

Python_模块中如何获取cmd调用的参数

通过sys中的sys.argv 是系统调用脚本的参数列表，在列表中第一项即为系统调用命令，第二项之后为调用脚本的参数
示例代码：

```
#coding=utf8
import sys
if __name__ == "__main__":
    print('argv:',sys.argv)
    a = input("press any key to continue")
```

通过命令行调用：

```
C:\Users\guoqingpei>python H:\workspace\DeveolpInPython\Decorator.py 1 2 3 4
argv: ['H:\workspace\DeveolpInPython\Decorator.py', '1', '2', '3', '4']
press any key to continue_
```

不过更加推荐：

python官方的argparse模块

用法实例：

```
import argparse
```

'''获取命令行参数，并把获取的参数输出想乘输出'''

```
if __name__ == "__main__":
    parser = argparse.ArgumentParser() #获取参数过滤器
    parser.add_argument("argA",type=int,help="this is the positional argA")#位置参数1(根据位置来取值)
    parser.add_argument("argB",type=int,help="this is the positional argB") #位置参数2(根据位置来取值)
    parser.add_argument("-argC","--C",type=int,default=1,help="this is the optinal factor")
    #可选参数，必须要有默认值，-argC是在命令行指定，--C中的C是指在程序中引用，help用于在输入-h选项时输出
    args = parser.parse_args() #实际从命令行中提取参数的过程
    print(args.argA * args.argB *args.C)
```

运行结果：

```
H:\workspace\DeveolpInPython>python python_SLOTS.py -h
usage: python_SLOTS.py [-h] [-argC C] argA argB
```

```
positional arguments:
  argA          this is the positional argA
  argB          this is the positional argB

optional arguments:
  -h, --help      show this help message and exit
  -argC C, --C C  this is the optinal factor
```

```
H:\workspace\DeveolpInPython>python python_SLOTS.py 2 3
6
```

```
H:\workspace\DeveolpInPython>python python_SLOTS.py 2 3 -argC 2
12
```