*# 第一个注释*

*'''*

*第二个注释*

*'''*

*print(1)*

*# python使用首行缩进来表示代码块，但是对于同一个代码块的语法必须包含相同 的缩进空格*

*#if True:*

*# print("True");*

*#else:*

*# print("False");#否则的话会导致错误*

*#如果语句比较的长的话，可以使用反斜杠来实现多行语句,但是如果是[],{},()之间的语句，则不需要反斜杠*

*total=['item\_one','item\_two','item\_three',*

*'item\_four','item\_five']*

*#四种数据类型，int,bool,float,complex*

*print(r"this is alline with \n");*

*#转义符\,用r可以使得反斜杠不发生转义*

*#字符串从左往右从0开始，不可改变，没有单独的类型*

*str="Jzysnd"*

*str2=str[0:4:2]#左开右闭*

*print(str2)*

*#分号运算符，语句之间用分号分割，可以在一行中使用多条语句，语句之间使用；分割*

*'''*

*print 默认换行，如果不需要换行的话为print(x,end="")*

*'''*

*# import somemodule,frome somemodule import somefunction,somefunction,from someodule import \**

*#python变量不需要声明，使用前都需要赋值*

*counter=100*

*miles=100.0*

*name="runoob"*

*print(counter)*

*print(miles)*

*print(name)*

*a=b=c=1*

*d,e,f=1,2,"runoob"*

*#Python数据类型：number,string,tuple(不可变)，list,dictionary,set(可变)*

*#number*

*#包括int,float,bool,complex*

*#0,1*

*#删除对象：del var1*

*var1=1*

*del var1*

*#print(var1)*

*#运算，+-\*/,%\*\*(乘方)*

*#bool:只有所有的空的东西都视为是false*

*#string*

*str='Runoob'*

*print(str)*

*print(str[0:-1])*

*print(str[0])*

*print(str[2:5])*

*print(str[2:])*

*print(str\*2)*

*print(str+"Test")*

*#但是和c语言字符串不同，python中的字符串不可以改变*

*#list列表*

*t=['a','b','c','d','e']*

*# 0 1 2 3 4*

*# -5 -4 -3 -2 -1*

*li=[1,2,3,4,5,6]*

*print(li)*

*li[0]=9*

*print(li)*

*li[1:4]=[9,9]*

*print(li)*

*#这些置换的操作，像直接对于其中的某一段的置换都是可以的*

*print(li[0:2:])*

*#中间的表示每隔2个数取一个*

*#tuple元组*

*#元组不可以改变元素，但是可以包含更多的东西*

*#set集合，可以存储唯一的元素，而这样的集合可以进行交并之类的操作*

*sites={1,2,3,4,5,6}*

*print(sites)*

*if 5 in sites:*

*print(111)#判断是否在里面的*

*#单元素特殊对待*

*#a&b交集，a|b并集，a-b差集,a^b a和b中不同时存在的元素*

*#dictionary 字典，无序的对象集合，但是字典中的元素是通过键来存取的*

*dict={}#定义字典，要与set的定义区分*

*dict['one']='jzy是废物'*

*dict['two']='clippers'*

*dict2={'three':'adbafb','four':11111,5:12341}*

*print(dict['one'])*

*print(dict['two'])*

*print(dict2.keys())*

*print(dict2.values())#对于字典是一种映射关系，其元素是键值对*

*#数据类型转换：隐式的为蒋较低的类型转换为较高的类型，否则的话使用强制的转换*

*#算数运算符*

*#+,-,\*,/,%,\*\*,//(下取整)*

*#比较运算符 ==，!=,>,<,>=,<=*

*#位运算符：&(仅有两个1为1)|(仅有两个0为0),^(相异为1) ~ << >>*

*#逻辑运算符： and or not*

*#成员运算符：in(如果在指定的序列中找到了值那么为True)*

*'''*

*常见数学函数*

*abs(x)*

*exp(x)为取x的e次幂*

*fabs(x)以浮点数形式返回*

*ceil(x)上取整*

*floor(x)下取整*

*log(a,b)返回logb a的值*

*log10(x)*

*max(x1,x2,.....),min(x1,x2,.....)*

*pow(x,y)x\*\*y的值*

*round(x,[.n])指的是保留带小数点后的第n位并进行四舍五入*

*sqrt(x)*

*'''*

*'''*

*常见随机数函数*

*choice(list)表示从list中的东西中随机挑选一个元素*

*random()在(0,1)之间随机生成一个实数*

*uniform(x,y)在x,y之间生成一个实数*

*pi,e为数学常量*

*'''*

*#字符串*

*#访问，拼接，转义，常见运算符(+,\*,[:],in)*

*print("jzy is %d pig" %(100));#格式化输出字符串*

*#list*

*list=[1,2,3,4,5,6,7]*

*del list[2]#表示删除第三个元素*

*#对于列表的+,\*操作符类似于字符串*

*'''*

*list.append(obj)表示在列表的结尾添加新的对象*

*list.index(obj)找到列表中某个值第一个匹配项的索引位置*

*list.insert(index,obj)在index的位置出插入obj*

*list.pop([index=-1])*

*'''*

*#对于元组tuple*

*#可以tup3=tup1+tup2,也可以整个删除*

*#对于字典dictionary*

*#建立dict(),定义，修改*

*dict={1:1,2:2,3:3,4:4}*

*del dict[1]*

*dict.clear()*

*del dict*

*#删除一个，清空，删除全部*

*#对于集合*

*#添加元素s.add(x)，删除元素s.remove(x)其中x式s中的元素，如果为随机删除则为s.pop()*

*#个数len(s),清空s.clear(),判断元素在集合中是否存在 x in s*

*'''*

*if 表达式：*

*elif 表达式：*

*else ：*

*match类似于switch但是比switch好一些*

*match status:*

*case 1:*

*print(1)*

*case 2:*

*print(2)*

*case 3:*

*print(3)*

*case \_:*

*print(alkdjfhalkdsh)这个东西类似于default*

*'''*

*'''*

*循环*

*while 条件表达式为真：*

*for 变量 in 列表：*

*然后每次变量会依次增加*

*比如for i in range(1,10):*

*range(a,b,c)表示从a到b每隔开c取一个数*

*range(n)表示从0~n-1依次循环*

*break,continue*

*'''*

*#推导式：即为一种简介的方法来构建一个新的列表*

*a=[1,2,3,4,5,6,7,8,9]*

*b=[i for i in a if i %2==0]#即为将a中的所有的满足一定的条件的东西拿出来*

*#python3迭代器与生成器，为一个可以记住遍历的位置的对象，从第一个元素开始访问*

*list=[1,2,3,4,5,6,7]*

*it=iter(list) #创建迭代器对象*

*print(it)#最一开始是在一个空的地方*

*print(next(it))*

*print(next(it))*

*for x in it:*

*print(x,end=" ")*

*print("")*

*#对于自行定义的一个类class,我们也可以用相同的方式搞出来迭代器*

*# 此时 \_\_iter\_\_(),\_\_next\_\_()就是这种东西*

*class MyNumbers:*

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

*self.a=1*

*return self*

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

*x=self.a*

*self.a+=1*

*return x*

*myclass=MyNumbers()*

*myiter=iter(myclass)*

*print(next(myiter))#对于类的知识到之后我们再弄*

*#迭代器与生成器*

*class Mynumbers:*

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

*self.a=1*

*return self*

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

*if self.a<=20:*

*x=self.a*

*self.a+=1*

*return x*

*else:*

*raise StopIteration*

*myclass=Mynumbers()*

*myiter=iter(myclass)*

*print(next(myiter))*

*#生成器：yield定义了生成器函数，对于迭代的过程中逐步产生值，只可以返回迭代器。*

*#将迭代器之后的表达式作为当前迭代的值返回*

*#函数*

*#格式 def 函数名(参数列表)：*

*#返回值return*

*#参数的传递，对于参数列表里面的东西只是名称，可以是指向不同的类型*

*def change(a):*

*a[0]=1000*

*return*

*a=[1,2,3,4,5]*

*change(a)*

*print(a)*

*#若传递的是不可变类型，则是新生成了一个a,对于可变类型，则是修改了一部分的值*

*#在传递的时候要保证传递的参数类型是一样的*

*'''*

*但是如果有多个参数比如change(age,name)*

*可以change(name="jzy",age=50)这样的关键字参数*

*'''*

*#若对于传递的时候参数个数未知，那么将使用元组和字典进行存储*

*def printinfo(arg1,\*\*vardict):*

*print("out:")*

*print(arg1)*

*print(vardict)*

*printinfo(1,a=2,b=3)*

*#在声明函数的时候，可以直接用\*来代替，但是之后的参数必须用关键字传入*

*#匿名函数lambda*

*f=lambda a,b,c,d:a+b+c+d*

*print(f(1,2,3,4))*

*#即为lambd 函数变量：表达式，即为用f()代替def的定义*

*#python数据结构*

*#stack我们可以将列表当作栈使用*

*stack=[]#创建栈*

*stack.append(1)#push操作*

*stack.append(2)*

*stack.append(3)*

*top\_element=stack[-1]#查看栈顶元素*

*is\_empty=len(stack)==0 #isempty*

*size=len(stack)*

*top\_element=stack.pop()*

*from collections import deque*

*import jzy*

*#使用python标准库实现队列*

*queue=deque()*

*queue.append('a')*

*queue.append('b')*

*queue.append('c')*

*print(queue)*

*first\_element=queue.popleft()#输出元素*

*print(first\_element)*

*front=queue[0]*

*print(front)*

*is\_empty=len(queue)==0*

*size=len(queue)*

*#但是我们也可以不使用这样的方法*

*'''*

*queue1=[]*

*queue1.append('a')*

*first\_e=queue1.pop(0)*

*front\_1=queue1[0]*

*'''*

*#链表也可以进行多重的嵌套和操作*

*#字典和序列的遍历方法*

*k={1:1,2:2}*

*for a in k.values():*

*print(a)#k.values,k.keys,k.items*

*jzy.hd(100)*

*#python 模块化设计*

*'''*

*import 文件名*

*使用的时候为*

*文件名.函数名(...)*

*而对于搜索路径为当前目录，环境变量指定的目录，python标准库目录，，pth中的目录*

*'''*

*#\_\_name\_\_变量表示当前的模块名称*

*'''*

*当作为主程序运行的时候\_\_name\_\_='\_\_main\_\_'*

*当作为模块插入的时候比如作为module.py插入的时候器值为当前的模块的名称*

*'''*

*# example.py*

*def greet():*

*print(1)*

*if \_\_name\_\_=="\_\_main\_\_":*

*print("直接运行")*

*greet()*

*else*

*print("模块")*

*#如果直接运行example的话，就是第一条*

*#example.greet()那么是第二条*

*#python文件输入输出*

*#repr(a)表示将变量a对应的各种各样的东西转换为字符串*

a=100

b=1000

s=**repr**((a,b,"11111"))

**print**(s)

*#而对于更加多样化的输出的话可以使用format*

**print**('{0}和{1}'.format('Google','Runoob'))

*#将{}内部的东西依次用format中对应的参数代换*

import math

**print**("{0:.3f}".format(math.pi))

*#同时也可以*

**print**("%.3f" %(math.pi))*#这样比较类似于c语言的写法*

*#str=input("...")输入一行文本*

'''

f=open(".../foo.txt","w")

f.write("jzy\njzy\n)

f.close()对于f.close,open都必须要有

str=f.read(size)读取一定数量的数据，如果size忽略或者为负的，那么该文件所有内容读取返回

str=f.readline()为读取单独的一行直到\n，如果返回一个空串，那么就是到了最后的一行

str=f.readlines()返回文件中包含的所有行

for line in f:

    print(line,end='')

f.write()写入一串内容，但是对于不是字符串的东西要提前进行转换

f.tell()会给出当前的读写位置，表示字节数偏移量

'''