**argparse模块(arg-parse)**

parser = argparse.ArgumentParser(description="your script description")

#创建一个解析对象

parser.add\_argument()

#向该对象中添加你要关注的命令行参数和选项

args = parser.parse\_args()

#进行解析

add\_argument(name or flags…[, action][, nargs][, const][, default][, type][, choices]

[, required][, help][, metavar][, dest])

#name or flags：指定参数的形式，一般写两个（-短参数 --长参数）；赋值时，-短参数 X

--长参数 = X

#nargs ：指定这个参数后面的value有多少个，例如，我们希望使用-n 1 2 3 4，来设置n的值为[1, 2, 3, 4] #parser.add\_argument(“-n”, “–num”, nargs=”+”, type=int)

#type：指定参数类型

#**choices**：**设置参数值的范围**

#**required**：**是否必选**

#metavar ：参数的名字，在显示帮助信息时才用到

#help：设置帮助信息

#**dest**：解析出来的**对应属性**

#**action**：指定属性对应的**处理逻辑**，默认为store（存储某个值或多个参数值收集到一个列表），可以为append,store\_true或者其它

#default：如果命令行没有出现这个选项，那么使用default指定的默认值

1. **import** argparse
3. parser = argparse.ArgumentParser(description='Process some integers.')
4. parser.add\_argument('integers', metavar='N', type=int, nargs='+',
5. help='an integer for the accumulator')
6. parser.add\_argument('-sum', dest='accumulate', **action='store\_const',**
7. **const=sum, default=max**,
8. help='sum the integers (default: find the max)')
9. parser.add\_argument("-square", help="display a square of a given number",
10. type=int)
12. args = parser.parse\_args()
13. **print**(args.accumulate(args.integers), args.square\*\*2)

>>>python prog.py 4 5 6 -square 7

结果： 6 49

>>>python prog.py -sum 4 5 6 --square 7

结果：15 49

**import** argparse

parser **=** argparse**.**ArgumentParser(description**=**'Search some files')

parser**.**add\_argument(dest**=**'filenames',metavar**=**'filename', nargs**=**'\*')

parser**.**add\_argument('-p', '--pat',metavar**=**'pattern', **required=True**,

**dest='patterns', action='append'**,

help**=**'text pattern to search for')

**#下面的参数说明允许某个参数重复出现多次，并将它们追加到一个列表中去**

parser**.**add\_argument('-v', dest**=**'verbose', action**=**'store\_true',

help**=**'verbose mode')

**#根据参数是否存在来设置一个 Boolean 标志**

parser**.**add\_argument('-o', dest**=**'outfile', action**=**'store',

help**=**'output file')

**#参数接受一个单独值并将其存储为一个字符串**

parser**.**add\_argument('--speed', dest**=**'speed', action**=**'store',

**choices={'slow','fast'}**, default**=**'slow',

help**=**'search speed')

**#接受一个值，但是会将其和可能的选择值做比较，以检测其合法性**

args **=** parser**.**parse\_args()

**print**(args**.**filenames)

**print**(args**.**patterns)

**print**(args**.**verbose)

**print**(args**.**outfile)

**print**(args**.**speed)

>>>python3 search**.**py **-**v -p spam --pat=eggs foo**.**txt bar**.**txt

filenames **=** ['foo.txt', 'bar.txt']

patterns **=** ['spam', 'eggs']

verbose **=** True

outfile **=** None

speed **=** slow

*#p3 fileColor .nwk***import** sys  
**import** argparse  
  
parser = argparse.ArgumentParser(description=**"change strain color in nwk file"**)  
group.add\_argument(**"fileColor"**,type=str,help=**"file contains: strainFileName and corresponding color"**)  
group.add\_argument(**"nwkFile"**,type=str,help=**"rerooted nwk file"**)  
args = parser.parse\_args()  
  
strainColor={}  
colorDict = {**"green"**:**"#00ff00"**,**"red"**:**"#990000"**,**"purple"**:**"#990099"**,**"blue"**:**"#0000ff"**,  
 **"bluegreen"**:**"#00ffcc"**,**"yellow"**:**"#ffff00"**,**"orange"**:**"#ff6600"**}  
  
**for** line **in** open(args.fileColor).readlines():  
 lineOfList = line.rstrip().split()  
 strainFile = lineOfList[0]  
 colorIndex = colorDict.get(lineOfList[1])  
 **for** line **in** open(strainFile).readlines():  
 line = line.rstrip()  
 strainColor[line] = line+**"[&!color="**+colorIndex+**"]"  
  
for** line **in** open(args.nwkFile).readlines():  
 **if** line.strip() **in** strainColor.keys():  
 print(**"\t"**+strainColor[line.strip()])  
 **else**:  
 print(line.rstrip())

**docopt模块（doc-option）**

代码的最开头使用 """ """文档注释的形式写出符合要求的**文档**，就会**自动生成**对应的parse

注释文档的格式

1）Usage：和一个空行之间的文本都会被识别为一个命令组合， **Usage 后的第一个字母将会被识别为这个程序的名字**，arguments = docopt(\_\_doc\_\_)后会在程序中导入一个arguments字典，所有参数取值为True或者False（借此可以把options的选项提取出来）

2）Options：可选项

*"""  
Usage:  
 tickets [-gdtkz] <from> <to> <date>  
  
Options:  
 -h,--help 显示帮助菜单  
 -g 高铁  
 -d 动车  
 -t 特快  
 -k 快速  
 -z 直达  
  
"""*