

みんなの Python

-4. 関数 -

山木翔馬

関数を呼び出す

```
>>> len("hello, world")
```

```
12
```

関数によっては引数の型や数が決まっている。

```
>>> len(1)      # エラー
```

```
# len関数の引数はシーケンス
```

関数の定義

関数名

引数

`def func_name(arguments):`
 `print(arguments)`

インデント

引数の型

- Python では引数の型指定がない
→どのようなオブジェクトでも引数に渡せる

- 必要であれば自分で型チェック

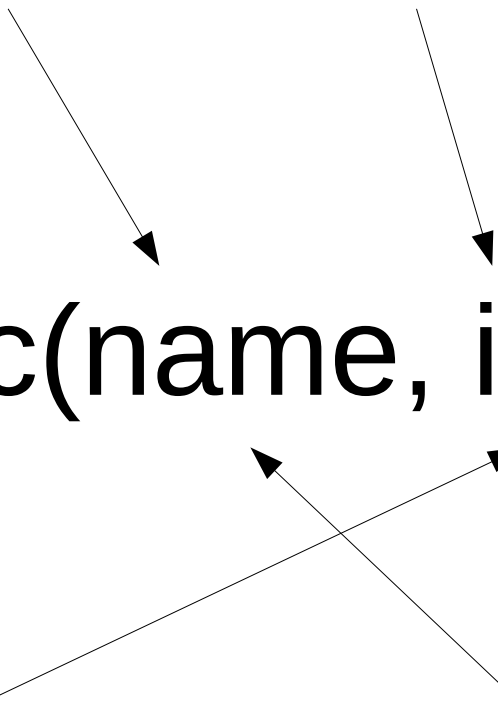
```
def division(x, y):  
    if (type(x) == int) and (type(y) == int):  
        return x / y  
    else:  
        #エラー処理
```

引数のキーワード指定

```
func('10t4065y', 'yamaki')
```

```
def func(name, id)
```

```
func(id='10t405y', name='yamaki')
```



引数のデフォルト値

```
>>> def hello(name="guest"):
```

```
...     print "hello, ", name
```

```
...
```

```
>>> hello() ← 省略可
```

```
hello, guest
```

```
>>> hello("yamaki")
```

```
hello, yamaki
```

引数の順序

デフォルト値を持った引数を、
デフォルト値を持たない引数の前に書くことは
できない

```
def func(name="guest", id): #エラー
```

```
def func(id, name="guest"): #OK
```

呼び出しも同様

```
func(name="yamaki", "10t4065y") #エラー
```

```
func("10t4065y", name="yamaki") #OK
```

ローカル変数

リストやディクショナリの扱いに注意

```
>>> def baz(localist):  
...     localist.append(100)  
...  
>>> a = [1, 2, 3]  
>>> baz(a)  
>>> a  
[1, 2, 3, 100]
```

練習問題

入れ子になっているリストを展開する関数を作ってください。

```
>>> a = [1, 2, [3, 4], [5, [6, 7]], [[8, 9], 10]]
```

```
>>> print extralist(a)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```