#!/usr/bin/env python  
# python的字符串内置函数  
#1、capitalize() 返回将字符串的第一个字符转换为大写的字符串  
# str1 = '''hellow word这是一个测试模块  
# 主要作用域IT教育  
# '''  
#结果：Hellow word这是一个测试模块 主要作用域it教育  
# print(str1.capitalize())  
  
# 2、center(width, fillchar) 返回一个指定的宽度 width 居中的字符串，fillchar 为填充的字符，默认为空格  
# print('hello all'.center(50,'\*'))  
#center方法可以传递两个参数，一个必传参数指定字符的宽度，另一个选传参数，指定用什么字符来填补空白区域  
#结果：\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*hello all\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
#print('a','\n''b','\n''c','\n''d')  
#print('a','b','c','d',sep='\n')  
  
#3、count(str, beg= 0,end=len(string))返回 str 在 string 里面出现的次数，如果 beg 或者 end 指定则返回指定范围内 str 出现的次数  
# list = ' 我是你好，它是那就好，就是那好 '  
# num = list.count('是',0,10) #前10次有多少个是  
# print(num)  
# 2  
  
#4、bytes.decode(encoding="utf-8", errors="strict") Python3 中没有 decode 方法，但我们可以使用 bytes 对象的 decode() 方法来解码给定的 bytes 对象，这个 bytes 对象可以由 str.encode() 来编码返回。\*\*  
#5、encode(encoding='UTF-8',errors='strict') 以 encoding 指定的编码格式编码字符串，如果出错默认报一个ValueError 的异常，除非 errors 指定的是'ignore'或者'replace'  
# a = 'hello word nihao jjbond'  
# code = a.encode(encoding='utf-8')#编码 ， 比特流  
# code1 = a.encode(encoding='gbk')  
# print(code,code1)  
# print(code.decode()) #解码，括号里要一直  
# b'hello word nihao jjbond' b'hello word nihao jjbond'  
# hello word nihao jjbond  
  
#6、endswith(suffix, beg=0, end=len(string)) 检查字符串是否以 obj 结束，如果beg 或者 end 指定则检查指定的范围内是否以 obj 结束，如果是，返回 True,否则返回 False.  
#Tab键和空格不能混着用，会出现报错  
  
#7、expandtabs(tabsize=8) 把字符串 string 中的 tab 符号转为空格，tab 符号默认的空格数是 8 。  
# c = 'helloword nihao jjbond'  
# print(c.expandtabs(tabsize=20))  
  
#8、find(str, beg=0 end=len(string)) 检测 str 是否包含在字符串中，如果指定范围 beg 和 end ，则检查是否包含在指定范围内，如果包含返回开始的索引值，否则返回-1\*\*  
#rfind(str, beg=0,end=len(string)) 类似于 find()函数，不过是从右边开始查找.  
# str8 = '''hellow word这是一\t个测试模块  
# 主要作用\tIT教育模块 nihao  
# '''  
# print(str8.find('模块'))  
# 结果：找到第一个匹配数：18  
# print(str8.find('模块',str8.find('模块')+1,-1))  
#结果：找到第二匹配数：30  
  
#9、isalnum() 如果字符串至少有一个字符并且所有字符都是字母或数字则返 回 True,否则返回 False0  
# result = input('请输入您的ID： ') #包含数字或者字母返回ture  
# if result.isalnum():  
# name=result  
# else:  
# name='游戏ID非法'  
# print(name)  
  
#isalpha() 如果字符串至少有一个字符并且所有字符都是字母则返回 True, 否则返回 False  
# result = input('请输入您的ID： ') #包含字母返回ture  
# if result.isalpha():  
# name=result  
# else:  
# name='游戏ID非法'  
# print(name)  
  
#isdigit() 如果字符串只包含数字则返回 True 否则返回 False..  
# result = input('请输入您的ID： ') #包含字母返回ture  
# if result.isdigit():  
# name=result  
# else:  
# name='输入不是数字'  
# print(name)  
  
#islower() 如果字符串中包含至少一个区分大小写的字符，并且所有这些(区分大小写的)字符都是小写，则返回 True，否则返回 False  
# if 'Hello'.islower():  
# print(Ture)  
# else:  
# print('Hello'.lower())  
#结果：hello  
  
#isnumeric() 如果字符串中只包含数字字符，则返回 True，否则返回 False  
# print('123'.isnumeric())  
# #True  
# print('onetwothree'.isnumeric())  
# #False  
# print('壹贰叁'.isnumeric())  
# #True  
  
#\*\*isspace() 如果字符串中只包含空白，则返回 True，否则返回 False.\*\*  
# print(' '.isspace())  
# True  
  
#istitle() 如果字符串是标题化的(见 title())则返回 True，否则返回 False  
# c = 'hello ni hao'  
# print(c.title())  
# Hello Ni Hao  
  
#\*\*join(seq) 以指定字符串作为分隔符，将 seq 中所有的元素(的字符串表示)合并为一个新的字符串\*\*  
# c = 'hello ni hao'  
# print(' '.join(c))  
# h e l l o n i h a o  
  
#\*\*len(string) 返回字符串长度\*\*  
# c = 'hello ni hao'  
# print(len(c))  
# 12  
  
#ljust(width[, fillchar\]) 返回一个原字符串左对齐,并使用 fillchar 填充至长度 width 的新字符串，fillchar 默认为空格。  
# d = 'hello'  
# print(d.center(20,'\*'))  
# print(d.ljust(20,'\*'))  
# print(d.rjust(20,'\*'))  
# \*\*\*\*\*\*\*hello\*\*\*\*\*\*\*\*  
# hello\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*hello  
  
#maketrans() 创建字符映射的转换表，对于接受两个参数的最简单的调用方式，第一个参数是字符串，表示需要转换的字符，第二个参数也是字符串表示转换的目标  
# m = '1234'  
# n= 'abcd'  
# c=str.maketrans(m,n)  
# print(c)  
# #结果：{49: 97, 50: 98, 51: 99, 52: 100}  
#  
# #translate(table, deletechars="") 根据 str 给出的表(包含 256 个字符)转换 string 的字符, 要过滤掉的字符放到 deletechars 参数中  
# j = 'asd 1234 jkl ouio'  
# print(j.translate(c))  
# #结果：asd abcd jkl ouio  
  
#\*\*replace(old, new [, max\]) 把 将字符串中的 str1 替换成 str2,如果 max 指定，则替换不超过 max 次。\*\*  
# j = 'asd 1234 jkl ouio'  
# print(j.replace('asd','123'))  
# #结果：123 1234 jkl ouio  
# print(j.replace('asd','123').replace('jkl','jjbond'))  
# #123 1234 jjbond ouio  
  
#\*\*split(str="", num=string.count(str)) num=string.count(str)) 以 str 为分隔符截取字符串，如果 num 有指定值，则仅截取 num+1 个子字符串\*\*  
# num=input('请输入一个人数字： ')  
# print(num.split())  
# 结果：请输入一个人数字： 1 2 3 4  
# ['1', '2', '3', '4']  
  
  
#splitlines([keepends\]) 按照行('\r', '\r\n', \n')分隔，返回一个包含各行作为元素的列表，如果参数 keepends 为 False，不包含换行符，如果为 True，则保留换行符。  
# s='''  
# ajkdfhjkl  
# lsfdglugyhgi  
# dfiklwesd  
# '''  
# print(s.splitlines())  
# print(s.splitlines(keepends=False))  
# 结果：['', 'ajkdfhjkl', ' lsfdglugyhgi', ' dfiklwesd']  
# ['', 'ajkdfhjkl', ' lsfdglugyhgi', 'dfiklwesd']  
  
  
  
  
#list 列表 可迭代对象 可变的数据类型 字符串不可变 数值不可变  
#list = ['hello',123,[1,2,3]]  
#定义的名称，在定义，会发生名称的冲突  
list = []  
list1 = ['Google', 'baidu', 1997, 2000]  
list2 = [1, 2, 3, 4, 5 ]  
list3 = ["a", "b", "c", "d"]  
# for i in list3:  
# print(i) #会将列表的数值全部打印出来  
  
# a = 'hello'  
# del a  
# print(dir()) #会打印出当前环境下所有的变量  
#结果：['\_\_annotations\_\_', '\_\_builtins\_\_', '\_\_cached\_\_', '\_\_doc\_\_', '\_\_file\_\_', '\_\_loader\_\_', '\_\_name\_\_', '\_\_package\_\_', '\_\_spec\_\_', 'list', 'list1', 'list2', 'list3']  
# a = 'hello'  
# print(dir())  
# del a  
#['\_\_annotations\_\_', '\_\_builtins\_\_', '\_\_cached\_\_', '\_\_doc\_\_', '\_\_file\_\_', '\_\_loader\_\_', '\_\_name\_\_', '\_\_package\_\_', '\_\_spec\_\_', 'a', 'list', 'list1', 'list2', 'list3']  
  
# list3 = ['windows','mac','linux','unix','centos','deepin','ubuntu','debian']  
# list4 = ['鼠标','键盘','显示器']  
# print(list3 + list4) 列表加  
# print(list4 \* 2) 列表乘  
# print(list3[1:-1:2] + list4) 列表拼接  
  
#list5 = [32,64,12,3,128,55]  
# for i in list5:  
# if i > 50:  
# a.append(i) 大于50的列表  
# else:  
# b.append(i) 小于50的列表  
# print(a,'\n',b)  
#list5.append(10) 加入一个元数  
# list5.append([1,2,3]) 将整个列表加进去  
# print(list5.count(12)) 取当前列表有多少个这个数  
# list5.extend([1,2,3]) 可以将列表都加进去,并且加入其中  
# print(list5.index(128)) 查找列表的这个数,会输出索引  
# list5.insert(0,'hello world') 在最前面加上个数  
# print(list5.pop()) 在括号里输入一个索引,就会去除那个值,不输入,则会出去最后一个,并且打印出来  
  
# import random  
#list6 = ['小明','老王','小汤','小张'] 随机去除一个  
# num = random.randint(0,len(list5) -1)  
# delete = list5.pop(num)  
# print(delete)  
# print(list5)  
# list6.remove('老王')  
# print(list6)  
#list6.sort()  
#print(list6)  
  
# a = [1,2,3]  
# b = a.copy() # 列表拷贝 深浅拷贝  
# print(id(a),id(b))  
# a[0] = 'world'  
# print(a,b)  
# 结果:['world', 2, 3] [1, 2, 3]

#集合  
# student = {'Tom', 'Jim', 'Mary', 'Tom', 'Jack', 'Rose'}  
# student1 = {'jj','bb','h','Tom','Jim'}  
# print(student)  
# print(student - student1)  
# print(student | student1)  
# print(student & student1)  
# print(student ^ student1)  
# 结果：{'Mary', 'Tom', 'Jim', 'Jack', 'Rose'}  
# {'Jack', 'Mary', 'Rose'}  
# {'Mary', 'Tom', 'bb', 'h', 'Jim', 'jj', 'Jack', 'Rose'}  
# {'Jim', 'Tom'}  
# {'bb', 'h', 'Mary', 'jj', 'Jack', 'Rose'}  
  
# a = set('abracadabra')  
# b = set('alacazam')  
# print(a - b)  
# print(a | b)  
# print(a & b)  
# print(a ^ b)  
# 结果：{'d', 'r', 'b'}  
# {'a', 'c', 'r', 'm', 'b', 'z', 'd', 'l'}  
# {'a', 'c'}  
# {'z', 'r', 'm', 'd', 'l', 'b'}  
  
  
# thisset = set(("Google", "Baidu", "Taobao"))  
# thisset.add("Facebook")  
# print(thisset)  
# {'Facebook', 'Baidu', 'Taobao', 'Google'}  
  
  
# thisset = set(("Google", "Baidu", "Taobao"))  
# thisset.remove("Taobao")  
# print(thisset)  
# {'Google', 'Baidu'}  
#将元素 x 从集合 s 中移除，如果元素不存在，则会发生错误。  
  
  
# thisset = set(("Google", "Baidu", "Taobao"))  
# thisset.discard("Facebook") # 不存在不会发生错误  
# print(thisset)  
# {'Taobao', 'Baidu', 'Google'}  
# 此外还有一个方法也是移除集合中的元素，且如果元素不存在，不会发生错误  
  
  
# thisset = set(("Google", "Baidu", "Taobao", "Facebook"))  
# x = thisset.pop()  
# print(x)  
  
  
# 集合内置方法完整列表  
# a = set(("1", "2", "3"))  
# a1 = set(("3","4","1","5"))  
# a.add("4") #为集合添加元素 add()  
# print(a)  
# a.clear() #移除集合中的所有元素 clear()  
# print(a)  
# b = a.copy() #拷贝一个集合 copy（）  
# print(b)  
# c = a.difference(a1) #返回多个集合的差集 difference()  
# print(c)  
# a1.difference\_update(a) #移除集合中的元素，该元素在指定的集合也存在 difference\_update()  
# print(a1)  
# a.discard(3) #\*\*删除集合中指定的元素\*\* discard()  
# print(a)  
# d = a.intersection(a1) #返回集合的交集 intersection()  
# print(d)  
# d = a.isdisjoint(a1) #判断两个集合是否包含相同的元素，如果没有返回 True，否则返回 False isdisjoint()  
# print(d)  
# d = a.issubset(a1) #判断指定集合是否为该方法参数集合的子集。指定集合为参数 issubset()  
# print(d)  
# d = a.issuperset(a1) #判断该方法的参数集合是否为指定集合的子集 issuperset()  
# print(d)  
# d = a.pop() #随机移除元素，并且打印出来 pop()  
# print(d)  
# print(a)  
# a.remove("2") #移除指定元素 remove()  
# print(a)  
# d = a.symmetric\_difference(a1) #\*\*返回两个集合中不重复的元素集合。 symmetric\_difference()  
# print(d)  
# d = a.symmetric\_difference\_update(a1) #移除当前集合中在另外一个指定集合相同的元素，并将另外一个指定集合中不同的元素插入到当前集合中。（取两个集合中不同的元素） symmetric\_difference\_update()  
# print(a)  
# d = a.union(a1) #返回两个集合的并集 union()  
# print(d)  
# a.update('14') # 给集合添加元素 update()  
# print(a)