一，什么是函数：函数是组织好的，可以重复使用，用来实现单一或相关功能的代码段，能提高应用的模块化和代码的重复利用率。

二，函数定义和调用：

1，定义函数：def 函数名（参数列表）：

函数体；

return 表达式

说明：①函数代码块以 def 开头，后面紧跟着的是函数名和圆括号

②函数名只能是字母、数字、下划线的任意组合，但是不能以数字开头，并且不能跟关键

字重名

③函数的参数必须放在圆括号中

④函数的内容以冒号起始，并且缩进

⑤return表示函数结束，选择性地返回一个值给调用方，不带表达式的return相当于返回

None

2，调用函数：通过函数名（）即可完成函数的调用；函数有参数时，调用函数的时候需要明确给定

参数（变量只有确切给定值以后才知道它是什么类型的参数）

三，函数的参数：

1，函数参数的传递：值传递（如果函数定义了多个参数，name在调用函数的时候，传递的数据要

和定义的参数一一对应）

2，默认参数：定义函数的时候，可以给函数的参数设置默认值，这个参数称为默认参数，当调用函

数的时候，由于默认参数在定义时已经被赋值，所有可以被忽略，而其他参数是必须要传入值

的。如果默认参数没有传入值，则直接使用默认的值，如果默认参数传入了值，则使用传入的

值。

注意：带有默认值的参数一定要位于参数列表的最后面，否则程序会报错

3，不定长参数：通常在定义一个函数时，若希望函数能够处理的参数个数比当初定义的参数个数要

多，此时可以在函数中使用不定长参数

```
def 函数名（[formal_args，]*args，**kwargs）：
    函数体
    return 表达式
```

说明：①上述格式中，函数共有3个参数，其中formal_args为形参，*args和**kwargs为不定长参数，

当函数调用的时候，函数传入的参数个数会有限匹配formal_args参数的个数，如果传入的

参数个数和formal_args的参数个数相同，不定长参数会返回空的元组或者字典，如果传入

的参数个数大于formal_args的参数个数，可以分为俩种情况：首先，如果传入的参数没有

指定名称，那么*args会以元组的形式存放这些多余的参数；其次，如果传入的参数指定了

名称，如 m =1，那么**kwargs会以字典的形式存放这些命名的参数，如{m：1}

②调用函数的时候，如果传统的参数匹配够了，多余的参数会组成一个元组，当调用函数的时

候，有”键=值“这样的形式出现，则这些键值对会保存在字典中

4，函数的返回值：在python中，函数的返回值是使用return语句来完成的

5，函数的四种类型：

①无参无返回值 ，既不能接收参数，又不能传输数据

②无参有返回值，不能接收参数，但是可以返回一值

③有参无返回值，能接收参数，但是不能返回数据

④有参有返回值，既能接收数据，又能返回数据

注：print("*"*30)连续打印三十个*

6，函数的嵌套调用：在一个函数中调用了另外一个函数，这就是所谓的函数的嵌套调用

7，变量的作用域：

1）LEGB原则：（python的作用域）

①L（local）：函数内的区域，包含局部变量和参数

②E（enclosing）：外面嵌套函数区域，常见的是闭包函数的外层函数

③G（global）：全局作用域

④B（build-in）：内建作用域

python中的变量是采用L---->E----->G----->B的规则查找，

2）全局变量和局部变量：所谓局部变量指的是定义在函数内的变量，局部变量只能在其被声明

的函数内部进行访问，而全局变量可以在整个程序范围内访问，全局变量是定义在函数外的

变量，它拥有全局的作用域

3）global和nonlocal关键字

①global关键字用来在函数或者其他局部作用域中使用全局变量但是如果不修改全局变量也

可以不使用global关键字，在python中，如果函数内部对全局变量进行修改，python会把

变量当做是局部变量，为了使得全局变量生效，我们可以在函数内使用global关键字进行

声明

②nonlocal关键字是在python3.0中新增的关键字，python2.x不支持，使用nonlocal关键字可以在一个嵌套的函数中修改嵌套作用域中的变量

注意：使用global关键字修饰的变量可以不存在，而使用nonlocal关键字修饰的变量必须在嵌套作用域中已经存

8，递归函数和匿名函数：

1）递归函数：一个函数在内部调用自身那么这就是一个递归函数

2）匿名函数：匿名函数就是没有名称的函数，也就是不再使用def语句定义的函数，如果需要声明

一个匿名函数，则需要使用lambda关键字，匿名函数的声明格式：

函数名 = lambda [arg1[,arg2,arg3.....argn]]:表达式

匿名函数的调用：函数名（参数列表）

注意：1，在调用函数前，必须要先声明函数，不然会报错

2，使用lambda声明的匿名函数能接收任何数量的参数，但只能返回一个表达式

的值，匿名函数不能直接调用print（）函数，这是因为lambda需要一个表达式

注意：def函数和lambda函数的不同点：

①def函数是有名称的，而lambda函数没有函数名称

②lambda函数通常会返回一个对象或者一个表达式，它不会将返回的结果赋值给一个变量，而

def函数则可以

③lambda函数中只有一个表达式，函数体比def函数简单得多，def函数的函数体是一个语句

④lambda表达式的冒号后面只有一个表达式，def函数可以有多个

⑤像if或for语句不能使用在lambda函数中，def函数中可以使用

⑥lambda一般用来定义简单的函数，def用来定义复杂的函数

⑦lambda函数不能被除了本程序之外的其他程序调用，但是def函数可以被其他程序调用

9，日期时间函数：python提供了time和calender模块用于格式化日期和时间，

1）时间函数：

①时间戳：表示从1970年1月1日00:00:00开始按*秒*计算的偏移量，返回时间戳的函数有

time（），clock（）等

引入时间模块：import time

time1 = time.time()

常用快捷键：

https://blog.csdn.net/t8116189520/article/details/80530414

②格式化的时间字符串：使用time模块的strftime函数来格式化日期

使用格式：time.strftime(format[,t])

格式化符号：%y表示俩位数的年份，%Y表示四位数的年份，%m表示月份，%d表示月内

中的一天，%H表示二十四小时制的小时数，%I表示十二小时制的小时

数，%M表示分钟数，%S表示秒数，%a表示本地化时间的简称，%A表示本

地完整的星期简称，%d表示本地简化的月份简称，%B表示完整的月份简

称，%c表示本地相应日期表示和时间表示，%j表示年内的一天，%p本地

的A.M或P.M等价符，%U一年内星期数，星期一为星期的开始，%w表示星

期，星期天为开始，%x表示本地相应的时间表示，%X本地相应的时间，

%Z当前时区的名称，%%表示%本身

③时间元组：（struct_time）返回struct_time的函数有gmtime（）、localtime（）、

strptime（），struct_time元组共有9个元素，

0--->tm_year表示四位数的年份

1--->tm_mon表示月份

2--->tm_mday表示天数

3--->tm_hour表示小时数

4--->tm_min表示分钟数

5--->tm_sec表示秒数

6--->tm_wday表示星期数

7--->tm_yday表示一年中的第几天

8--->tm_isdst表示决定是否为夏令时的标识，返回值为-1,0,1

另外，time模块还提供了很多其他的函数，sleep（）函数用于推迟调用线程的运行

衡量不同程序的耗时状况，time.clock（）比time.time更有用

time模块的俩个非常重要的属性：①time.timezone是当地时区（未启动夏令时）距离格林威治的偏移秒数（>0 ，美洲；<=0大部分欧洲，亚洲和非洲）②time.tzname属性包含一对根据情况的不同而不同的字符串，分别是带夏令时的本地时区名称和不带名称的

2）日历函数（calender）：

1）），calender模块有很广泛的方法来处理年历和月历

引入日历模块：import calendar

打印1月份的月历：

```
c1 = calendar.month(2018,1)
print(c1)
```

2）），calender模块的内置函数

①calender.calender（year，w=2，l =1，c=6）返回一个多行字符串格式的

year年年份，三个月一行，间隔距离为c，每日宽度间隔为w字符，每行长度

为21*w+18+2*c，l为每星期行数

如，打印一整年的年历：

```
c2 = calendar.calendar(2017,w = 3,l=1,c=6)
print(c2)
```

②calender.firstweekday()返回当前每周起始日期的设置，默认情况下，首次载入calender模块是返回0，即周一

```
如：c3 = calendar.firstweekday()
    print(c3)
```

③calender.isleap(year)year年是闰年返回true，否则返回false

④calender.leapdays(year1,year2)返回在year1和year2之间的闰年总数

⑤calender.month（year，month，w = 2，l = 1）返回一个多行字符串的

year年month月日历，俩行标题，一周一行，每日宽度间隔为w字符，每行

的长度为7*w+6，l是每星期的行数

⑥calender.monthcalender（year，month）返回一个整数的单层嵌套列表，

每个子列表装载代表一个星期的整数，year年month月外的日期都设为0，

范围内的日子都由该月的第几日表示。

⑦calender.monthrange（year，month）返回俩个整数，第一个是该月的星期

几的日期码，第二个是该月的日期码，日从0--6，月份从1--12

⑧calender.prcal（year，w = 2，l = 1，c = 6）相当于

```
print（calender.calendar(year,w,l,c)）
```

⑨calender.prmonth（year，month，w = 2，l = 1）

```
print（calender.calendar(year,w,l,c)）
```

⑩calender.setfirstweekday（weekday）设置每周的起始日期码，0（星期

一）----6（星期日）

①calender.timegm（tupletime）和time.gmtime相反

②calender.weekday（year，month，day）返回给定日期的日期码

10），随机数函数(引入random模块)

①random.random()返回0<=x<1之间的随机浮点数

②random.uniform（a,b）返回a到b之间的浮点数，范围[a,b]，如果a<b,生成的随机

数的范围是a<= x <=b,如果a>b,生成的随机数的范围是b<= x <=a

③random.randint（a，b）返回一个a<= x <=b的随机整数，a<b且a和b都只能是整数

④random.randrange（[start],stop[,step])返回递增基数集合中的一个随机数,基

数的默认值为1，其中，start参数用于指定范围内的开始值，包含其在内，end参

数用于指定范围内的结束值，其不包含在内，step表示地政基数

⑤random.choice（sequence）表示从sequence中返回一个随机的元素，其中，

sequence参数可以是列表、元组、字符串

⑥random.shuffle（x[，random])用于将列表中的元素打乱顺序，俗称洗牌

⑦random.sample（sequence，k）从指定序列中随机获取k个元素作为一个片段返

回，sample函数不会修改原有的序列