

1. Pythonの可変長引数（*args, **kwargs）の使い方

関数定義で引数に `*` と `**`（1個または2個のアスタリスク）を付けると、任意の数の引数（可変長引数）を指定することができる。

慣例として `*args` , `**kwargs` という名前が使われることが多いが、`*` と `**` が頭についていれば他の名前でも問題はない。

- `*args` =arguments: 複数の引数をタプルとして受け取る。
- `**kwargs` =keyword arguments: 複数のキーワード引数を辞書として受け取る。

1.1. *args: 複数の引数をタプルとして受け取る

`*args` のように `*` をつけた引数を定義すると、任意の数の引数を指定することができる。

```
def my_sum(*args):  
    print('args: ', args)  
    print('type: ', type(args))  
    print('sum : ', sum(args))  
  
print(my_sum(1, 2, 3, 4))  
# args: (1, 2, 3, 4)  
# type: <class 'tuple'>  
# sum : 10
```

位置引数と組み合わせることもできる.

位置引数より後ろ(右側)で指定した値が `args` にタプルとして渡される.

位置引数の場合は空のタプルになる.

```
def func_args(arg1, arg2, *args):  
    print('arg1: ', arg1)  
    print('arg2: ', arg2)  
    print('args: ', args)
```

```
func_args(0, 1, 2, 3, 4)
```

```
# arg1: 0  
# arg2: 1  
# args: (2, 3, 4)
```

```
func_args(0, 1):
```

```
# arg1: 0  
# arg2: 1  
# args: ()
```

1.2. **kwargs: 複数のキーワード引数を辞書として受け取る

`**kwargs` のように `**` をつけた引数を定義すると、任意の数のキーワード引数を指定することができる。

関数の中では引数名がキー `key`、値が `value` となる辞書として受け取られる。

```
def func_kwargs(**kwargs)
    print('kwargs: ', kwargs)
    print('type: ', type(kwargs))

func_kwargs(key1=1, key2=2, key3=3)
# kwargs: {'key1': 1, 'key2': 2, 'key3': 3}
# type: <class 'dict'>
```

関数呼び出し時に辞書オブジェクトに**をつけて引数に指定することで、展開してそれぞれの引数として渡すことも可能。

```
def func_kwargs_positional(arg1, arg2, **kwargs):  
    print('arg1: ', arg1)  
    print('arg2: ', arg2)  
    print('kwargs: ', kwargs)  
  
d = {'key1': 1, 'key2': 2, 'arg1': 100, 'arg2': 200}  
  
func_kwargs_positional(**d)  
# arg1: 100  
# arg2: 200  
# kwargs: {'key1': 1, 'key2': 2}
```