

Python

1 添加新元素

`append()`总是把新的元素添加到list的尾部。

`L.insert(0,'Paul')`的意思是'Paul'将被添加到索引为0的位置上，其他同学自动向后移一位。

2 删除元素

`pop()`方法总是删掉list的最后一个元素，并且它还返回这个元素，所以我们执行`L.pop()`后，会打印出删掉的元素。

`L.pop(n)`把第n+1个元素删掉。

3 替换元素

`L[2]='Paul'`和`L[-1]='Paul'`都可以

4 创建tuple

tuple一旦创建就不能修改了，创建时与list唯一差别是用`()`代替`[]`。

5 创建单元素tuple

空tuple: `t=()`打印出`()`

`t=(1)`打印出1: 因为 `()` 既可以表示tuple，又可以作为括号表示运算时的优先级。

`t=(1,)`打印出 `(1,)`

6 可变的tuple

tuple所谓的不变是说tuple的每个元素指向永远不变，指向一个list就不能改成指向其他对象，但指向的这个list本身可变。

7 if语句

缩进规则：具有相同缩进的代码被视为代码块。

4个空格，不要使用tab

if后接表达式，然后用：表示代码块开始。

Python交互环境下，推出缩进需要多敲一行回车。

8 if-else

利用if-else语句，我们可以根据条件表达式的值为true或false分别执行if代码块或者else代码块。

9 if-elif-else

这一系列的条件判断会从上到下依次判断，如果某个判断为true，执行完对应的代码块，后面的条件判断就直接忽略，不再执行了。

10 for循环

```
L=['Adam','Lisa','Bart']  
for name in L:  
    print name
```

name这个变量是在for循环中定义的，意思是，依次取出list中的每一个元素，并把元素赋值给name，然后执行for循环体（就是缩进的代码块）

11 while循环

while循环不会迭代list或tuple的元素，而是根据表达式判断循环是否结束。

while循环每次先判断条件是否满足，如果为true，则执行循环体的代码块，否则，退出循环。

12 break退出死循环

用for循环或者while循环时，如果要在循环体内直接退出循环，可以使用break语句。可以在死循环内设if语句，满足时break即可退出循环。

13 continue继续循环

continue可以跳过后续循环代码，继续下一次循环。

```
L=[75,98,59,81,66,43,69,85]
sum=0.0
n=0
for x in L:
    if x<60:
        continue
    sum=sum+x
    n=n+1
print sum/n
```

14 多重循环

```
for x in ['A','B','C']:
    for y in ['1','2','3']:
        print x+y
```

x每循环一次，y就会循环3次。可以利用多重循环打印出全排列。

15 什么是dict

组成查找表

```
d={
    'Adam':95,
    'Lisa':85,
    'Bart':59
}
```

我们把名字称为key,对应的成绩称为value,dic就是通过key来查找value;表示这是一个dic,然后按照key:value，写出来即可。最后一个key:value的逗号可以省略。

由于dic也是集合，len()函数可以计算任意集合的大小len(d)

16 访问dict

可以简单的使用d[key]的形式来查找对应的value，这和list很像，不同之处是，list必须使用索引返回对应的元素，而dict使用key.

一是判断key是否存在

```
if 'Paul' in d:
    print d['Paul']
```

如果'Paul'不存在，if语句判断为False，自然不会执行print，从而避免了错误。

二是使用dict本身提供的一个get方法，在key不存在的时候，返回none

```
print d.get('Bert')
```

17 dict的特点

key不能重复；存储的key-value序对是没有顺序的；key的元素必须不可变。

```
{
    '123': [1,2,3], # key是str,value是list
    123: '123', # key是int,value是str
    ('a','b'): True #key是tuple,并且tuple的每个元素都是不可变对象，value是boolean
}
```

18 更新dict

dict是可变的，也就是说，我们可以随时往dict中添加新的key-value；如果key已经存在，则赋值会用新的value值替换掉原来的value

19 遍历dict

由于dict是一个集合，所以，遍历dict和遍历list类似，都可以通过for循环实现。

直接使用for循环可以遍历dict的key:

```
d={'Adam':95,'Lisa':85,'Bert':59}
for key in d:
    print key
```

由于通过key可以获取对应的value,因此，在循环体内，可以获取到value的值。

20 set

dict的作用就是建立一组key和一组value的映射关系，dict的key是不能重复的。

set持有一系列元素，这一点和list很像，但是set的元素没有重复，而且是无序的，这点和dict的key很像。

创建set的方式是调用set()并传入一个list,list的元素将作为set的元素：

```
s=set(['A','B','C'])
```

可以查看set的内容

```
print s
set(['A','B','C'])
```

因为set不能包含重复的元素，所以，当我们传入包含重复元素的list，set会自动去掉重复的元素。

21 访问set

由于set存储的是无需集合，所以我们没法通过索引来访问。访问set中的某个成员实际上就是判断一个元素是否在set中。

我们可以用in操作符判断，大小写不同会被认为是两个不同的元素。

22 set的特点

set的内部结构和dict很像，唯一区别是不存储value，因此，判断一个元素是否在set中速度很快。

set存储的元素和dict的key类似，必须是不变对象，因此，任何可变对象是不能当如set中的。

最后，set存储的元素也是没有顺序的。

```
weekdays=set(['MON','TUE','WED','THU','FRI','SAT',
'SUN'])
#再判断输入是否有效，只需要判断该字符串是否在set中：
x='???'
if x in weekdays:
    print 'input ok'
else:
    print 'input error'
```

23 遍历set

元素的顺序和list的顺序很可能是不同的

```
s=set(['Adam','Lisa','Bart'])
for name in s:
    print name
```

24 更新set

一是把新的元素添加到set中，二是把已有元素从set中删除

添加元素时，用set的add()方法：

```
s=set([1,2,3])
s.add(4)
print s
set([1,2,3,4])
```

如果添加的元素已经有了，不会报错，但是不会加进去了

删除set中的元素时，用set的remove()方法：

```
s=set([1,2,3,4])
s.remove(4)
print s
set([1,2,3])
```

如果删除的元素不存在set中，remove()会报错