引き継ぎ資料 Vol.3

CにないPythonの世界

2016/07/??

コンセプト

Cには無いPythonの世界を堪能しよう!

目次

目次

Cの言語仕様にはないデータ構造

- タプル
- リスト
- ディクショナリ

タプルとは

数が並んだもの

Cで近い機能は配列

タプル

簡単な例

```
>>> t = (1, 2, 3)
>>> print(t)
(1, 2, 3)
>>> print(t[0])
1
```

ここまではCの配列と同じ

タプル

あえてCで書くなら...

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char** argv) {
    int t[] = { 1, 2, 3 };
    printf("%d\n", t[0]);
    return 0;
}
```

簡単!

難しい例

```
>>> t = 1, 2, 3
>>> x, y, z = t
>>> def hoge():
...         return 4, 5, 6
...
>>> a1, a2, a3 = hoge()
>>> a = hoge()
>>> u, v, w = z, y, x
```

それぞれの変数の中身は?

答

```
>>> t = 1, 2, 3
>>> x, y, z = t
>>> def hoge():
...     return 4, 5, 6
...
>>> a1, a2, a3 = hoge()
>>> a = hoge()
>>> u, v, w = z, y, x
```

```
>>> print(t)
(1, 2, 3)
>>> print(x, y, z)
1 2 3
>>> print(a1, a2, a3)
4 5 6
>>> print(a)
(4, 5, 6)
>>> print(u, v, w)
3 2 1
```

タプルの要点

- タプルに必要なのは“,”(カンマ)
- 複数の値を返す関数はタプルを一つ返す関数
- タプルは自動的に展開され複数の変数に代入
- タプルでは要素の変更は不可

タプルでは要素の変更は認められない

```
>>> t = (1, 2, 3)
>>> t[1] = 4
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'tuple' object does not support item
assignment
```

リストとは

オブジェクト(整数，小数，その他)が並んだもの

後から変更が可能

例

```
>>> t = []
>>> t.append(1)
>>> t.append(2)
>>> print(t)
[1, 2]
>>> t[1] = 3
>>> print(t)
[1, 3]
>>> lst = [1, 3, 5, 7, 9]
>>> print(lst)
[1, 3, 5, 7, 9]
>>> [a, b] = [1, 3]
```

便利な例

```
>>> t = [1, 3, 5, 7, 9]
>>> print(t[1:4])
[3, 5, 7]
>>> print(len(t))
5
>>> s = [2, 4, 6, 8]
>>> print(t + s)
[1, 3, 5, 7, 9, 2, 4, 6, 8]
>>> print(t * 2)
[1, 3, 5, 7, 9, 1, 3, 5, 7, 9]
>>> print(2 in t, 3 in t)
False True
```

ディクショナリとは

添字に数字以外が使えるリスト

言語によっては連想配列・Map・HashMapなど

簡単な例

```
>>> d = {"a": 1, "b": 2, "c": 3}
>>> print(d["a"])
1
>>> d["b"] = 5
>>> print(d)
{'b': 5, 'c': 3, 'a': 1}
```

注意点

- 存在しない要素を取得しようとするとエラー
- 存在しない要素に代入すると要素を追加

まとめ

タプル・リスト・ディクショナリはCには無い

Pythonにはデフォルトで存在

お互いにネスト(入れ子)が可能

Cよりも柔軟に複雑なデータ構造を表現可能

目次

Cの関数どこまでマスターしてますか?

Cの関数

```
int add(int a, int b) {
    return a + b;
}
void swap(int* a, int* b) {
    int tmp = *a;
    *a = *b;
    *b = tmp;
}
void map(int* array, int N, int (*func)(int)) {
    for (int i = 0; i < N; ++i)
        array[i] = func(array[i]);
}
```

Cで出てきた概念

- 引数
- 戻り値
- 値渡し・参照渡し
- 関数ポインタ

Cで出てきた概念

- 引数
- 戻り値
- 値渡し・参照渡し
- 関数ポインタ

Pythonはもう少し難しい概念を持つ

目次

目次

目次