[Top](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/PYTHON1/DAY05/CASE/01/index.html" \l "page_top_case)

# NSD Python1 DAY05

1. [案例1：用列表构建栈结构](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/PYTHON1/DAY05/CASE/01/index.html" \l "case1)
2. [案例2：模拟用户登陆信息系统](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/PYTHON1/DAY05/CASE/01/index.html" \l "case2)
3. [案例3：比较文件内容](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/PYTHON1/DAY05/CASE/01/index.html" \l "case3)

## 1 案例1：用列表构建栈结构

### 1.1 问题

创建stack.py脚本，要求如下：

1. 栈是一个后进先出的结构
2. 编写一个程序，用列表实现栈结构
3. 需要支持压栈、出栈、查询功能

### 1.2 方案

创建空列表存储数据，创建4个函数，分别实现压栈、出栈、查询以及判断函数调用的方法。

此程序需要注意的是堆栈的结构特点，先进后出，后进先出：

1.调用show\_menu()函数后，利用while循环交互端输出提示，请用户input0/1/2/3任意数值，如果输入的值不是0/1/2/3，打印输入值无效请重新输入并重新开始循环，如果输入的值是3，停止整个循环，如果输入的值是0/1/2通过字典键值对关联关系，调用相对应函数

2.如果输入的值是0，字典cmds中0键对应的值是push\_it，push\_it()调用压栈函数，压栈函数利用stack.append()方法将输入数据追加到列表结尾

3.如上，如果输入的值是1，调用出栈函数pop\_it()，出栈函数如果stack列表中有数据，弹出列表最后一个元素（根据堆栈结构特点stack.pop()中参数为空），如果stack列表没有数据，输出空列表

4.如果输入的值是2，调用查询函数view\_it()，显示当前列表

### 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

让输出的文字带颜色：\033[31;1m高亮度红色字体、\033[31;1m高亮度绿色字体、\033[0m关闭所有属性

1. [root@localhost day04]# vim stack.py
2. #!/usr/bin/env python3
3. stack = []
4. def push\_it():
5. item = input('item to push: ')
6. stack.append(item)
7. def pop\_it():
8. if stack:
9. print("\033[31;1mPopped %s\033[0m" % stack.pop())
10. else:
11. print('\033[31;1mEmpty stack\033[0m')
12. def view\_it():
13. print("\033[32;1m%s\033[0m" % stack)
14. def show\_menu():
15. prompt = """(0) push\_it
16. (1) pop\_it
17. (2) view\_it
18. (3) quit
19. Please input your choice(0/1/2/3): """
20. cmds = {'0': push\_it, '1': pop\_it, '2': view\_it}
21. while True:
22. # strip() 方法用于移除字符串头尾指定的字符（默认为空格）
23. choice = input(prompt).strip()[0]
24. if choice not in '0123':
25. print('Invalid input. Try again.')
26. continue #结束本次循环
28. if choice == '3':
29. break #结束整个循环
31. cmds[choice]() # push\_it() pop\_it()
32. # if choice == '0':
33. # push\_it()
34. # elif choice == '1':
35. # pop\_it()
36. # elif choice == '2':
37. # view\_it()
38. if \_\_name\_\_ == '\_\_main\_\_':
39. show\_menu()

步骤二：测试脚本执行

1. [root@localhost day04]# python3 stack.py
2. (0) push\_it
3. (1) pop\_it
4. (2) view\_it
5. (3) quit
6. Please input your choice(0/1/2/3): 6
7. Invalid input. Try again.
8. (0) push\_it
9. (1) pop\_it
10. (2) view\_it
11. (3) quit
12. Please input your choice(0/1/2/3): 0
13. item to push: nihao
14. (0) push\_it
15. (1) pop\_it
16. (2) view\_it
17. (3) quit
18. Please input your choice(0/1/2/3): 1
19. Popped nihao
20. (0) push\_it
21. (1) pop\_it
22. (2) view\_it
23. (3) quit
24. Please input your choice(0/1/2/3): 2
25. []
26. (0) push\_it
27. (1) pop\_it
28. (2) view\_it
29. (3) quit
30. Please input your choice(0/1/2/3): 0
31. item to push: a
32. (0) push\_it
33. (1) pop\_it
34. (2) view\_it
35. Please input your choice(0/1/2/3): 0
36. item to push: b
37. (0) push\_it
38. (1) pop\_it
39. (2) view\_it
40. (3) quit
41. Please input your choice(0/1/2/3): 0
42. item to push: c
43. (0) push\_it
44. (1) pop\_it
45. (2) view\_it
46. (3) quit
47. Please input your choice(0/1/2/3): 1
48. Popped c
49. (0) push\_it
50. (1) pop\_it
51. (2) view\_it
52. (3) quit
53. Please input your choice(0/1/2/3): 2
54. ['a', 'b']
55. (0) push\_it
56. (1) pop\_it
57. (2) view\_it
58. (3) quit
59. Please input your choice(0/1/2/3): 3
60. (3) quit

## 2 案例2：模拟用户登陆信息系统

### 2.1 问题

编写login.py脚本，实现以下目标：

1. 支持新用户注册，新用户名和密码注册到字典中
2. 支持老用户登陆，用户名和密码正确提示登陆成功
3. 主程序通过循环询问进行何种操作，根据用户的选择，执行注册或是登陆操作

### 2.2 方案

创建空字典存储用户名、密码，用三个函数分别实现用户注册、用户登录以及判断调用函数这三个方法，完成模拟用户登录：

1.调用show\_menu()函数后，利用while循环交互端输出提示，请用户input0/1/2任意数值，如果输入的值不是0/1/2，打印选择无效请重新输入并重新开始循环，如果输入的值是2，停止整个循环，如果输入的值是0/1/2通过字典键值对关联关系，调用相对应函数

2.如果输入的值是0，字典cmds中0键对应的值是register，register ()调用注册函数，函数利用if方法判断输入用户名是否存在，如果用户名在字典中，输出用户名已存在，否则用户输入密码，并将用户名与密码以键值对形式放入字典中

3.如上，如果输入的值是1，调用登录函数login()，利用if方法判断输入的用户名的对应的密码是否和字典中存储用户对应密码相同，如果不同显示登录失败，否则登录成功，此函数中导入getpass模块使用方法，作用是输入密码不可见。

### 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 day05]# vim login.py
2. #!/usr/bin/env python3
3. import getpass
4. userdb = {}
5. def register():
6. username = input('username: ')
7. if username in userdb:
8. print('\033[31;1m%s already exists.\033[0m' % username)
9. else:
10. password = input('password: ')
11. userdb[username] = password
12. def login():
13. username = input('username: ')
14. password = getpass.getpass('password: ')
15. # if username not in userdb or userdb['username'] != password:
16. if userdb.get(username) != password:
17. print('\033[31;1mLogin incorrect\033[0m')
18. else:
19. print('\033[32;1mLogin successful\033[0m')
20. def show\_menu():
21. prompt = """(0) register
22. (1) login
23. (2) quit
24. Please input your choice(0/1/2): """
25. cmds = {'0': register, '1': login}
26. while True:
27. choice = input(prompt).strip()[0]
28. if choice not in '012':
29. print('Invalid choice. Try again.')
30. continue
31. if choice == '2':
32. break
33. cmds[choice]()
34. if \_\_name\_\_ == '\_\_main\_\_':
35. show\_menu()

步骤二：测试脚本执行

1. [root@localhost day05]# python3 login.py
2. (0) register
3. (1) login
4. (2) quit
5. Please input your choice(0/1/2): 0
6. username: a
7. password: 123
8. (0) register
9. (1) login
10. (2) quit
11. Please input your choice(0/1/2): 1
12. username: a
13. password:
14. Login successful
15. (0) register
16. (1) login
17. (2) quit
18. Please input your choice(0/1/2): 1
19. username: b
20. password:
21. Login incorrect
22. (0) register
23. (1) login
24. (2) quit
25. Please input your choice(0/1/2): 2

## 3 案例3：比较文件内容

### 3.1 问题

编写程序，完成以下要求：

1. 有两个文件：a.log和b.log
2. 两个文件中有大量重复内容
3. 取出只有在b.log中存在的行

### 3.2 方案

使用集合去重效率高，操作简单。只要将两个文件内容转换成列表，再进行差补运算即可得到仅在b.log中存在的行。

步骤一：准备两个文件

1. [root@localhost day05]# cp /etc/passwd /tmp/a.log
2. [root@localhost day05]# cp /etc/passwd /tmp/b.log
3. # 修改b.log，使之与a.log有不同的行
4. [root@localhost day05]# vim /tmp/b.log

步骤二：编写程序

1. [root@localhost day05]# vim fdiff.py
2. with open('/tmp/a.log') as f1:
3. aset = set(f1)
4. with open('/tmp/b.log') as f2:
5. bset = set(f2)
6. with open('/tmp/result.txt', 'w') as f3:
7. f3.writelines(bset - aset)

步骤三：验证

1. [root@localhost day05]# cat /tmp/result.txt

查看结果，result.txt中将是只有b.log中有，而a.log没有的行。