

Python 函数

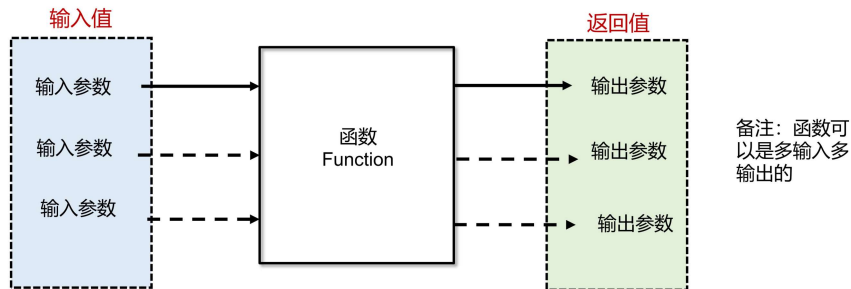
什么是Python函数



- 如何使用一个函数

变量=函数名 (函数参数1, 函数参数2)

形如`a=add(1,2)` #调用一个加法函数, 输入1,2, 返回结果3



函数语法

In [1]:

```
def functionname( parameters ):
    "函数_文档字符串"
    function_suite
    return [expression]
```

一个简单的例子：获取字符串长度

在Python中我们可能会多次用到获取字符串的长度。对于给定的某个字符串“qiwehqwhoeq”，我们可以通过如下代码：

In [3]:

```
n=0 #计数器
for i in "qiwehqwhoeq":
    n=n+1

print("该字符串的长度为：",n)
```

该字符串的长度为： 11

但实际上这样的执行方式使得代码冗余，降低了可读性。因此Python 提供了一个功能，即允许我们将常用的代码以固定的格式包装成一个独立的模块，只要知道这个模块的名字就可以重复使用它，这个模块就叫做**函数 (Function)**

完成一个获取字符串长度的函数

In [9]:

```
def getStrLength(string):  
    """  
    一句话的简介：获取字符串的长度  
    输入参数：string 任意的字符串  
    输出参数： n 字符串的长度  
    """  
  
    n=0 #计数器  
    for i in string:  
        n=n+1 #计数器+1  
    return n #返回计数器的值  
  
#####  
string="uhqeqwhueq3wtftffwe2163612q"  
stringl="uhqeqwhueq3wtdadsdasdftffwe2163612q"  
#函数的调用  
num1=getStrLength(string)  
num2=getStrLength(stringl)  
#看一下n的值  
print(num1)  
print(num2)
```

28
36

形参与实参

形参：在函数的定义部分中参与运算的参数

实参：在函数被定义调用过程中参与运算的参数

"实参和形参的区别，就如同剧本选主角，剧本中的角色相当于形参，而演角色的演员就相当于实参。"

小练习：定义一个函数，输入参数为a，返回 $a^2 + 1$ 的值

In [13]:

```
def function(a):  
    return a**2+1  
  
#调用函数  
result=function(89)  
  
print(result)
```

7922

多输入参数

当一个函数有多个输入参数时，如何调用该函数呢？

In [17]:

```
def function(a, b):  
    return a/b  
  
#我想计算5/3的值  
result=function(3,5)  
  
print(result)
```

0.6

1. 按位置传参

In []:

```
def function(a, b):  
    return a/b  
  
#我想计算5/3的值  
result=function(5,3)  
  
print(result)
```

2. 按关键词传参

In [18]:

```
def function(by_div, div):  
    return by_div/div  
  
#我想计算5/3的值  
result=function(by_div=5, div=3)  
  
print(result)
```

1.6666666666666667

练习1: 定义一个函数 `max_num_in_list(lst)`, 输入一个列表 `lst`, 返回这个列表中的最大值。提示: 可以用 `for` 循环遍历列表, 用 `if` 判断找到最大值。例如:

输入:[1,3,5,7,15,9,11]

返回: 15

In [25]:

```
def max_num_in_list(lst):
    maxNum=lst[0] #手上拿的这个数
    for i in lst: #遍历列表的每一个元素
        if i>maxNum:
            maxNum=i
    return maxNum

#测试
L=[1, 3, 5, 7, 15, 99, 9, 11]

print(max_num_in_list(L))
```

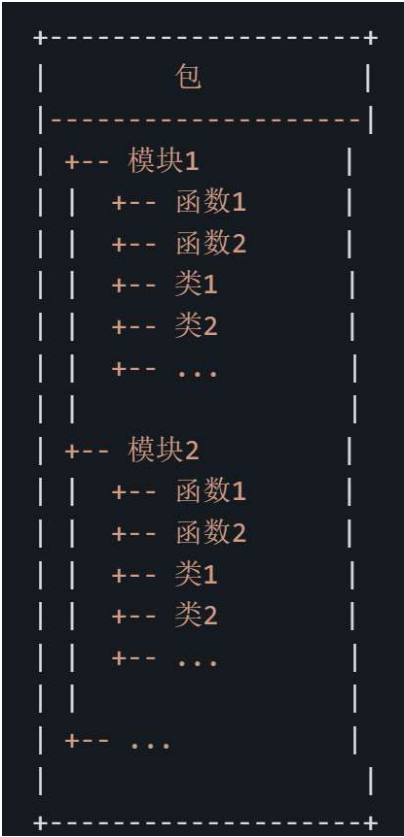
99

Type *Markdown* and LaTeX: α^2

模块(Module)和包 (package)

在Python中，模块指的是包含了一组函数、变量和类等定义的文件。模块的作用是将相关的功能组织在一起，使代码更加模块化，从而更容易维护和修改。模块可以被其他程序引入，以便在这些程序中使用模块中定义的功能。

而包是一种组织Python模块的方式，它将相关的模块组织在一起，形成一个层次结构。



重复造轮子



```

'''
import xxx    #这里的xxx指代模块名

xxx.someFunction()    #这里的someFunction() 指xxx模块内部的函数

xxx.someFunction(parameter1,parameter2,...)    #这里的parameter1,parameter2,... 指的是该函数需要传入的参数

'''

```

```
'\nimport xxxx    #这里的xxx指代模块名\n\nxxxx.someFunction()    #这里的someFunction()指xxx模块内部的函数\n\nxxxx.someFunction(parameter1,parameter2,...)    #这里的parameter1,parameter2,...指的是该函数需要传入的参数\n\n'
```

Time时间模块

获取系统时间（时间戳）

In [12]:

```
import time # 引入time模块

ticks = time.time()
print ("当前时间戳为:", ticks)
```

当前时间戳为: 1680681388.8965292

获取系统时间（年月日时分秒）

In [27]:

```
import time

localtime = time.localtime(time.time())
print ("本地时间为 :", localtime)
```

本地时间为 : time.struct_time(tm_year=2023, tm_mon=4, tm_mday=5, tm_hour=20, tm_min=4, tm_sec=52, tm_wday=2, tm_yday=95, tm_isdst=0)

判断程序执行时间

In [2]:

```
import time

start=time.time()#获取初始的时间

#需要测试的程序
for i in range(1000):
    for j in range(120):
        a=1

end=time.time() #获取结束的时间

print("程序执行的时间为: %s秒"%(end-start))
```

程序执行的时间为: 0.005000114440917969秒

延时函数

In [5]:

```
print(time.time())
print("hello")
time.sleep(3) #让程序暂停执行3s
print("world!")
print(time.time())
```

```
1680696781.6639135
hello
world!
1680696784.6706145
```

定时任务

假设我们需要在8:21的时候发出一条hello,world!

In [12]:

```
import time

#需要知道当前时间
currentTime=time.localtime()
current_Hour=currentTime.tm_hour
current_Min=currentTime.tm_min
current_Sec
#需要知道目标时间
goalTime_Hour=20
goalTime_Min=25
goalTime_Sec
#需要知道我们延长的时间s

delayTime= (goalTime_Hour-current_Hour)*3600+(goalTime_Min-current_Min)*60

#延长时间
time.sleep(delayTime)

#定时任务
print("hello, world")
```

```
hello, world
```

一种更高效的定时任务

while(当前不等于设置的时间的时候): 等待

In [8]:

```
currentTime=time.localtime()

currentTime.tm_hour
```

Out[8]:

```
20
```

In []: