函数

1. 打印一个99乘法表

I = 1

While I <= 9:

J = 1

While j <=I :

Print(“%d\*%d=%d\t” % (j,I,i\*j), end=””)

J+=1

I+=1

Print(“”)

1. **函数介绍**

2.1、定义函数的格式如下：

**def 函数名():**

**代码**

2.2、调用函数

**定义了函数之后，就相当于有了一个具有某些功能的代码，通过调用来执行。**

1. **有传参有返回值的函数**

**3.1、需求：**

**用户输入2个数字让其相加，结果返回，接受后打印。**

**def add\_num(a,b):**

**add = a + b**

**return add**

**#调用函数**

**result = add\_num(100,100)**

**print(‘result==%d’ % result)**

1. **函数的类型**

**4.1、无参数，无返回值**

**用于打印提示等类似功能**

**4.2、无参数，有返回值**

**采集数据使用此类函数**

**4.3、有参数，无返回值**

**对某些变量设置数据而不需要结果**

**4.4、有参数，有返回值**

**需要数据处理并需要结果的应用**

1. **函数的参数**

**5.1、缺省参数**

**调用函数时，缺省参数的值如果没有传入，则被认为是默认值。**

**注意：带有默认值得参数位于参数列表最后**

**5.2、不定长参数**

**需要一个函数能处理比当初声明时更多的参数，这些参数叫做不定长参数。声明时不会命名**

**5.3、不定长参数\*\*kwargs**

**加了\*的变量args会存放所有未命名的变量参数，args为元组。加了\*\*的变量kwargs会存放命名参数，形如key=value的参数，kwargs为字典。**

**5.4、引用传参**

**可变类型与不可变类型的变量分别作为函数时，对于不可变类型，运算不会影响变量自身。而可变类型函数体中的运算有可能会更改传入的参数变量。**

**Python中函数参数是引用传递（不是值传递），不可变类型，因变量不能修改。**

**5.4.1、a+=a方式：**

**使用原来的地址**

**5.4.2、a=a + a方式：**

**开辟新的空间，重新引用**

**5.5、函数返回值**

**5.5.1:使用return返回函数的值**

**5.5.2:保存函数的返回值**

**5.5.3:函数返回多个值**

**5.5.4:函数的嵌套调用**

**课堂练习1：写一个函数求3个数的和**

**课堂练习2：写一个函数求3个数的平均值**

**def sum3Number(a,b,c):**

**return a+b+c**

**def avg3Number(a,b,c):**

**sumResult = sum3Number(a,b,c)**

**avgResult = sumResult/3.0**

**return avgResult**

**result = avg3Number(4,5,6)**

**print(“avg us %d”%result)**

**5.6、全局变量**

**5.6.1:全局变量的命名**

**前面加上个g**

**g\_a = 32 #全局变量**

**5.6.2:global**

**如果需要修改全局变量，需要在函数值前面添加global修饰。**

1. **面向对象**

**6.1、面向对象和面向过程的区别**

**面向过程：用一长段代码来实现指定功能，开发过程的思路是将数据与函数按照执行的逻辑顺序组织在一起，数据与函数分开考虑。**

**面向对象：将数据与函数绑定到一起，进行封装。这样能够更快速开发程序，减少重复代码的重写过程 。**

**6.2、类的构成**

**6.2.1、类的名称:**

**类的属性：**

**类的方法:**

**例1：**

**#定义一个类**

**class Cat:**

**def eat(self):**

**print(‘猫在吃鱼’)**

**def drink(self):**

**print(‘猫在喝水’)**

**def print\_info(self):**

**print(‘%s的年龄是:%d’%(self.name,self.age))**

**#创建对象**

**tom = Cat()**

**#调用方法**

**tom.eat()**

**tom.drink()**

**6.2、类的\_\_init\_\_()方法**

**1.当创建实例对象成功后，有python解释器来调用该方法，无需手动调用。**

**class Cat:**

**def\_ \_init\_\_(self):**

**self.name=’tom’**

**self.age =30**

**print(‘\_init\_调用了)**

**def eat(self):**

**print(‘在吃鱼’)**

**tom = Cat()**

**print(‘猫的名字为:%s’%tom.name)**

**print(‘猫的名字为:%d’%tom.age)**

**2. \_\_init\_\_()方法总结**

**1)\_\_init\_\_()方法：不用手动调用，创建对象时默认被调用。**

**2)\_\_init\_\_(self):如果创建对象时传递2个实参，那么\_\_init\_\_(self)中除了self作为第一个形参外还需要2个形参数，**

**如：\_\_init\_\_(self,x,y)**

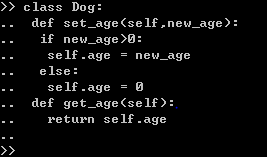
**3)\_\_init\_\_(self)中的self参数，不需要开发者传递，python解释器会自动把当前的对象引用传递进去。**

**6.3、\_\_str\_\_()方法**

**当使用print输出对象的时候，只要自己定义了\_\_str\_\_(self)方法，那么就会打印从这个方法中return的数据。**

**该方法，在使用print打印对象的时候，会调用该方法，通常输出当前对象的相关数据。**

**6.4、set和get方法隐藏数据**



**6.5、面向对象的3大特征**

**6.5.1、封装的定义：**

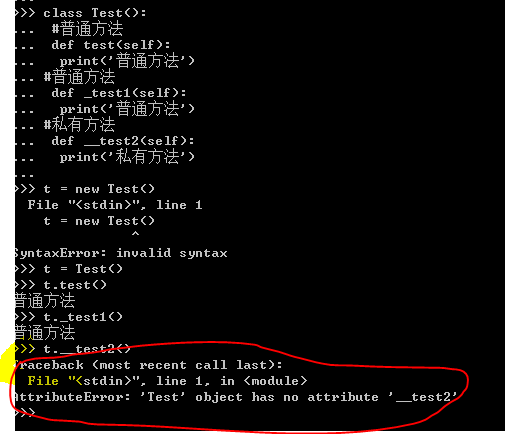
**面向对象的程序设计中，某个类把所需要的数据（类的属性），和对数据的操作（类的行为）全部封装在类中。分别称为类的成员变量和方法，这种把成员变量和成员函数封装在一起的编程特性称为封装。**

**6.5.2、公有成员变量与私有成员变量：**

**私有成员变量：‘\_\_’开头，私有成员变量只能在类的内部访问，公有成员变量可以在类的外部访问。**

**6.5.3、私有方法：**

**在python中没有像java语言的public private等修饰符，在python中定义私有变量只需要在变量名或函数名前面加上’\_\_’两个下划线，那么这个函数或变量就是私有。**



**Class Test:**

**def \_\_send\_msg(self):**

**print(‘---发送短信—‘)**

**def send\_msg(self,new\_money):**

**if(new\_money >1000):**

**self.\_\_send\_msg()**

**else:**

**print(‘余额不足’)**

**6.6、\_\_del\_\_()方法：**

**创建对象后，python解释器默认调用\_\_init\_\_（）方法，当删除一个对象时，python解释器也会默认调用一个方法，这个方法就是\_\_del\_\_()方法。**

**不管是手动还是调用还是python自动回收都会触发\_\_del\_\_方法执行。**

**小结：当有1个变量保存了对象的引用时，此对象的引用计数会+1**

**1）当使用del删除变量指向的对象时，如果对象的引用计数不是1，比如3，那么此时只会让这个引用计数-1，变为2.**

**2）当再次调用del时，变为1，如果再调用1次del,此时会真的把对象进行删除。**

**3）del 不会主动调用\_\_del\_\_方法，只有引用计数==0时，\_\_del\_\_()才会被执行，并且定义了\_\_del\_\_()的实例无法被python的循环垃圾收集器收集。所以尽量不要自定义\_\_del\_\_().**

