[**OptionParser命令参数介绍及使用**](https://www.cnblogs.com/blacksunny/p/5315545.html)

**使用optionParse解析命令行参数分以下几个步骤：**

1. **创建parser实例**
2. 使用add\_option添加我们要处理的命令行参数
3. 得到解析sys.argv后的options对象，查看用户的输入

**代码展示**

**from optparse import OptionParser**

**parser = OptionParser(...)**

**parser.add\_option(.....)**

**参数介绍**

* **OptionParser不要求一定要传递参数**

|  |  |
| --- | --- |
| 1  2  3  4  5 | OptionParser(usage="%prog",version="%prog 1.0",description="hello")      #%prog 在这里会自动替换为程序名字      #usage 可以打印用法      #version 在使用%prog --version的时候输出版本信息      #description 描述信息 |

* 使用add\_option添加命令行参数

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | #action  指示optparser解析参数时候该如何处理。默认是'store'，表示将命令行参数值保存options    对象里 。<br>action的值有：store,store\_true,store\_false,store\_const,append,count,callback.  #type 默认是“string",也可以是"int","float"等  #dest 如果没有指定dest参数，将用命令行参数名来对options对象的值进行存取。  #store store可以为store\_true和store\_false两种形式。用于处理命令行参数后面不带值的情况。如-v,-q等命令行参数。  #default 设置默认值  #help    指定帮助文档  #metavar 提示用户期望参数 |

* 最后调用parse\_args()解析命令行形参

|  |  |
| --- | --- |
| 1  2  3  4  5 | （options, args) = parser.parse\_args()   可以传递一个参数列表给parse\_args(),否则，默认使用命令行参数（sysargv[1:])。  parse\_args()返回两个值：      options, 这是一个对象（optpars.Values)，保存有命令行参数值。只要知道命令行参数名，如file,就可以访问其对应的值：options.file。      args，一个由positional arguments组成的列表。 |