**Python3中有六种标准的数据类型**

**数字（Number）**

Python 数字数据类型用于存储数值。

Python 支持三种不同的数值类型：整型（int）、浮点型（float）、复数（complex）

**整型：通常称为整数，是整数或者负数，不带小数点。python3整型类型没有大小限制。**int型数字：0、10 -780、-089、-0x270、0x90。

**浮点型：浮点型由整数部分与小数部分组成。**float型数字：0.0、15.2、-2.9、-32.5e10、32+e18。

浮点数可以用科学计算法表示：2.5e2 = 2.5 x 10^2 = 250

**复数型：复数由实数部分和虚数部分构成，可以用a + bj,或者complex(a,b)表示， 复数的实部a和虚部b都是浮点型。**complex:3.14j、9.23e-35j、4.53e-7j。

**python数字类型相互转换**

int(x) 将x转换为一个整数。  
float(x) 将x转换到一个浮点数。  
complex(x) 将x转换到一个复数，实数部分为 x，虚数部分为 0。complex(x, y) 将 x 和 y 转换到一个复数，实数部分为 x，虚数部分为 y。x 和 y 是数字表达式。

**python数字运算**

**和其他不一样的是：  /  ：返回的是浮点数、// :向下取整数、 \*\* ： 幂运算**

1. >>>7/2
2. 3.5
3. >>17//3
4. 5
5. >>>2\*\*3
6. 8

**字符串（String）**

双引号或者单引号中的数据，就是字符串。

1. str1 = "hello,world"
2. str2 = "Jim"

单字符在python中也是当做一个字符串来使用。

**python访问字符串中的值**

1. var1 = "hello world"
2. var2 = "python3"
3. print("var1[0]",var1[0] #var1[0] h
4. print("var2[3:6]",var2[3:6]) #var2[3:6] hon

**python字符串的分片**

通过上边界不包括在内。

1. str = "string"
2. str[1:3] # "tr"
3. #获取从偏移为1到偏移为3的字符串，不包括偏移为3的字符
4. str[1:] # "tring"
5. #获取从偏移为1到最后的一个字符，不包括最后一个字符
6. str[:3] #"str"
7. # 获取从偏移为0的字符一直到偏移为3的字符串，不包括偏移为3的字符串
8. str[:-1] #strin"
9. #获取从偏移为0的字符一直到最后一个字符（不包括最后一个字符串）
10. str[:] #"string"
11. #获取字符串从开始到结尾的所有元素
12. str[-3:-1] #"in"
13. #获取偏移为-3到偏移为-1的字符，不包括偏移为-1的字符
14. str[::-1] # "gnirts"
15. #反转输出

“+”：实现字符串的拼接

string = "hello"+"world" #"hello world"

**字符串的方法**

**replace()方法**

Python replace() 方法把字符串中的 old（旧字符串） 替换成 new(新字符串)，如果指定第三个参数max，则替换不超过 max 次。

1. str = "This is A Test"
2. print(str.replace("is", "was")) # Thwas was A Test"
3. print(str.replace("is", "was", 1)) #Thwas is A Test

**①find()方法**

find() 方法检测字符串中是否包含子字符串 str ，如果指定 beg（开始） 和 end（结束） 范围，则检查是否包含在指定范围内，如果指定范围内如果包含指定索引值，返回的是索引值在字符串中的起始位置。如果不包含索引值，返回-1。

语法：

str.find(str,beg = 0, end = len(str))

参数：

str -- 指定检索的字符串  
beg -- 开始索引，默认为0。  
end -- 结束索引，默认为字符串的长度。  
返回值：

如果包含子字符串返回开始的索引值，否则返回-1。

1. str1 = "python web: www.python.org"
2. str2 = "we"
3. print(str1.find(str2)) # 7
4. print(str1.find(str2, 3)) # 7
5. print(str1.find(str2, 8)) # -1

**②index()方法**

和find()方法相似，唯一的区别就是find方法不包含索引值会返回-1，而index()不包含索引值会抛出异常。

**③join()方法**

连接字符串数组。将字符串、元组、列表中的元素以指定的字符(分隔符)连接生成一个新的字符串。join是重要的字符串方法，用来在队列中添加元素。

**注意：需要添加的元素必须都是字符串。**

语法

str.join(sequence)

1. str = "-"
2. a = [1,2,4]
3. b = ["a","b","c"]
4. print(str.join(a)) #wrong
5. print(str.join(b)) #a-b-c

④split()方法

和join方法相反，split方法把字符串分成序列。

语法

str.split(str="", num=string.count(str)).

str -- 分隔符，默认为所有的空字符，包括空格、换行(\n)、制表符(\t)等。  
num -- 分割次数。  
返回值  
返回分割后的字符串列表。

1. str = "ithis is string examplei"
2. print(str.split()) #['ithis', 'is', 'string', 'examplei']
3. print(str.split("s",1)) #['ithi', ' is string examplei']
4. print(str.split("i")) #['', 'th', 's ', 's str', 'ng example', '']

记住一点儿：str1.split(str2)按照 str2分割时候，就把str2 换成 "," 形式。以" i  ipython  oi" 为例，把字符串的所有 i 换成 ","形式，结果为：" ", " ","python o"," "

**⑤len()方法**

Python len() 方法返回对象（字符、列表、元组等）长度或项目个数。

1. str = "i love python"
2. print(len(str)) #13
3. List = [1,3,5,7]
4. print(len(List)) #4

**⑥lower()方法**Python lower() 方法转换字符串中所有大写字符为小写。

**⑦upper()方法**Python lower() 方法转换字符串中所有小写字符为大写。

**⑧isnumeric()方法**如果 string 中只包含数字字符，则返回 True，否则返回 False

**⑨isspace()方法**如果 string 中只包含空格，则返回 True，否则返回 False.

1. str = "This Is A Test"
2. print(str.lower()) # this is a test
3. print(str.upper()) # THIS IS A TEST
4. str = "tt44333"
5. print(str.isnumeric()) #False
6. str = "44333"
7. print(str.isnumeric()) #True
8. print(str.isspace()) #False
9. str=" "
10. print(str.isspace()) #True

**字符格式化操作**

1. age = 10
2. str = "python"
3. print("今年%d岁"%age,"the string %s"%str)

**列表（List）**

当我们需要存储一组数据时候，就需要用序列，序列给每个元素都分配一个索引，第一个是0，第二个是1，依次论推。常用的序列有：列表和元组，当我们需要改变序列的元素时候，就用列表，因为某些原因，序列不能修改时候，使用元组更加合适。

创建一个列表，只要把逗号分隔的不同的数据项使用方括号括起来即可。列表的数据项不需要具有相同的类型。

1. list1 = ["yqq", "boy", 27]
2. list2 = [1, 2, 3, 4]
3. list3 = ["a", "b", "c"]

python列表的索引

可以通过下标访问列表中的值，也能通过下标来重新赋值。字符串不能通过下边类重新赋值。

1. list1 = ["yqq", "boy", 27]
2. list1[1] = "girl"
3. print(list1) #['yqq', 'girl', 27]
4. str = "4444"
5. str[0]="5"
6. print(str) #wrong

**python列表的分片**

使用索引可以实现单个元素的访问，分片可以操作一定范围的元素。

格式：list[n, m]

n: 提取部分的第一个元素的编号。

m: 分片剩余部分的第一个元素的编号。

第一个是分片的开始，m-n 是分片的个数（n:包含在分片中，m不包含字啊分片中）

1. list = [1,2,3,4,5,6,7,8,9,10]
2. print(list[3:6]) #[4,5,6]
3. print(list[0:1]) #[1]
4. print(list[7:10]) #[8,9,10]
5. print(list[-3:]) #[8,9,10]
6. print(list[-1:]) #[10] #分片操作
7. print(list[-1]) #10 #访问列表的值

可以在分片时候，指定步长，步长可以为正数，也可以是负数。步长为负数，从右向左提取元素。

1. list = [1,2,3,4,5,6,7,8,9,10]
2. print(list[::2]) #[1, 3, 5, 7, 9]
3. print(list[::-1]) #[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
4. #从索引8到索引3，步长为 -1.
5. print(list[8:3:-1]) #[9,8,7,6,5]
6. print(list[0:10:-2]) #[]
7. print(list[5::-2]) #[6, 4, 2]
8. print(list[:5:-2]) #[10,8]

**序列相加**

只有相同类型的序列才能相加

1. >>>[1,2]+[3,4]
2. [1,2,3,4]
3. >>>[1,2,3]+"hello"
4. wrong

**序列迭代**

依次输出列表的元素

1. for x in [1, 2, 3]:
2. print x,

**删除元素**

del 语句来删除元素

1. list1 = ['physics', 'chemistry', 1997, 2000];
2. del list1[2]

**python列表的函数**

len(list) 列表元素个数

max(list) 列表元素的最大值  
max(list) 列表元素的最小值

list(seq)  把seq转化为列表

1. aTuple = (123, 'Google', 'Runoob', 'Taobao')
2. list1 = list(aTuple)
3. print ("列表元素 : ", list1) # 列表元素 :  [123, 'Google', 'Runoob', 'Taobao']
4. str="Hello World"
5. list2=list(str)
6. print ("列表元素 : ", list2) #列表元素 :  ['H', 'e', 'l', 'l', 'o', ' ', 'W', 'o', 'r', 'l', 'd']

**python列表包含的函数**  
list.append(obj)

列表的末尾添加新的对象

1. List =[1,2,3,4]
2. List1 =[3,4,5]
3. List.append(5)
4. print(List) #[1, 2, 3, 4, 5]
5. List.append(List1)
6. print(List)**#[1, 2, 3, 4, 5, [3, 4, 5]]**

list.extend(seq)

在列表末尾一次性追加另一个序列中的多个值

1. List =[1,2,3,4]
2. List1 =[3,4,5]
3. List.extend(List1)
4. print(List)**#[1, 2, 3, 4, 5, 3, 4, 5]**
5. List.append(List1)
6. print(List)**#[1, 2, 3, 4, 5,[ 3, 4, 5]]**

list.insert(index,obj)   
函数用于将指定对象插入列表的指定位置。

1. List =[1,2,3,4]
2. List1 =[3,4,5]
3. List.insert(3, List1)
4. print(List)#[1, 2, 3, [3, 4, 5], 4]

list.pop()  
函数用于移除列表中的一个元素（默认最后一个元素），并且返回该元素的值。  
list.pop(obj=list[-1])

list.remove(obj)

函数用于移除列表中某个值的第一个匹配项。

list.count(obj)

用于统计某个元素在列表中出现的次数。

**元组（Tuple）**

某些情况下，我们需要的序列不可修改，这个时候，就需要用元组，元组和列表相似，**但是元组的元素值不可修改也不能删除**，可以进行分片和连接。

元组创建很简单，用小括号括起来，用逗号隔开。元组使用小括号，列表使用中括号。

1. Tuple = () #创建元组
2. Tuple = ("a","b","c","d")
3. Tuple = ([1,2,3],"a",3,)

注意：一个元素的元组，后边加一个“,”。

Tuple =(30,)

**访问元组**

1. >>> tuple = ("abcd",1,[1,2,4])
2. >>> tuple[0]
3. 'abcd'
4. >>> tuple[1]
5. 1
6. >>> tuple[2]
7. [1, 2, 4]
8. >>>

**修改元组**

1. >>> tuple = ("abcd",1,[1,2,4])
2. >>> tuple[0]="ad" #wrong

**元组合并**

1. >>> tuple = ("abcd",1,[1,2,4])
2. >>> tuple1 =("ad",3,4)
3. >>> tuple + tuple1
4. ('abcd', 1, [1, 2, 4], 'ad', 3, 4)

**删除元组**  
元组的元素不允许删除，但是我们可以删除整个元组。

1. >>> tuple = ("abcd",1,[1,2,4])
2. >>> del tuple

**任意的以“，”分开的序列，默认是元组。**

1. >>> a = 1,2,3,4
2. >>> a
3. (1, 2, 3, 4)
4. >>> a =[1,2,3,4],5,"str"
5. >>> a
6. ([1, 2, 3, 4], 5, 'str')

**元组的函数**  
tuple()方法语法：Python 元组 tuple() 函数将列表转换为元组。

**集合（Set）**

以后补充.......

**字典（Dictionary）**

列表中元素通过下标进行定位，但是元素位置发生变化，则很难定位，python提供一种新的类型，那就是字典。字典中元素可以通过 key 访问。

说明：

字典是由花括号括起来的包含，key : value 两部分。 dict = {'name':'班长', 'id':100, 'sex':'f', 'address':'地球亚洲中国北京'}

字典和列表一样，也能够存储多个数据  
列表中找某个元素时，是根据下标进行的  
字典中找某个元素时，是根据'名字'（就是冒号:前面的那个值，例如上面代码中的'name'、'id'、'sex'）

**访问字典的值**

1. dict = {"name":"yqq", "age":27}
2. print(dict["name"],dict["age"])

**修改字典**

通过 key 来修改值。

1. dict = {"name":"yqq", "age":27}
2. dict["age"] =25
3. print(dict) # {'name': 'yqq', 'age': 25}
4. dict["school"] ="bj"
5. print(dict) # {'name': 'yqq', 'school': 'bj', 'age': 25}

**删除字典或者元素**

1. del dict[key] #删除 key 条目
2. del dict #删除字典
3. dict.clear() #清除所有条目

**字典的方法**

dict.len(),        测量字典中，键值对的个数。

dict.values()    返回一个包含字典**所有value的列表**

dict.keys()     返回一个包含字典所有KEY的**列表**

dict.items()     返回一个包含所有（键，值）**元祖的列表**

1. dict = {'name': 'yqq', 'school': 'bj', 'age': 25}
2. print(len(dict)) #3
3. print(dict.keys()) #dict\_keys(['name', 'age', 'school'])
4. print(dict.values()) #dict\_values(['yqq', 25, 'bj'])
5. print(dict.items()) #dict\_items([('name', 'yqq'), ('age', 25), ('school', 'bj')])

**字典的遍历**

遍历键、遍历值、遍历项、遍历键值对

1. for key in dict.keys():
2. print(key)
3. # name
4. # age
5. # school
6. for value in dict.values():
7. print(value)
8. # yqq
9. # 25
10. # bj
11. for item in dict.items():
12. print(item)
13. # ('name', 'yqq')
14. # ('age', 25)
15. # ('school', 'bj')
16. for key,value in dict.items():
17. print("key=%s"%key,"value=%s"%value)
18. # key=name value=yqq
19. # key=age value=25
20. # key=school value=bj