

# Python講座

第三回 ラムダ式と高階関数

# 高階関数とは

- 高階関数とは、他の関数を引数として受け取る関数のことである
- pythonには予め用意された高階関数として、map, filter, reduceなどがある
- これらの関数はループ構文の省略として使用することができる
- 今回は、for文で記述した場合と比較しながら高階関数を確認していく

# map(function,iterable)

- リストの全ての要素に同じ関数を適用させることができる
- 引数functionで関数, 引数iterableはリストを受け取ることができる
- 返り値は関数を適用させた結果のリスト

# 3page\_lec3-1.py

```
# coding: utf-8

def plusOne(x):
    return x+1

input_list = [1,2,3]

output_list = []

for i in input_list:
    output_list.append(plusOne(i))

for i in output_list:
    print(i)
```

# 4page\_lec3-2.py

```
# coding: utf-8

def plusOne(x):
    return x+1

input_list = [1,2,3]

output_list = map(plusOne,input_list)

for i in output_list:
    print(i)
```

# mapまとめ

- このようにfor文を書かずにリストに対しての処理を行うことができる
- 変数の宣言も減らすことができ、バグを少なくすることができる
- for文のみでプログラムを記述するとインデントが増えるので高階関数で読みやすいソースを記述することをおすすめする

# ラムダ式

- 名前の無い関数(匿名関数)を定義することができる.
- ラムダ式による関数の定義
  - lambda 引数:処理
- 関数定義との比較
  - 関数定義  
def plusOne(x):  
    return x+1
  - ラムダ式  
plusOne = lambda x:x+1

# ラムダ式的具体例(page5\_lec3-3.py)

```
# coding: utf-8
```

```
plusOne = lambda x:x+1
```

```
result = plusOne(2)
```

```
print(result)
```

- Output
  - 3



# mapとラムダ式(page6\_lec3-5.py)

```
# coding: utf-8
```

```
input_list = [1,2,3]
```

```
output_list = map(lambda x:x+1,input_list)
```

```
for i in output_list:
```

```
    print(i)
```

# filter(function,iterable)

- リストの全ての要素に同じ関数を適用させることができる
- 引数functionで関数, 引数iterableはリストを受け取ることができる
  - 但し, 関数はTrue or Falseを返す
- 返り値は関数を適用させTrueを返した要素のリスト

# for文でfilter (page7\_lec3-6.py)

```
# coding: utf-8

input_list = [1,2,3,4,5,6]

output_list = []

for var in input_list:
    if var % 2 == 0:
        output_list.append(var)
    else: continue

for i in output_list:
    print(i)
```

# filter(page7\_lec3-7.py)

```
# coding: utf-8
```

```
input_list = [1,2,3,4,5,6]
```

```
output_list = filter(lambda x:x % 2 == 0,input_list)
```

```
for i in output_list:
```

```
    print(i)
```

# filterまとめ

- filterはmap以上に記述を省略することができる
- 条件式をラムダ式で表すのが困難な場合もあり、その時は関数で定義した方がいい

# reduce

- 要素の畳み込み演算を行える

# for文でreduce(page8\_lec3-8.py)

```
# coding: utf-8
```

```
input_list = [1,2,3,4,5,6]
```

```
result = 1
```

```
for var in input_list:
```

```
    result = result * var
```

```
print(result)
```

# reduce(page9\_lec3-9.py)

```
# coding: utf-8
```

```
from functools import reduce
```

```
input_list = [1,2,3,4,5,6]
```

```
result = reduce(lambda x,y:x*y,input_list)
```

```
print(result)
```



# 高階関数でStringをintに変換したい

Page9\_lec3-10.py

```
# coding: utf-8

input_data = "1,3,4,5,6,7,8"

#空白で区切り, Stringのリストを作成

tmp_list = input_data.split(",")

#int型に変換する関数intをtmp_listに適用する

result = map(int,tmp_list)

for var in result:

    print("value:"+str(var))

    print(type(var))
```

# 高階関数でintをStringに変換したい

Page10\_lec3-11.py

```
# coding: utf-8

from functools import reduce

input_data = [1,3,4,5,6,7,8]

tmp_list = map(str,input_data)

result = reduce(lambda x,y:x+","+y,tmp_list)

print(result)

print(type(result))
```