

# day04

---

python官方帮助文档

<https://docs.python.org/zh-cn/3/> -> 标准库参考

搜狗翻译 -> <https://fanyi.sogou.com/>

## 查看帮助

```
>>> import shutil
>>> help(shutil)
>>> help(shutil.copy)
```

## subprocess模块

### 常用方法

```
>>> import subprocess
>>> subprocess.run('ls') # ls
>>> subprocess.run(['ls', '/home']) # 没有shell环境
>>> subprocess.run('ls /home') # FileNotFoundError

# 在shell环境中运行ls /home
>>> subprocess.run('ls /home', shell=True)
```

没有shell环境，就没有环境变量、命令扩展

```
>>> rc = subprocess.run(['ls', '~'])
ls: 无法访问~: 没有那个文件或目录
CompletedProcess(args=['ls', '~'], returncode=2)
>>> rc.returncode # 相当于 $?
2
>>> rc = subprocess.run('ls ~', shell=True)
>>> rc.returncode
```

### 获取输出

可以通过subprocess.PIPE将命令的错误和输出保存到stderr和stdout中。这两个参数是bytes类型。

```
>>> rc = subprocess.run('id root; id wangwu', shell=True,
stderr=subprocess.PIPE, stdout=subprocess.PIPE)
>>> rc.args # 执行的指令
'id root; id wangwu'
>>> rc.stderr
b'id: wangwu: no such user\n'
>>> rc.stdout
b'uid=0(root) gid=0(root) \xe7\xbb\x84=0(root)\n'
>>> rc.stdout.decode()
'uid=0(root) gid=0(root) 组=0(root)\n'
```

## 字符串分为bytes类型和str类型

```
>>> s1 = '达内'
>>> type(s1)
<class 'str'>
>>> b1 = s1.encode() # 编码成bytes类型，使用utf8编码
>>> b1
b'\xe8\xbe\xbe\xe5\x86\x85'
>>> type(b1)
<class 'bytes'>
>>> b1.decode() # 解码成str类型
'达内'
```

## 语法

```
>>> x = y = 10
>>> a, b = 10, 20
>>> a
10
>>> b
20
>>> c, d = (10, 20)
>>> c
10
>>> d
20
>>> e, f = [10, 20]
>>> e
10
>>> f
20
>>> a, b = b, a # 将a、b的值互换
```

## 关键字

```
>>> import keyword
>>> keyword.kwlist
['False', 'None', 'True', 'and', 'as', 'assert', 'break',
'class', 'continue', 'def', 'del', 'elif', 'else',
'except', 'finally', 'for', 'from', 'global', 'if',
'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or',
'pass', 'raise', 'return', 'try', 'while', 'with',
'yield']
```

## 内建

内建不是关键字，能够被覆盖，但是不推荐这么做

### 内置函数

```
>>> len('abcd')
4
>>> len = 10
>>> len
10
>>> len('abcd') # 现在len是10
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'int' object is not callable
>>> 10('abcd') # 数字是不能被调用的
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'int' object is not callable
```

## python代码布局

```
#!/usr/bin/python3
'文档字符串，用于在帮助中显示'

# 模块导入部分
import random
import time

# 全局变量定义
all_chs = '0123456789qwertyuiopasdfghjklzxcvbnm'

# 类的定义
class MyClass:
    pass

# 函数定义
def func():
```

```
pass

# 程序主体代码
if __name__ == '__main__':
    mc = MyClass()
    func()
```

## 编程思路

以创建文件为例

1. 发呆。思考程序的运行方式：交互式？非交互式？

```
文件名： /etc/hosts
文件已存在，请重试
文件名： /tmp/mytest.txt
请输入内容，输入END结束
> Hello World!
> 2nd line.
> 3rd line.
> END
# cat /tmp/mytest.txt
Hello World!
2nd line.
3rd line.
```

2. 思考程序有哪些功能，把这些功能编写成功能函数
3. 编写程序主体代码，依次调用相关函数
4. 编写函数的具体代码

## 序列对象

```
>>> str(10)
'10'
>>> list('abcd')
['a', 'b', 'c', 'd']
>>> tuple('abcd')
('a', 'b', 'c', 'd')
>>> alist = [10, 20, 30, 40]
>>> tuple(alist)
(10, 20, 30, 40)
```

## enumerate方法

```
>>> alist = ['tom', 'jerry', 'bob', 'alice']
>>> enumerate(alist)
<enumerate object at 0x7f8afbb99e58>
>>> list(enumerate(alist))
[(0, 'tom'), (1, 'jerry'), (2, 'bob'), (3, 'alice')]
```

## reversed方法

```
>>> import random
>>> alist = [random.randint(1, 100) for i in range(10)]
>>> alist
[1, 66, 61, 91, 37, 71, 92, 85, 31, 95]
>>> reversed(alist)
<list_reverseiterator object at 0x7f8afcb3eeb8>
>>> list(reversed(alist))
[95, 31, 85, 92, 71, 37, 91, 61, 66, 1]
>>> for i in reversed(alist):
...     print(i)
```

## sorted方法

```
>>> sorted(alist)
[1, 31, 37, 61, 66, 71, 85, 91, 92, 95]
>>> sorted(alist, reverse=True)
[95, 92, 91, 85, 71, 66, 61, 37, 31, 1]
```

## 查看字符的编码值

```
>>> ord('a')
97
>>> ord('A')
65
>>> ord('中')
20013
```

## 字符串格式化操作符

```
>>> '%s is %s years old' % ('bob', 22)
'bob is 22 years old'
>>> 'I am %s' % 'tom'    # 只有一个%s, %后面的数据不用写到元组
'I am tom'
>>> '%s is %d years old' % ('bob', 22)  # 整数可以用%d
'bob is 22 years old'
>>> '%d is %d years old' % ('bob', 22)  # 字符串不能用%d
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
```

```

TypeError: %d format: a number is required, not str
>>> '%10s%8s' % ('name', 'age') # name占10列, age占8列
'      name      age'
>>> '%10s%8s' % ('tom', 22)
'      tom       22'
>>> '%-10s%-8s' % ('name', 'age') # 负数表示左对齐
'name      age      '
>>> '%-10s%-8s' % ('tom', 22)
'tom       22       '

```

其他简单了解的格式化方法

```

>>> '97: %c' % 97
'97: a'
>>> '11: %#0' % 11
'11: 0o13'
>>> '11: %#x' % 11
'11: 0xb'
>>> '5 / 3 = %f' % (5 / 3)
'5 / 3 = 1.666667'
>>> '5 / 3 = %.2f' % (5 / 3)
'5 / 3 = 1.67'
>>> '%10d' % 5
'          5'
>>> '%010d' % 5
'0000000005'

```

## format方法

也用于实现字符串的格式化，与%s/%d类似

```

>>> '{} is {} years old'.format('bob', 22)
'bob is 22 years old'
>>> '{} is {} years old'.format(22, 'bob')
'22 is bob years old'
>>> '{1} is {0} years old'.format(22, 'bob')
'bob is 22 years old'

# 第0个位置的数据是列表，根据第0个位置的下标取值
>>> '{0[0]} is {0[1]} years old'.format(['bob', 22])
'bob is 22 years old'

```

