

第1回 Python講座

2017/06/22

インストーラ(Windows)

1. 公式サイトからPython3.x.xをダウンロードする
2. インストーラを実行する
3. テキストを参考にPATHを設定する
4. コンソールでpythonコマンドを実行してみる

インストール(Mac)

1. brewコマンドを使ってpython3をインストール
2. PATHを設定する
3. 端末でpython3を実行する

インストール(Linux)

- 既にインストールされているはずです
- 端末でpython3を実行してみる

インデント

Pythonでインデント（字下げ）は

重要！！

```
// C · Java
```

```
int a = 5;
```

```
    int b = 1;
```

```
for (i = 0; i < 5; i++){
```

```
    if ((a + b) % 2 == 0){
```

```
        printf("a + b = %d\n", a+b);
```

```
            a++;
```

```
    }
```

```
}
```

```
// C・Java
```

```
int a = 5;
```

```
    int b = 1;
```

```
for (i = 0; i < 5; i++){
```

```
if ((a + b) % 2 == 0){
```

```
printf("a + b = %d\n", a+b);
```

```
    a++;
```

```
}
```

```
}
```

インデントがバラバラ

```
# Python
```

```
for i in range(5):
```

```
    if (a + b) % 2 == 0:
```

```
        print(f"a+b={a+b}")
```

```
        a += 1
```

C・Java

ブロックは { … } で記述される

Python

ブロックはインデントで決まる

コメント

- コメントの記述には「#」, 「''' '''」, 「""" """」を使う
- 「#」… 1行のコメント
- 「''' '''」, 「""" """」… 複数行のコメント

コメント

''' 複数行にわたって
コメントを書く'''

出力

コンソール上に値を出力するためにはprint関数を使用する

```
# coding: utf-8  
  
print("Hello Python!!")
```

変数

- 「動的型付け」… コンパイル時に型チェックを行う
 - シンプルで自由度の高いコードを書くことができる
 - コンパイル時にエラーが発生しない
- 変数宣言時に型を明示する必要がない

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

```
a = 5  
print(type(a))
```

```
s = "Python"  
print(type(s))
```

入力

- コンソールからの入力を受け取るにはinput関数を使う

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

```
N = input()  
print(N)  
print(type(N))
```

```
M = int(input())  
print(M)  
print(type(M))
```

条件分岐

if 条件1:

 处理1

elif 处理2:

 处理 2

.....

elif 条件n:

 处理n

else:

 处理

条件分岐

```
if 条件1:  
    処理1  
elif 処理2:  
    処理2  
.....  
elif 条件n:  
    処理n  
else:  
    処理
```

else if ~
ではない

