**argparse.ArgumentParser()用法解析**

argparse是一个Python模块：命令行选项、参数和子命令解析器。

主要有三个步骤：

创建 ArgumentParser() 对象

调用 add\_argument() 方法添加参数

使用 parse\_args() 解析添加的参数

**1. 创建解析器**

parser = argparse.ArgumentParser(prog=None, usage=None, description=None, epilog=None, parents=[], formatter\_class=argparse.HelpFormatter, prefix\_chars='-', fromfile\_prefix\_chars=None, argument\_default=None, conflict\_handler='error', add\_help=True, allow\_abbrev=True)

prog - 程序的名称（默认：sys.argv[0]）

usage - 描述程序用途的字符串（默认值：从添加到解析器的参数生成）

description - 在参数帮助文档之前显示的文本（默认值：无）

epilog - 在参数帮助文档之后显示的文本（默认值：无）

parents - 一个 ArgumentParser 对象的列表，它们的参数也应包含在内

formatter\_class - 用于自定义帮助文档输出格式的类

prefix\_chars - 可选参数的前缀字符集合（默认值：’-’）

fromfile\_prefix\_chars - 当需要从文件中读取其他参数时，用于标识文件名的前缀字符集合（默认值：None）

argument\_default - 参数的全局默认值（默认值： None）

conflict\_handler - 解决冲突选项的策略（通常是不必要的）

add\_help - 为解析器添加一个 -h/–help 选项（默认值： True）

allow\_abbrev - 如果缩写是无歧义的，则允许缩写长选项 （默认值：True）

**2. 添加参数**

parser = add\_argument(name or flags...[, action][, nargs][, const][, default][, type][, choices][, required][, help][, metavar][, dest])

name or flags - 一个命名或者一个选项字符串的列表，例如 foo 或 -f, --foo。

action - 当参数在命令行中出现时使用的动作基本类型。

nargs - 命令行参数应当消耗的数目。

const - 被一些 action 和 nargs 选择所需求的常数。

default - 当参数未在命令行中出现时使用的值。

type - 命令行参数应当被转换成的类型。

choices - 可用的参数的容器。

required - 此命令行选项是否可省略 （仅选项可用）。

help - 一个此选项作用的简单描述。

metavar - 在使用方法消息中使用的参数值示例。

dest - 被添加到 parse\_args() 所返回对象上的属性名。

**3. 解析参数**

args = parser.parse\_args()

**示例：**

# -\*- coding: utf-8 -\*-

import argparse

parser = argparse.ArgumentParser()

parser.add\_argument("square", help="display a square of a given number", type=int)

args = parser.parse\_args()

print(args.square\*\*2)

将上面的代码保存为文件 argparse\_usage.py，在终端运行，结果如下：

$ python argparse\_usage.py 9

81