

2. 列表

列表（List）是最常用的 Python 数据类型，用来存储多个数据，特点如下：

- 存储的数据是有序的。
- 索引从 0 开始，即第一个数据的索引为 0。
- 数据值可变。
- 所含元素数量可变。
- 元素类型可不同。

记住以上的特点，操作起列表就容易多了。

（1）创建列表

创建列表语法如下：

【语法】

列表名 = [数据 1, 数据 2, ……]

如下所示，两个列表分别存储了员工姓名和员工工资：

```
#创建列表
employee = ['王保华', '李伟新', '张强', '陈鑫然']
salary = [4000, 5600, 3400, 6000]
```

（2）遍历列表元素

在实际的开发中，遍历列表元素是非常常用的操作，应为有时我们并不知道全部的列表元素个数，那么如何获得全部信息呢？这时需要与 for 结构配合来遍历列表元素。看下面这个需求。

按照列表元素的顺序遍历输出所有工资。

```
employee = ['王保华', '李伟新', '张强', '陈鑫然']
salary = [4000, 5600, 3400, 6000]
for s in salary:
    print(s)
```

输出结果为：

4000

5600

3400

6000

（3）访问列表元素

列表的元素数据是有序，访问列表单个元素使用索引。列表索引有两种表现形式。

- 正向：第 1 个元素索引值为 0，依次进行，最后一个元素索引值为（列表长度-1）
- 反向：最后一个元素索引值为-1，依次进行，第 1 个元素索引值为（-列表长度），注意负号。

访问列表元素的语法如下：

【语法】

变量名 = 列表名[索引]

如下所示，打印输出第二名员工李伟新的工资：

```
#创建列表
employee = ['王保华', '李伟新', '张强', '陈鑫然']
salary = [4000, 5600, 3400, 6000]
print('员工姓名:', employee[1])
print('工资:', salary[1])
```

输出结果为：

员工姓名： 李伟新

工资： 5600

除了访问某个元素外，也可以使用方括号的形式截取元素，如下所示：

```
#创建列表
employee = ['王保华', '李伟新', '张强', '陈鑫然']
print(employee[1:3])
print(employee[1:])
```

输出结果为：

['李伟新', '张强']

['李伟新', '张强', '陈鑫然']

可以看一下访问列表元素的视频操作，动动手，自己也练习一下。

问题又来了，如果要在遍历的过程中修改元素值，或者获取当前元素的索引值，该怎么办呢？可以使用 `len()` 方法来获取列表长度，再用 `range()` 生成遍历列表的索引数列，通过 `for` 循环就可以了。

修改所有月薪小于 5000 的员工薪水，变更为 5000。

```
employee = ['王保华', '李伟新', '张强', '陈鑫然']
salary = [4000, 5600, 3400, 6000]
for index in range(0, len(salary)):
    if salary[index] < 5000:
        salary[index] = 5000
print(salary)
```

输出结果为: [5000, 5600, 5000, 6000]

（4）更新列表元素值

列表允许对其中的元素进行增删改查的操作，前面已经介绍了查，现在来学习增删改该如何实现。

下表中列举了常用的更新列表的方法。

方法名	说明
<code>append(obj)</code>	在列表末尾添加元素 <code>obj</code>
<code>insert(index,obj)</code>	指定索引 <code>index</code> 的位置插入新元素 <code>obj</code> ，其后元素依次后移
<code>pop(index = -1)</code>	删除列表中最后一个元素，并返回该元素的值
<code>del list[index]</code>	删除指定索引为 <code>index</code> 的元素，其后元素依次前移
<code>list[index] = obj</code>	修改索引为 <code>index</code> 的元素值为 <code>obj</code>

针对刚才的员工工资信息，我们完成如下需求：

- 程曦入职，工资 7900
- 陈鑫然离职，删除其信息
- 王保华涨工资为 6000

实现代码如下：

```

employee = ['王保华', '李伟新', '张强', '陈鑫然']
salary = [4000, 5600, 3400, 6000]
#程曦入职, 工资 7900
employee.append('程曦')
salary.append(7900)
print(employee, ' ', salary)
#陈鑫然离职, 删除其信息
del employee[3]
del salary[3]
print(employee, ' ', salary)
#王保华涨工资为 6000
salary[0] = 6000

```

运行结果如下图。

```

Test x
D:\23Python\Study\venv\Scripts\python.exe D:/23Python/Study/mypackage/Test.py
['王保华', '李伟新', '张强', '陈鑫然', '程曦']      [4000, 5600, 3400, 6000, 7900]
['王保华', '李伟新', '张强', '程曦']      [4000, 5600, 3400, 7900]
['王保华', '李伟新', '张强', '程曦']      [6000, 5600, 3400, 7900]
>> Process finished with exit code 0
278 chars, 12 line breaks  127:1  CRLF UTF-8

```

（5）常见操作列表的函数

前面列举了常用的更新列表的方法，还有其他一些方法也是非常常用。

方法名	说明
len(list)	返回列表 list 的元素个数
max(list)	返回列表元素最大值
min(list)	返回列表元素最小值
list(seq)	将元组 seq 转换成列表结构（元组内容接下来介绍）
list.count(obj)	统计元素 obj 在列表 list 中出现的次数
list.index(obj)	返回元素 obj 在列表 list 中第一次出现的索引值
list.clear()	清空列表 list