格式化输出：

%s,%s %(name,age)

UTF-8:

英文：一个字节一个字符 0000 0001

欧洲：两个字节 0000 0000 0000 0001

亚洲：三个字节 0000 0000 0000 0000 0000 0001

GBK:

英文：一个字节 0000 0001

中文：两个字节 0000 0000 0000 0001

**基础数据类型：**

Int:用于计算运算

Bool：用于判断真假

Str：存储少量的数据

索引：str[1]

切片：str[0:6] str[:6] str[-1,-3,-1]

Str.capitalize():首字母大写，其他小写

Str.swapcase:交换字母大小写

Str.center(20,’\*’):将字符串居中，其余填充为第二个参数

Str.title():首字母大写，字符串非字母元素隔开的每个单词的首字母大写

Str.lower（）Str.upper()

Str.startswith() Str.endswith()

Str.find(“”) Str.find(“”，1,2) 某个字符的索引

Str.strip() Str.strip(‘’) Str.lstrip() Str.rstrip()

Str.splite() Str.rsplite() Str.lsplite()

Str.replace(‘’,’’,1)

‘\_’.join(str) 将str中的每个字符用\_连起来

格式化：msg = ‘我是{name},爱好{hoppy }’

Mag.format(name=’’,hoppy=‘’)

Str.isalum() #字符串是否有字母和数字组成

Str.isalpha() #字符串是否有字母组成

Str.isdigit() #字符串是否有数字组成

Str.count(‘a’) # 计算字符串中某元素出现的个数

Len(str) # str的长度

For i in str:

Print(i)

List：有序的数列，可以存储大量数据

List[1,6] # 切片，顾首不顾尾

List.append() #追加

List.pop() #按照索引删除，有返回值

List.remove(“”)#按照元素删除

List.clear() #全部清空

Del List[1,6] #可以按照切片删除

List[0] = ‘’#按索引修改

List[1::2] #按切片、步长修改

计数某元素出现的次数：list.count(“”)

通过元素找索引：list.index(“”,start,stop)

排序:list.sort() #默认从小到大排序

list.sort(reverse=True) #默认从大到小排序

翻转顺序:list.reverse()

Range(a,b) #顾首不顾尾

Range(a,b,n)

删除列表中索引为奇数的元素:del list[1::2] 或者反向循环删除

Tuple：有序的数列，可以存储大量数据，为只读列表

存储一些固定的重要的数据，不能被增加删除修改，只能查询。

其他功能和列表一样

可变的数据类型：list,dict,set

不可变的数据类型：int bool str tuple

如果元组中只有一个元素且没有逗号，那么类型不是元组，而是该元素的类型

Dict：无序的类型，可以存放大量的关系型数据。

List,tuple查询速度相对满，不能储存关联性较强的数据。

字典的查询速度很快，遵循哈希算法。

字典的键不能重复

增：dict[“”] = “”;

默认值：dict.setDefault(‘’,)

删:dict.pop(‘’，没有此键的提示) #按照键删除，返回对应的值

Del dict[‘’]

清空:dict.clear()

更新：dict.update(dict2) #将dict2中的所有键值添加覆盖到dict

查询:ret = dict.get(‘’) #不会报错

赋值操作:a,b = b,a

获取所有键:dict.keys()

获取所有值:dict.values()

获取所有item:dict.items() #每个item都是元组

For k,v in dict.items:

Print(k,v)

创建字典:dict.fromkeys([1,2,3],”qq”)

在循环一个字典的时候，不能改变字典的长度

如果要删除某些键值对，可将键存入列表。

Set：无序的类型，不重复，主要用于关系测试，交集，并集等

其中智能存储不可变的数据类型

增删改查：set1.add()

Set1.update() #迭代着增加

······set1.remove()

Set1.pop #随机删除，返回被删除的元素

Set1.clear()

Del Set1

关系测试:

交集:set1 & set2 或者set1.intersection(set2)

并集:set1 | set2 set1.union(set2)

差集;set1 – set2 set1.difference(set2)

对称差集: set1 ^ set2 # 反交集

Set1 > set2 或者 ser1.issuperset(set2) #set1是se2t的父集合

Set2 < set1 或者 set2.issubset(set1) #子集合

冻集合:fro = forzenset(set1)

**编码**

ASCII：一个字符需要一个字节

Unicode：一个字符需要四个字节。

Utf-8：英文：一个字符需要一个字节

欧洲：一个字符需要两个个字节

中文：一个字符需要三个字节

Gbk:国标

英文：一个字符需要一个字节 中文：一个字符需要两个个字节、

Bytes类型:

Str有的方法,bytes都有

Str的内部编码方式是unicode,

Byte:表现形式:b’alex’,编码:非unicode

编码：unicode转化成非unicode: encode

非unicode转化成unicode: decode

8bytes = 1KB

1024k = 1M

对象与代码块:

== :内容是否相等

Is：是否是同一个对象

缓存机制：

针对：int,str,bool

同一个代码块下，值相同的变量指向同一个内存

节省内存，提高性能

浅copy:l1和l2的内存地址不一样，但是他们的子元素内存地址一样。

L1 = []

L2 = l1.copy()

深层copy: l1和l2的内存地址不一样，但是他们的子元素（int,bool,str）一样，其他类型的数据是重新创建的。

L2 = deepcopy(l1)