

# Python 入門

繰り返し処理

条件分岐と並んでプログラムの根幹をなすのが繰り返し処理です。  
ここでは、この繰り返し処理について扱います。

# while 文

ある条件が満たされている間処理を繰り返したいときに使います。

`while` 条件式:

処理 # 条件式が満たされている間はずっと繰り返される

このように書くと、`条件式` が満たされている限り、`処理` が繰り返されます。

# 例

「0 以上 5 未満の整数を順に出力してください。」

```
i = 0

while i < 5:
    print(i)

    i += 1
```

```
0
1
2
3
4
```

## for 文

ある整数からある整数まで 1 ずつ増やしながら処理を繰り返したいときに便利

```
for 変数 in range(スタートの数, ゴールの数):  
    処理
```

- 変数が スタートの数 から ゴールの数 未満まで 1 ずつ増えながら 処理 が繰り返される。

「0 以上 5 未満の整数を順に出力してください。」

```
for i in range(0, 5):  
    print(i)
```

```
0  
1  
2  
3  
4
```

# break

ループを途中で抜きたいときは `break` を使います。

```
for i in range(0, 5):  
    if i == 3:  
        print("ぬける")  
        break  
  
    print(i)  
  
print("終了")
```

```
0  
1  
2  
ぬける  
終了
```

## continue

繰り返しの途中で処理をスキップしたいときは `continue` を使います。

```
for i in range(0, 5):  
    if i == 3:  
        print("スキップ")  
        continue  
  
    print(i)  
  
print("終了")
```

```
0  
1  
2  
スキップ  
4  
終了
```



## 補足 1

### 省略した書き方

```
for i in range(0, N):
```

のように for 文が 0 から始まるときは

```
for i in range(N):
```

と、0 を省略して書くことができます。

## 補足 2

### 逆順にループを回したいとき

```
for i in range(A, B):
```

を  $B - 1, B - 2, \dots, A$  のように逆順に回したいときは

```
for i in reversed(range(A, B)):
```

と書くとよいです。

# 演習

これまでの内容で APG4bPython の演習問題

- EX 8
- EX 9

を解くことができます。実際に手を動かしてやってみましょう！