

课程：函数

目标

- 函数的作用
- 函数的使用步骤
- 函数的参数作用
- 函数的返回值作用
- 函数的说明文档
- 函数嵌套

一. 函数的作用

需求：用户到ATM机取钱：

1. 输入密码后显示"选择功能"界面
2. 查询余额后显示"选择功能"界面
3. 取2000钱后显示"选择功能"界面

特点：显示“选择功能”界面需要重复输出给用户，怎么实现？



函数就是将一段具有独立功能的代码块整合到一个整体并命名，在需要的位置调用这个名称即可完成对应的需求。

函数在开发过程中，可以更高效率的实现代码重用。

二. 函数的使用步骤

2.1 定义函数

```
1 def 函数名(参数):
2     代码1
3     代码2
4     .....
```

2.2 调用函数

```
1 函数名(参数)
```

注意：

1. 不同的需求，参数可有可无。
2. 在Python中，函数必须先定义后使用。

2.3 快速体验

需求：复现ATM取钱功能。

1. 搭建整体框架(复现需求)

```
1 print('密码正确登录成功')
2
3 # 显示"选择功能"界面
4
5 print('查询余额完毕')
6
7 # 显示"选择功能"界面
8
9 print('取了2000元钱')
10
11 # 显示"选择功能"界面
```

2. 确定"选择功能"界面内容

```
1 print('查询余额')
2 print('存款')
3 print('取款')
```

3. 封装"选择功能"

注意：一定是先定义函数，后调用函数。

```

1  # 封装ATM机功能选项 -- 定义函数
2  def select_func():
3      print('-----请选择功能-----')
4      print('查询余额')
5      print('存款')
6      print('取款')
7      print('-----请选择功能-----')

```

4. 调用函数

在需要显示“选择功能”函数的位置调用函数。

```

1  print('密码正确登录成功')
2  # 显示"选择功能"界面 -- 调用函数
3  select_func()
4
5  print('查询余额完毕')
6  # 显示"选择功能"界面 -- 调用函数
7  select_func()
8
9  print('取了2000元钱')
10 # 显示"选择功能"界面 -- 调用函数
11 select_func()

```

三.函数的参数作用

思考：完成需求如下：一个函数完成两个数1和2的加法运算，如何书写程序？

```

1  # 定义函数
2  def add_num1():
3      result = 1 + 2
4      print(result)
5
6
7  # 调用函数
8  add_num1()

```

思考：上述add_num1函数只能完成数字1和2的加法运算，如果想要这个函数变得更灵活，可以计算任何用户指定的两个数字的和，如何书写程序？

分析：用户要在调用函数的时候指定具体数字，那么在定义函数的时候就需要接收用户指定的数字。函数调用时候指定的数字和定义函数时候接收的数字即是函数的参数。

```
1 # 定义函数时同时定义了接收用户数据的参数a和b，a和b是形参
2 def add_num2(a, b):
3     result = a + b
4     print(result)
5
6
7 # 调用函数时传入了真实的数据10 和 20，真实数据为实参
8 add_num2(10, 20)
```

四.函数的返回值作用

例如：我们去超市购物，比如买烟，给钱之后，是不是售货员会返回给我们烟这个商品，在函数中，如果需要返回结果给用户需要使用函数返回值。

```
1 def buy():
2     return '烟'
3
4 # 使用变量保存函数返回值
5 goods = buy()
6 print(goods)
```

4.1 应用

需求：制作一个计算器，计算任意两数字之和，并保存结果。

```
1 def sum_num(a, b):
2     return a + b
3
4
5 # 用result变量保存函数返回值
6 result = sum_num(1, 2)
7 print(result)
```

五.函数的说明文档

思考：定义一个函数后，程序员如何书写程序能够快速提示这个函数的作用？

答：注释

思考：如果代码多，我们是不是需要在很多代码中找到这个函数定义的位置才能看到注释？如果想更方便的查看函数的作用怎么办？

答：函数的说明文档

函数的说明文档也叫函数的文档说明。

5.1 语法

- 定义函数的说明文档

```
1 def 函数名(参数):  
2     """ 说明文档的位置 """  
3     代码  
4     .....
```

- 查看函数的说明文档

```
1 help(函数名)
```

5.2 快速体验

```
1 def sum_num(a, b):  
2     """ 求和函数 """  
3     return a + b  
4  
5  
6 help(sum_num)
```

```
C:\Users\黑马程序员\AppData\Local\Programs\Python\Python37\python3.exe  
Help on function sum_num in module __main__:  
  
sum_num(a, b)  
    求和函数  
  
Process finished with exit code 0
```

六.函数嵌套调用

所谓函数嵌套调用指的是一个函数里面又调用了另外一个函数。

- 示例

```

1  def testB():
2      print('---- testB start----')
3      print('这里是testB函数执行的代码...(省略)...')
4      print('---- testB end----')
5
6  def testA():
7      print('---- testA start----')
8      testB()
9      print('---- testA end----')
10
11 testA()

```

- 效果

```

C:\Users\黑马程序员\AppData\Local\Programs\Python\Python37\python3.exe
---- testA start----
---- testB start----
这里是testB函数执行的代码...(省略)...
---- testB end----
---- testA end----

Process finished with exit code 0

```

- 执行流程



```

def testB():
    print('---- testB start----')
    print('这里是testB函数执行的代码...(省略)...')
    print('---- testB end----')

def testA():
    print('---- testA start----')

    testB()

    print('---- testA end----')

testA()

```

- 如果函数A中，调用了另外一个函数B，那么先把函数B中的任务都执行完毕之后才会回到上次 函数A执行的位置。

七. 函数应用

7.1 打印图形

1. 打印一条横线

```
1 def print_line():
2     print('-' * 20)
3
4
5 print_line()
```

```
C:\Users\黑马程序员\AppData\Local\Programs\Python\Python37\python3.exe
-----

Process finished with exit code 0
```

2. 打印多条横线

```
1 def print_line():
2     print('-' * 20)
3
4
5 def print_lines(num):
6     i = 0
7     while i < num:
8         print_line()
9         i += 1
10
11
12 print_lines(5)
```

```
C:\Users\黑马程序员\AppData\Local\Programs\Python\Python37\python3.exe
-----
-----
-----
-----
-----

Process finished with exit code 0
```

7.2 函数计算

1. 求三个数之和

```
1 def sum_num(a, b, c):
2     return a + b + c
3
4
5 result = sum_num(1, 2, 3)
6 print(result) # 6
```

2. 求三个数平均值

```
1 def average_num(a, b, c):
2     sumResult = sum_num(a, b, c)
3     return sumResult / 3
4
5 result = average_num(1, 2, 3)
6 print(result) # 2.0
```

八. 总结

- 函数的作用：封装代码，高效的代码重用
- 函数使用步骤
 - 定义函数

```
1 def 函数名():
2     代码1
3     代码2
4     ...
```

- 调用函数

```
1 函数名()
```

- 函数的参数：函数调用的时候可以传入真实数据，增大函数的使用的灵活性
 - 形参：函数定义时书写的参数(非真实数据)
 - 实参：函数调用时书写的参数(真实数据)
- 函数的返回值
 - 作用：函数调用后，返回需要的计算结果
 - 写法

```
1 return 表达式
```

- 函数的说明文档
 - 作用：保存函数解释说明的信息

- 写法

```
1 def 函数名():  
2     """ 函数说明文档 """
```

- 函数嵌套调用：一个函数内部嵌套调用另外一个函数