

# 4. 制御文

---

## □ 概要

- Pythonの制御文について学びます。

## □ 学習内容

- 4.1 比較演算子、論理演算子
- 4.2 if文
- 4.3 pass文
- 4.4 while文
- 4.5 for文
- 4.6 breakとcontinue
- 4.7 演習問題(別紙)



## 4.1 比較演算子、論理演算子

### □ 比較演算子

- 条件式などで、値を比較するために使用する記号のこと

演算子	記述例	意味
==	a==b	aとbは等しい
!=	a!=b	aとbは等しくない
>	a>b	aはbより大きい
<	a<b	aはbより小さい
>=	a>=b	aはb以上
<=	a<=b	aはb以下

## 4.1 比較演算子、論理演算子

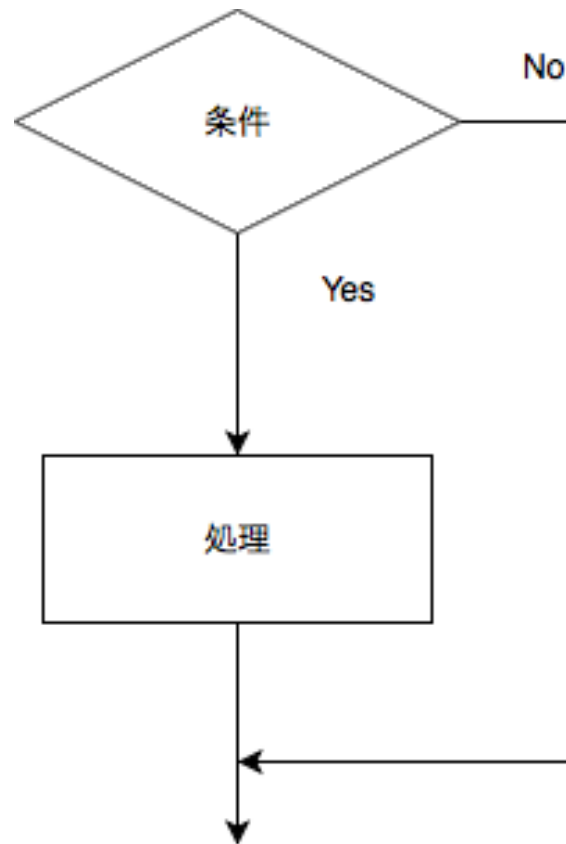
### □ 論理演算子

- 複数の条件に当てはまるような条件式を記述したい場合に使用する。

演算子	記述例	意味
and	条件1 and 条件2	条件1と条件2の両方に当てはまる
or	条件1 or 条件2	条件1か条件2のどちらかに当てはまる
not	not 条件1	条件1が当てはまらない時に処理を実行

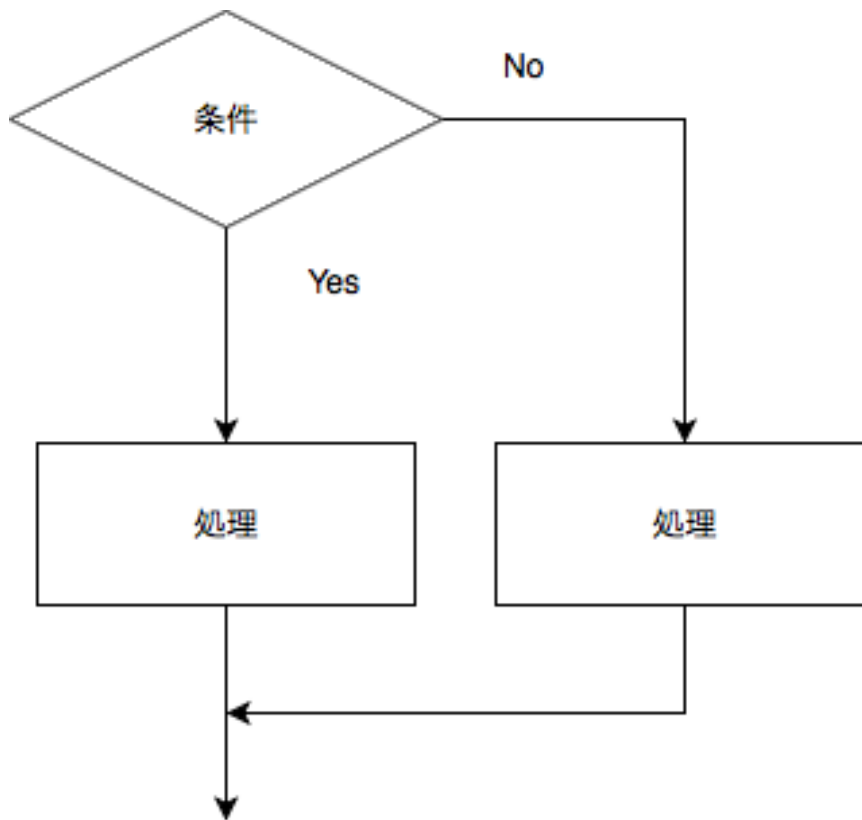
## 4.2 if文

### □ 分岐処理のフロー



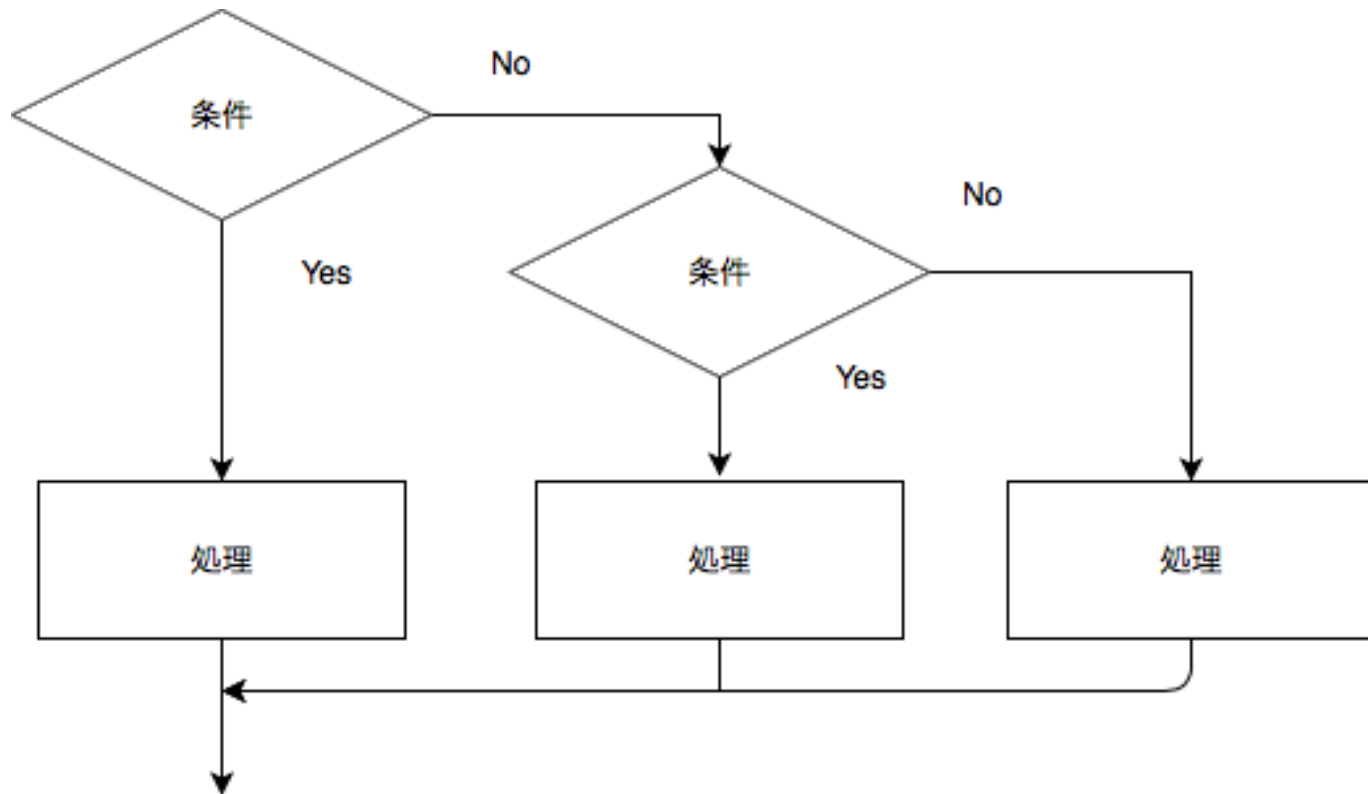
## 4.2 if文

### □ 分岐処理のフロー



## 4.2 if文

### □ 分岐処理のフロー



## 4.2 if文

---

- ある条件で処理を分岐させたい場合には、if文を使用します。

**if** [条件式]:

[実行する処理]

## 4.2 if文

---

- 条件で分岐させ異なる処理を実行する場合
  - if else構文を使用します。

```
if [条件式]:
```

```
    [実行する処理1]
```

```
else:
```

```
    [条件に当てはまらない処理]
```



## 4.2 if文

---

### □ 複数の条件で分岐させたい場合

- 複数の条件で分岐をさせたい場合は、elifやelseを使用します。

```
if [条件式]:
```

```
    [実行する処理1]
```

```
elif [条件式]:
```

```
    [実行する処理2]
```

```
else:
```

```
    [全ての条件に当てはまらない処理]
```



## 4.3 pass文

---

- Pythonには、「何もしない」ということを表すpass文というものがあります。

条件分岐の際のelse節などで使用します。

- なぜpass文が必要？

Pythonは、インデントのブロック構造のため、文法上、インデントの後に何も処理が記述されていないとエラーになってしまいます。そのため、何も処理をしない場合は、「何もしない」という処理を示すpass文を記述する必要があります。

## 4.4 while文

---

- ある条件を満たしている間、繰り返し処理を実行したい場合には、while文を使用します。式の書き方はif文と変わりません。

**while** [条件式]:

[条件式を満たす場合に実行する処理]

## 4.5 for文

---

- 繰り返し処理を実行したい場合には、for文を使用します。

```
for [変数] in [繰り返すデータ]:  
    [実行する処理]
```

## 4.6 breakとcontinue

---

### □ break

- breakを使用すると、以降の処理を行わず、for文やwhile文の処理を終了させます。

### □ continue

- continueを使用すると、以降の処理をスキップし、for文やwhile文の先頭に処理を戻します。



## 4.7 演習問題(別紙)

---

□ 演習問題を解いてみましょう。

