**编程思想（面向过程基础）===》函数**

以前的代码 缺点：

所有的代码的糅在一起（代码分工职责并不明确）

无法控制代码执行时机（用户 点击操作===》代码才执行）

并没有代码思路

解决以上问题：面向过程（语法实现基础===》函数）

**什么是函数？**

**就是一个具有特定功能、固定的职责的代码块。**

分为两种：

1. 内建函数，python提供，如print，input，int，len等。直接使用。

----有哪些内置函数直接：**print(dir(\_\_builtins\_\_))打印的函数都是内置函数，可以直接用的**

1. 用户自定义函数，使用def语句定义。必须先定义，再使用。

**函数格式：**

**1.函数定义格式：define---的缩写；**

def语句格式：

def 函数名(形参1，形参2，…，形参n):

函数代码块

return 返回值

说明:

1. 函数名遵守标识名的命名规则；
2. 函数名后有小括号()，之后有冒号:
3. 参数可以多个，也可以没有；
4. return可以有，可以没有。 ----如果没有就返回是None。

**2.函数调用格式：**

函数名(实参1，实参2，…，实参n)

通过设计实现一个加法函数，并实际使用。

**def** add(a,b):####这是形式参数  
 sun=a+b  
 **return** sun  
  
s1=add(4,3)#########这是实际参数。  
print(**"结果："**,s1)  
s2=add(45,23)  
print(**"结果："**,s2)

每日一练：脱稿4分钟。

1.黑盒测试用例设计方法，分别举例---在软件的功能中体现。

2.自动化测试和手工测试有什么优缺点。

函数作用：复用代码，减少代码量

形参和实参关系和区别：

1. 共同点====》都表示函数的参数 一般呈**一一对应关系**
2. 函数的**形参** 只会出现在**函数定义**的位置
3. 函数的**实参** 只会出现在**函数调用**的位置

return:

* + 1. 如果没有return，意味着函数没有返回值===就是None
    2. 函数有return，可以把return的返回值，赋值给到一个变量
    3. 方法内部的变量或执行结果，必须return，才能在方法外部使用

**作业**

* **编写函数，判断一个整数是否为素数，并编写主程序调用该函数。**

**def** primeNum(a):  
 **if** a>1:  
 **for** i **in** range(2,a-1):  
 **if** a%i==0:  
 print(**"对不起！你不是素数哦！"**)  
 **break  
 else**:  
 print(**"很开心，你是一个素数哦！"**)  
 **else**:  
 print(**"请输入大于1的整数！"**)  
  
  
aa=int(input(**"请输入"**))  
primeNum(aa)

**思考：文件1中定义的函数a如何在文件2中使用？？**

**函数还可以给不同文件调用（提高代码复用性）====》包与模块**

包和模块的制作和调用====》**目的将函数 可以给不同的文件去使用**

1.包 package ====》文件夹，包含多个python文件

2.模块 ========》 文件

一个包可以包含多个模块

一个文件包含多个函数

实例：**文件1中定义的函数add如何在文件2中使用**

1.包pacg1，包含一个文件(模块)py1，包含一个函数：

**def** add(a,b):  
 sun=a+b  
 **return** sun  
  
s1=add(4,3)  
print(**"结果："**,s1)

2.包pacg2，包含一个文件(模块)py2，在py2调用py1的函数add.

**from** pacg1.py1 **import** add  
  
s=add(30,40)  
print(**"加法结果"**,s)

注意：**导入包和模块时，可以使用快捷键导入：首先输入方法名，然后按快捷键alt+enter，然后选择需要导入的内容。**

需求：文件(模块)1 的函数被文件（模块）2调用，不希望执行文件1中测试代码

解决：在 文件1的测试代码上 增加 if \_\_name\_\_==”\_\_main\_\_”:

作用：如果在执行自己文件时候（会执行该if结构下的代码）

如果文件（文件1）被别人（文件2）使用，运行文件2时，文件1中 if \_\_name\_\_==”\_\_main\_\_”: 此结构下的代码不会被执行。

改造后py1文件：

**def** add(a,b):  
 sun=a+b  
 **return** sun  
  
**if** \_\_name\_\_==**"\_\_main\_\_"**:  
 s1=add(4,3)  
 print(**"结果："**,s1)

函数调用总结：

1. 函数定义的文件内调用 ====》直接调用

函数引用格式：add(30,40)

1. 函数定义的文件外调用 ====》需要先导入
   1. 直接导入函数

格式：**from** 包.文件 **import** 函数名

**from** pacg1.py1 **import** add

函数引用格式：add(30,40)

* 1. 导入函数所在文件

格式：**from**包 **import**文件

**from** pacg1 **import** py1

函数引用格式：py1.add(30,40) ===》文件名.方法名()

1. if \_\_name\_\_==”\_\_main\_\_” 该结构下代码， 如果执行该文件自己，该文件下的if中的代码就会被执行，如果该文件被别的文件引用执行，该文件结构下的代码就不会被执行。

关于if \_\_name\_\_==”\_\_main\_\_”；如果是主函数执行\_\_name的值为\_\_main\_\_

如果是子函数：\_\_name\_\_为子函数的文件名；

主函数是指：Python需要运行的文件。因为Python一次只能运行一个文件。

因为Python需要导包，其它的都不是主文件，都是子文件。。。

在Python中：只需要运行的Python文件需要加\_\_name\_\_==”\_\_main\_\_”,其它的子文件都不用;

按照标准写法：

**if \_\_name\_\_ ==”\_\_main\_\_”：上面的代码：要么就是函数，要么就是一个类**;