Python命令行

参数管理

通过`sys.argv[1:]`获取有效参数，通过`getopt.getopt()`来管理参数

```
import sys
import getopt

# 第0个参数是 xx.py本身
print("全部sys.argv为:", sys.argv)

# 我们需要的参数从1开始，注意：解析出来的是，按照空格隔开的列表
arguments = sys.argv[1:]
print("我们需要的参数是:", arguments)

# 通过getopt进行参数管理，接收三个参数：
# 1. 当前脚本的参数
# 2. 短参数
# 3. 长参数
arg = getopt.getopt(sys.argv[1:], '-h', ['host'])
# arg = getopt.getopt(sys.argv[1:], '-h', ['host='])
print("通过getopt解析结果是:", arg)
```

短参数

有的参数不需要带值，比如：`-v` 版本号 `-h`帮助信息 等；而有些参数参数需要带参数，比如：`-p 8080`指定端口 ，所以针对这些参数的管理也需要做区分

不带值

使用`arg = getopt.getopt(sys.argv[1:], '-h', ['host'])`管理参数

1. 直接`-h`来传参
2. 不支持`-h`以外的特殊参数，比如传：`-he`就会报错
3. 支持传入多个其他的普通参数，但不跟`-h`绑定

```
(mypython3-Xz6Q90qj) bash-3.2$ python py_script_pram.py -h
全部sys.argv为: ['py_script_pram.py', '-h']
我们需要的参数是: ['-h']
通过getopt解析结果是: ([('-h', '')], [])

(mypython3-Xz6Q90qj) bash-3.2$ python py_script_pram.py -h arg1 arg2
全部sys.argv为: ['py_script_pram.py', '-h', 'arg1', 'arg2']
我们需要的参数是: ['-h', 'arg1', 'arg2']
通过getopt解析结果是: ([('-h', '')], ['arg1', 'arg2'])
```

带值

使用`arg = getopt.getopt(sys.argv[1:], '-h:', ['host'])`管理参数，注意`-h:`多了个冒号

1. 必须通过`-h{空格}{参数}`来传值，如果不传参数会报错，`-h`和`{参数}`是绑定关系
2. 不接收`-h`以外的传参方式，比如传`-d`就会报错
3. 不能使用`-h=1`传参，否则报错
4. 支持传入多个其他的普通参数，但只有第一个跟`-h`绑定

```
(mypython3-Xz6Q90qj) bash-3.2$ python py_script_pram.py -h 1
全部sys.argv为: ['py_script_pram.py', '-h', '1']
我们需要的参数是: ['-h', '1']
通过getopt解析结果是: ([('-h', '1')], [])

(mypython3-Xz6Q90qj) bash-3.2$ python py_script_pram.py -h 1 2
全部sys.argv为: ['py_script_pram.py', '-h', '1', '2']
我们需要的参数是: ['-h', '1', '2']
通过getopt解析结果是: ([('-h', '1')], ['2'])
```

长参数

不带值

使用`arg = getopt.getopt(sys.argv[1:], '-h', ['host'])`方式管理

1. `--host`直接作为参数
2. 不接收`--host`以外的长参数，比如传：`--hosts`就会报错
3. 支持传入多个其他的普通参数，但不跟`--host`绑定

```
(mypython3-Xz6Q90qj) bash-3.2$ python py_script_pram.py --host
全部sys.argv为: ['py_script_pram.py', '--host']
我们需要的参数是: ['--host']
通过getopt解析结果是: ([('--host', '')], [])

(mypython3-Xz6Q90qj) bash-3.2$ python py_script_pram.py --host 123 456
全部sys.argv为: ['py_script_pram.py', '--host', '123', '456']
我们需要的参数是: ['--host', '123', '456']
通过getopt解析结果是: ([('--host', '')], ['123', '456'])
```

带值

使用`arg = getopt.getopt(sys.argv[1:], '-h', ['host='])`方式管理

1. 支持`--host{空格}{参数}`传参，第一个参数和`host`绑定
2. 支持`--host={参数}`传参，和`host`绑定
3. 不支持传空`--host=`，会报错
4. 不接收`--host`以外的传参方式，比如：`--hosts`会报错
5. 支持传入多个其他的普通参数，但只有第一个参数和`--host`绑定

```
(mypython3-Xz6Q90qj) bash-3.2$ python py_script_pram.py --host 1
全部sys.argv为: ['py_script_pram.py', '--host', '1']
我们需要的参数是: ['--host', '1']
通过getopt解析结果是: ([('--host', '1')], [])

(mypython3-Xz6Q90qj) bash-3.2$ python py_script_pram.py --host=1
全部sys.argv为: ['py_script_pram.py', '--host=1']
我们需要的参数是: ['--host=1']
通过getopt解析结果是: ([('--host', '1')], [])

(mypython3-Xz6Q90qj) bash-3.2$ python py_script_pram.py --host=1 2 3
全部sys.argv为: ['py_script_pram.py', '--host=1', '2', '3']
我们需要的参数是: ['--host=1', '2', '3']
通过getopt解析结果是: ([('--host', '1')], ['2', '3'])
```

小结

短参数不带参数：`-h`

长参数不带参数：`--host`

短参数带参数：`-h:`；可以不传整个`-h`，但如果要传`-h`必须要绑定参数

长参数带参数：`--host=`；可以不传整个`--host`，也可以使用`-host=`传空

参数解析

案例：脚本可以指定启动地址和监听端口，例如：`python py_script_pram_many.py --host=127.0.0.1 -p 5001`

```
import sys
import getopt

opts, args = getopt.getopt(sys.argv[1:], '-h:-p:', ['host=', 'port='])
for opt_name, opt_value in opts:
    if opt_name in ('-h', '--host'):
        host = opt_value
        print("host address is:", host)
    if opt_name in ('-p', '--port'):
        port = opt_value
        print("port is:", port)
```

执行结果：

```
(mypython3-Xz6Q90qj) bash-3.2$ python py_script_pram_many.py --host=127.0.0.1
-p 8080
host address is: 127.0.0.1
port is: 8080
```