[Top](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/PYTHON1/DAY03/CASE/01/index.html" \l "page_top_case)

# NSD Python1 DAY03

1. [案例1：模拟cp操作](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/PYTHON1/DAY03/CASE/01/index.html" \l "case1)
2. [案例2：生成随机密码](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/PYTHON1/DAY03/CASE/01/index.html" \l "case2)

## 1 案例1：模拟cp操作

### 1.1 问题

创建cp.py文件，实现以下目标：

1. 将/bin/ls“拷贝”到/root/目录下
2. 不要修改原始文件

### 1.2 方案

获取用户原文件名和新文件名，打开原文件，打开新文件，从打开的原文件中读取数据，写入到打开的新文件中，关闭两个文件

cp代码的过程中，需要注意的部分在于：

如果一个文件过大，你将无法直接读取数据到内存，此时，使用while循环语句，分次读取数据，每次读4096字节，读取数据为空时，结束循环，将数据写入到目标文件

### 1.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：编写脚本

1. [root@localhost day03]# vim cp.py
2. #!/usr/bin/env python3
3. f1 = open('/bin/ls', 'rb')
4. f2 = open('/root/ls', 'wb')
5. data = f1.read()
6. f2.write(data)
7. f1.close()
8. f2.close()

或将上面的代码改为以下写法：

循环读取文件中数据，避免读取数据过大

1. [root@localhost day03]# vim cp2.py
2. #!/usr/bin/env python3
3. src\_fname = '/bin/ls'
4. dst\_fname = '/tmp/ls'
5. src\_fobj = open(src\_fname, 'rb')
6. dst\_fobj = open(dst\_fname, 'wb')
7. while True:
8. data = src\_fobj.read(4096) # 每次读4096字节
9. if data == b'': # 读不到数据意味着读写完毕，中断循环
10. break
11. dst\_fobj.write(data) # 将数据写到目标文件
12. src\_fobj.close()
13. dst\_fobj.close()

或将上面的代码改为以下写法：

With打开文件读取数据或写入数据后，文件会直接关闭

1. [root@localhost day03]# vim cp3.py
2. #!/usr/bin/env python3
3. src\_fname = '/bin/ls'
4. dst\_fname = '/root/ls'
5. with open(src\_fname, 'rb') as src\_fobj:
6. with open(dst\_fname, 'wb') as dst\_fobj:
7. while True:
8. data = src\_fobj.read(4096)
9. if not data:
10. break
11. dst\_fobj.write(data)

或将上面的代码改为以下写法：

sys.argv方法表示空列表，执行脚本时输入命令： python3 cp\_func.py /bin/ls /root/ls，表示sys.argv=[cp\_func.py，‘/bin/ls’，‘/root/ls’]，所以，调用copy函数时，列表切片方式获取实参为（‘/bin/ls’，‘/root/ls’）

1. [root@localhost day03]# vim cp\_func.py
2. #!/usr/bin/env python3
3. import sys
4. def copy(src\_fname, dst\_fname):
5. src\_fobj = open(src\_fname, 'rb')
6. dst\_fobj = open(dst\_fname, 'wb')
7. while True:
8. data = src\_fobj.read(4096)
9. if not data:
10. break
11. dst\_fobj.write(data)
12. src\_fobj.close()
13. dst\_fobj.close()
14. copy(sys.argv[1], sys.argv[2])

步骤二：测试脚本执行

1. [root@localhost day03]# python3 cp.py
2. [root@localhost day03]# cd /root
3. [root@localhost ~]# ls
4. core ls
5. [root@localhost day03]# python3 cp2.py
6. [root@localhost day03]# cd /root
7. [root@localhost ~]# ls
8. core ls
9. [root@localhost day03]# python3 cp3.py
10. [root@localhost day03]# cd /root
11. [root@localhost ~]# ls
12. core ls
13. [root@localhost day03]# python3 cp\_func.py /bin/ls /root/ls
14. [root@localhost day03]# cd /root
15. [root@localhost ~]# ls
16. core ls

## 2 案例2：生成随机密码

### 2.1 问题

创建randpass.py脚本，要求如下：

1. 编写一个能生成8位随机密码的程序
2. 使用random的choice函数随机取出字符
3. 改进程序，用户可以自己决定生成多少位的密码

### 2.2 方案

导入random模块，通过random静态对象调用choice()方法，从自定义字符串all\_chs中获取随机项，将获取到的随机字符ch与原result值进行拼接，将最终字符串结果返回给函数，for循环每循环一次获取一个随机字符，密码位数由循环次数决定，循环次数由传递参数值决定。

此程序需要注意的部分在于：

1.导入String模块，其中ascii\_letters是生成所有字母，从a-z和A-Z，digits是生成所有数字0-9

2.将整个生成随机密码的代码封装进gen\_pass()函数中，当模块文件直接执行时，调用函数即可输出结果

3.参数传递问题：调用函数无实参时，函数调用默认参数，有实参时，函数调用实际参数

### 2.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：编写脚本

1. [root@localhost day03]# vim randpass.py
2. #!/usr/bin/env python3
3. import random
4. import string
5. all\_chs = string.digits + string.ascii\_letters
6. def gen\_pass(n=8):
7. result = ''
8. for i in range(n):
9. ch = random.choice(all\_chs)
10. result += ch
11. return result
12. if \_\_name\_\_ == '\_\_main\_\_':
13. print(gen\_pass())
14. print(gen\_pass(4))

或将上面的代码改为以下写法：

利用列表推导式更简洁输出数据

1. [root@localhost day03]# vim randpass2.py
2. #!/usr/bin/env python3
3. from random import choice
4. from string import ascii\_letters, digits
5. all\_chs = ascii\_letters + digits
6. def randpass(n=8):
7. result = [choice(all\_chs) for i in range(n)]
8. return ''.join(result) # 将列表的字符拼接起来
9. if \_\_name\_\_ == '\_\_main\_\_':
10. print(randpass())
11. print(randpass(4))

步骤二：测试脚本执行

1. [root@localhost day03]# python3 randpass.py
2. 82wi2gOP
3. XzMi
4. [root@localhost day03]# python3 randpass.py
5. 5wMoDEgC
6. BDpc
7. [root@localhost day03]# python3 randpass.py
8. Ige2VGod
9. Az0z
10. [root@localhost day03]# python3 randpass2.py
11. eajAocMH
12. edW1