

Python 可变参数

刘鹏

2020 年 6 月 25 日

1 可变参数之数组参数 *args

这里介绍一下什么是 *args 可变参数。

```
[1]: def argsFunc(a, *args):  
      print(a)  
      print(args)  
  
      argsFunc(1, 2, 3, 4, 5)
```

```
1  
(2, 3, 4, 5)
```

2 可变参数之字典参数 **kwargs

形参名前加 ** 表示，参数在函数内部将被存放在以形式名为标识符的 dictionary 中，这时调用函数的方法则需要采用

```
arg1 = value1,  
arg2 = value2
```

这样的形式。

```
[2]: # 字典参数示例  
  
def dictParameterDemo(**args):  
    print(args)  
  
dictParameterDemo(a = 10, b = 20)
```

```
{'a': 10, 'b': 20}
```

下面列举一个很简单的例子来展示这种参数的某个使用场景。

这是一个不太优雅的实现

```
import mysql.connector

db_conf = {
    'user': 'xx',
    'password': 'yy',
    'host': 'xxx.xxx.xxx.xxx',
    'database': 'zz'
}

cnx = mysql.connector.connect(
    user=db_conf['user'],
    password=db_conf['password'],
    host=db_conf['host'],
    database=db_conf['database']
)
```

如果用字典不定参数就可以直接搞定。

```
import mysql.connector

db_conf = {
    user='xx',
    password='yy',
    host='xxx.xxx.xxx.xxx',
    database='zz'
}

cnx = mysql.connector.connect(**db_conf)
```

如果 `*args` 和 `**kwargs` 一起用会发生什么呢？

```
[3]: def together(*args, **kwargs):
      print(args)
      print(kwargs)
```

```
together(1, 2, 3, a=10, b = 10)
```

```
(1, 2, 3)
```

```
{'a': 10, 'b': 10}
```

```
[ ]:
```