のファイルに記述する必要はない。

sample.py

```
def Keisan(a, b):  
    kaerichi = a + b  
    print(kaerichi)  
    return  
Num1 = 10;  
Num2 = 5;  
Keisan(Num1, Num2);
```

フォルダ : example

main.py

```
from sansuu import Keisan  
Num1 = 10;  
Num2 = 5;  
Keisan(Num1, Num2);
```

sansuu.py

```
def Keisan(a, b):  
    kaerichi = a + b  
    print(kaerichi)  
    return
```

CONTENTS

1	Python 関数
2	外部自作モジュール
3	演習
4	Tips

3. 演習

ファイルのダウンロード

数値計算基礎（中級）

- 拡散方程式（陽解法）：[\[pdf\]](#) [\[pptx\]](#) [\[.py演習\]](#) [\[.ipynb演習\]](#)
[\[.py解答\]](#) [\[ipynb解答\]](#)
- 拡散方程式（陰解法）：[\[pdf\]](#) [\[pptx\]](#) [\[.py演習\]](#) [\[.ipynb演習\]](#)
[\[.py解答\]](#) [\[ipynb解答\]](#)
- 疎行列：[\[html\]](#) [\[.ipynb\]](#)
- 関数プログラミング：[\[pdf\]](#) [\[pptx\]](#) [\[zip演習用\]](#)
- 線形と非線形：[\[pdf\]](#)

CONTENTS

1	Python 関数
2	外部自作モジュール
3	演習
4	Tips

4. Tips

関数の使いどころ

- ◆ 事前に処理の流れをよく検討する。
- ◆ main関数（すべての関数を呼び出す関数）を作るとなおよい
- Jupyter Notebook (.ipynb) はデータ処理には向いていても、シミュレータ開発には不適切のように感じる（私見）。