Python:

1. 基本数据类型：
2. 列表list: lis = [1, ’a’]

元素可重复、有序、长度可变：添加- list.append(元素)，

读元素lis[1: ]，去重转化成集合：set1= set(lis)

1. 集合set：set2= set([1,2]) 或 {1,2}

元素不重复、无序（不支持索引操作）、容纳量可变：添加-add(元素)，

读元素 需转化成列表后读：lis2 = list(set2)

1. 元祖tuple：(1, )

元素可重复、有序、不可变(元组本身不可变，如果元组中还包含其他可变元素，这些可变元素可以改变)、读取元素：t[0], t[1: 3]

1. 字典dict：dict1={'a’,:1, ‘b’:2 }

键唯一且不可变类型数据，无序自动正序，

增加： info["stu1104"] = "Lucy"

修改：info["stu1101"] = "武滕兰"

查找："stu1102" in info #标准用法 ，返回True

获取：info.get("stu1102")#如果有，返回值，如果没有返回空None

　　　info['shanshan']#如果有，返回值，如果没有报错，一般不用这个

删除：info.pop()

　　(1)info.pop('shanshan') #删除后返回删除值的信息

　　(2)info.popitem()#随机删除，没有顺序

(3)del info["stu1102"]

Python: 序列-6个内置类型（列表|元组|字符串|Unicode字符串|buffer对象|xrange 对象）

1. 序列通用的操作：

索引、长度、组合（序列相加）、重复（乘法）、分片、检查成员、遍历、最小值和最大值。

列表和元组：

相同点：

1、均具有序列的特性，均可以进行序列通用的操作；

2、数据项不需要具有相同的类型；

3、均包含内置函数max、min、len；

4、均可以转换为对方：1）tuple1 = tuple(lis1)；2）lis1 = list(tuple1)

异同点：

1）列表可修改2）元组不可修改

内置方法

1）列表提供11个内置方法list.append(obj)、list.count(obj)、list.extend(seq)、list.index(obj)、list.insert(index, obj)、list.pop([index=-1]])、list.remove(obj)、list.reverse()、list.sort(cmp=None, key=None, reverse=False)、list.clear()、list.copy()；

2）元组未提供内置方法

* **List:**

索引：ls[0]

切片：ls[1:5]，

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

a[-1]=5

a[:-1]= [1, 2, 3, 4]

a[-2:]=[4, 5]

a[::-1]=[5, 4, 3, 2, 1]，倒序列表

其他列表排序方法：

list.sorted(iterable, key=None, reverse=False); reverse: False升序,True降序

list.sort(cmp=None, key=None, reverse=False);

key:一个只有一个参数的函数，这个函数会被用在序列里的每一个元素上，所产生的结果将是排序算法依赖的对比关键字。

Key=lambda x:x\*-1

key=str.lower

key=str.len

更新：ls.append(‘值’)

删除：del ls[索引值]

合并列表List1+list2；重复列表：list(index)\*2

* **tuple**

索引：t[0]

切片：t[1:5]

合并：t1+t2, （‘hi’，）\*4=（‘hi’,‘hi’）

删除整个元祖：del t1

内置函数：len（t1）；max(t1) 返回最大值元素；min(t1)；tuple(iterable)转换元组

* **dict**

修改：dict[‘key’]=value

删除：del dict[‘key1’] #删除键 或者 pop[‘key1’]

del dict #删除字典

dict.clear() #清空字典

键：不可变对象-字符串、元组、数字等

内置方法：len(dict1) ; str(dict1)输出字典；

radiansdict.copy() 返回一个字典浅复制

dict.fromkeys(keylist, default=none)

dict.get(key,default=none)

dict.setdefault(key, default=none):

dic.setdefault(key, []).append(元素)#更新value为列表的字典；dic.setdefault(key, {})[vkey]=vv

区别：当查找的key值不存在，则返回默认值并更新dict, get只会返回并不跟新dict:

if key in dict: ;return T/F

dict.items() 返回（key,value）元组数组列表

dict.keys() 返回键的迭代器，list(dic.keys())转化成列表操作

dict.values() 同上

合并：dic1.update(dic2) 跟新2到1中

pop(key)

popitem()

* **集合set**

set()创建空集合，{}创建空字典

操作运算符：- | & ^:同时不包含ab集合的

集合推导式 set1={x for x in ‘adkjhfksd’ if x not in ‘abc’}

添加元素：set1.add(元素) 或 set1.update({1,2})

移除：set1.remove(x) / set1.discard(x) 如果元素不存在第一种将报错

set1.pop() 随机删除一个元素

内置：

set1.clear() 清空集合

len(set1)

x in set1

copy()

set1.difference(set2)等于 set1–set2 都是返回一个新的差集

set1.difference\_update（set2） 直接更新set1,set1=set1-set2

& intersection() intersection\_update()

isdisjoint() a,b中是否包含相同的元素，是返回true

x.issuperset(y) y是否为x的子集

union() 并集 |

* **字符串**

+，\*， in，not in；print(r’\n’) 转义字符失效；%字符串格式化 print(“%s%d”%(‘xiaoming’,10))

\\ \n换行；\r 回车；

字符串格式化也可用：str.format(var) var可以是str, dict,list,,,,或者对象

“名字:{key1}, {key2}”.format(\*\*dic)

方法：

string.count(str, beg=0, end=len(string)) 返回str在string中出现的次数

**string.endswith(str1, beg=0,end=len(string)) 返回trun false**

**String.startswith(,,)**

**String.find(str,beg=0,end=len(sttring)) 返回找到str的开始的位置，没找到返回-1**

**Str.rfind(str) 同上，返回str最后一次出现的位置**

**String.index(str,,,) 同上，找不到会报异常**

**String.isalnum() string至少有一个字符，且所有字符都是字母或数字返回True, 否则false**

**Sring.isdecimal() 十进制**

**Str.isdigit() 数字**

**str.islower()**

**Str.isuper()**

**Str.isnumeric()**

**Str.isspace()**

**Str.lower() 所有为小写**

**Str.upper()**

**Str.swapcase() 翻转str中的大小写**

**Str.join(seq) str为分隔符连接seq中的字符串**

**Str.maketrans(intab, outtab) 搭配str.translate(mt)使用**

**Str.translate(str1,del=””) 根据str1给出的表转换string的字符，要过滤掉的字符放到del参数中**

**max(str)**

**min(str)找出str中最小的字母**

**Str.replace(str1,str2) str替换str1**

**Str.split(str=””,num=) 以str为切片分隔string, 分隔num+个字符串**

**Str.strip([chars]) 移除str字符串的开头或结尾中的[chars]，不考虑顺序**

**time模块和calendar模块**

**当前时间戳：time.time()**

**time.asctime(time.locatime(time.time())) asctime获取格式化的时间，time.locatime获取时间元组**

**格式化日期：time.strftime(“%Y-%m-%d %H:%M:%S”, time.localtime())**

**格式化的字符串转化成时间戳：time.mktime(time.strptime(a,”%Y-%m-%d %H:%M:%S”))**

**Calendar.month(2016,1)**

**os 模块**



ast模块

帮助Python应用来处理抽象的语法解析的。

ast.literal\_eval()：则会判断需要计算的内容计算后是不是合法的Python类型，如果是则进行运算，否则就不进行运算。

**读取大文件**

获得大文件内容行数：统计文件中换行符的个数'\n '(或者包含'\n'的字串,如在windows系统中):

count = 0

thefile = open(thefilepath, **'rb')**  //参数'rb'是必须的,否则在windows系统上,上面的代码会非常慢.

while True:

**buffer = thefile.read(8192\*1024)**

if not buffer:

break

**count += buffer.count('\n')**

thefile.close( )