函数

（一） 定义:

可重复调用的，实现特定功能的代码段

（二）使用原因:

重复的代码在重复的写入时会使代码变的冗长，且可读性降低。不如将其写成函数来重复调用，不仅减少代码量，也可以增加程序的可读性。

（三）语法:

```
def 函数名 (形式参数,默认参数 = 默认值,*可变参数,**关键字参数):
    """
        函数说明
    """
    函数体
    return 返回值
```

其中:
（1）:
传入参数：函数外部将数据通过参数将数据传入函数中来进行函数的使用或者计算

参数可以传入任意个，但要根据函数的要求进行隔开。

参数可以不设置，没有参数也是可以的
eg:

```
def p():
    print("没有参数")
p()
```

形式参数：接受任意类型的数据，并传入给函数。

默认参数：当该参数没有值传入时,参数为默认值

可变参数：接受任意类型的数据，但是会将这些数据变为一个元组再传入函数中

关键字参数：接受任意类型的数据，但是会将这些数据变为一个字典再传入参数

(2):
函数说明：解释说明函数的用法，用途，函中的各个参数。在函数中以"""开头，以"""结尾
eg:

```
def f(n):
 """
     这是一个计算菲波纳茨数列的第n个的值
     : n 菲波纳茨数列的位数
 """
     if (n <= 2 and n > 0):
         return 1
     else:
         return f(n - 1) + f(n - 2)
```

(3):
函数体：函数的主体

(4):
return 返回值

当没有指定返回值时，函数默认返回 None

当指定返回值时，则返回指定的值

当函数执行完 return语句 时，则会直接结束函数本身

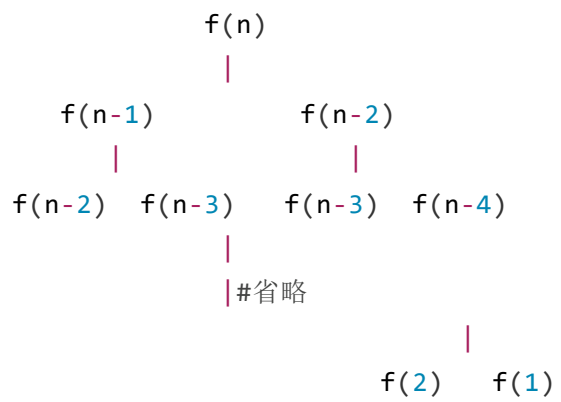
(5)
None 值作为特殊的字面量，在bool类型中为False，也可以定义变量为None
eg:

```
a = None
```

（四）嵌套:
函数嵌套:所谓函数的嵌套是一种流程，在一个函数中调用另一个函数(也包括自身)，例如函数(A)调用函数(B)，则会先执行完函数(B),在执行函数(A)

```
def f(n):
    if (n <= 2 and n > 0):
        return 1
    else:
        return f(n - 1) + f(n - 2) #调用了自身本身
```

函数递归解释图：



自身不断调用自身，直到最后的值确定下来，即f(2) = 1 , f(1) = 1.当函数最终有确定值时，将确定的值返回给上一层，直到返回第f(n)，求出f(n)的值。

（五）变量的作用域：

(1):

局部变量：只在循环体。函数体中存在的变量，当函数或者循环执行完成时，该变量就会结束。

不可被循环体或者函数外进行调用，调用会报错：未被定义

那么该如何调用呢?

(2)

全局变量：在整个程序中都可以被调用（包括循环和函数）,函数时内声明时需要再前面加上global。

```
global a
a = 5#定义一个名为a的全局变量，且a的值为5
```

在函数或者循环中通过加减乘除等表达式对全局变量进行修改时,需要在函数中先声明该变量为全局变量，不然程序会不知道该变量是临时变量还是全局变量而导致变量，从而导致报错。

eg:

```
a = 5
def f():
    global a
    a = a + 5
    print(a)
f()# a 为 10
```