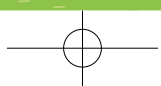
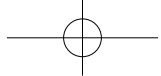


第 5 单元

混合的队伍

——列表





5.1 知识点定位

青少年编程能力等级 Python 一级中的核心知识点 5：列表类型。

5.2 能力要求

掌握并熟练编写带有列表类型的程序，具备解决一组数据处理基本问题的能力。

5.3 建议教学时长

本单元建议 3 课时。

5.4 教学目标



知识目标

本单元以列表类型学习为主，通过联系生活案例，让学生理解列表的概念及创建方法，列表元素的索引及访问，了解列表元素的控制方法，为后续循序





渐进地学习程序编写打好基础。



能力目标

通过对 Python 列表类型的学习，学会借助列表类型来解决现实中的问题，锻炼学习者从计算机的角度去思考问题，培养计算思维能力。



素养目标

具有一定信息素养，能够合理运用计算机去解决其他课程中的问题，增强社会主义核心价值观的认识，增强学习者的爱国主义情怀。

5.5 知 识 结 构

本单元的知识结构如图 5-1 所示。

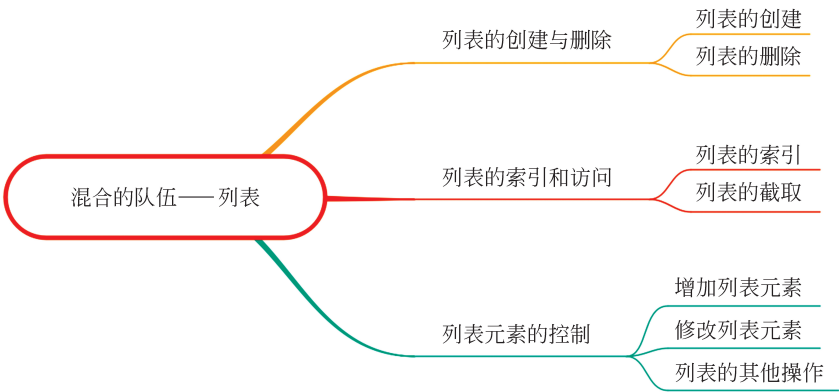
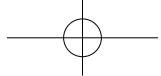


图 5-1 列表的知识结构





5.6 课程补充知识点



Python 列表元素的遍历

列表的遍历主要通过 for-in 语句实现。

格式 1 如下：

```
for <任意变量名> in <列表名>:  
    <语句块>
```

格式 2 如下：

```
for <任意变量名> in enumerate(<列表名>):  
    <语句块>
```

注：enumerate(sequence, [start=0])，返回枚举对象

参数

- Sequence——一个序列、迭代器或其他支持迭代对象。
- Start——下标起始位置。

格式 3 如下：

```
for <任意变量名> in iter(<列表名>):  
    <语句块>
```

注：iter(object[, sentinel]) 函数用来生成迭代器，返回迭代对象。

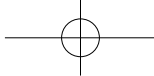
参数

- object——支持迭代的集合对象。
- sentinel——如果传递了第二个参数，则参数 object 必须是一个可调用的对象（如，函数），此时，iter 创建了一个迭代器对象，每次调用这个迭代器对象的 __next__() 方法时，都会调用 object。

格式 4 如下：

```
for <任意变量名> in range(len(<列表名>)):  
    <语句块>
```





注：range(start, stop[, step]) 函数返回类型是 ndarray，可用 list() 返回一个整数列表，一般用在 for 循环中。

参数

- start——计数从 start 开始。默认是从 0 开始。例如，range(5) 等价于 range(0, 5)。
- end——计数到 end 结束，但不包括 end。例如，range(0, 5) 是 [0, 1, 2, 3, 4]，没有 5。
- step——步长，默认为 1。例如：range(0, 5) 等价于 range(0, 5, 1)。

列表遍历示例程序代码如下：

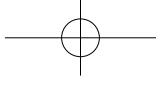
```
list=['one','two','three']
print("***** 格式1 实例 *****")
for i in list:
    print(i)
print("***** 格式2 实例 *****")
for i in enumerate(list):
    print(i)
print("***** 格式3 实例 *****")
for i in iter(list):
    print(i)
print("***** 格式4 实例 *****")
for i in range(len(list)):
    print(list[i])
```

运行结果如图 5-2 所示。

```
*****格式1实例*****
one
two
three
*****格式2实例*****
(0, 'one')
(1, 'two')
(2, 'three')
*****格式3实例*****
one
two
three
*****格式4实例*****
one
two
three
```

图 5-2 列表遍历示例运行结果





列表 (list) 推导式

Python 推导式是一种独特的数据处理方式，可以从一个数据序列构建另一个新的数据序列的结构体。列表 (list) 推导式即为其中之一，其格式如下：

[表达式 for 变量 in 对象]

或者

[表达式 for 变量 in 对象 if 条件]

例如：

```
>>> list1=[a for a in range(11)]
>>> print(list1)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
>>> list2=[a for a in range(21) if a%3==0]
>>> print(list2)
[0, 3, 6, 9, 12, 15, 18]
>>> week=["Mon","Tues","Wed","Thurs","Fri","Sat","Sun"]
>>> new_week=[w.upper() for w in week]
>>> print(new_week)
['MON', 'TUES', 'WED', 'THURS', 'FRI', 'SAT', 'SUN']
```



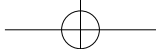
Python 列表类型特有的函数或方法

Python 列表类型特有的函数或方法如表 5-1 所示。

表 5-1 Python 列表类型特有的函数或方法

函数或方法	描 述
<code>ls[i] = x</code>	替换列表 <code>ls</code> 第 <code>i</code> 数据项为 <code>x</code>
<code>ls[i:j] = lt</code>	用列表 <code>lt</code> 替换列表 <code>ls</code> 中第 <code>i~j</code> 项数据（不含第 <code>j</code> 项，下同）
<code>ls[i:j:k] = lt</code>	用列表 <code>lt</code> 替换列表 <code>ls</code> 中第 <code>i~j</code> 以 <code>k</code> 为步长的数据
<code>del ls[i:j]</code>	删除列表 <code>ls</code> 第 <code>i~j</code> 项数据，等价于 <code>ls[i:j]=[]</code>
<code>del ls[i:j:k]</code>	删除列表 <code>ls</code> 第 <code>i~j</code> 以 <code>k</code> 为步长的数据
<code>ls += lt</code> 或 <code>ls.extend(lt)</code>	将列表 <code>lt</code> 元素增加到列表 <code>ls</code> 中
<code>ls *= n</code>	更新列表 <code>ls</code> ，其元素重复 <code>n</code> 次





续表

函数或方法	描 述
<code>ls.append(x)</code>	在列表 <code>ls</code> 最后增加一个元素 <code>x</code>
<code>ls.clear()</code>	删除 <code>ls</code> 中所有元素
<code>ls.copy()</code>	生成一个新列表，复制 <code>ls</code> 中所有元素
<code>ls.insert(i, x)</code>	在列表 <code>ls</code> 第 <code>i</code> 的位置增加元素 <code>x</code>
<code>ls.pop(i)</code>	将列表 <code>ls</code> 中第 <code>i</code> 项元素取出，并删除该元素
<code>ls.remove(x)</code>	将列表中出现的第一个元素 <code>x</code> 删除
<code>ls.reverse(x)</code>	列表 <code>ls</code> 中元素反转



Python 的列表和数组

列表是 Python 的内置数据类型，是 Python 中最基本的数据结构——序列中的一种，它提供一个或多个表示一组元素的方法；Python 原生没有数组的概念，要使用数组需通过 `import array` 或者 `import numpy` 中的 `array` 来引进数组。

Python 中原生的列表虽然使用方法与数组类似，但它们之间有本质区别。创建列表时不需要指定列表长度，可以在使用中动态插入任何数量的元素；也不需要指定数据类型，列表中不同元素的类型可以相同，也可以不同。而创建数组时必须指定数组长度和数据类型。另外，列表还具有诸如索引、切片、更新、删除等操作，也为程序编写提供了更大的设计空间，例如，绘图时可以将颜色值作为列表元素，绘制出多彩的图形，如图 5-3 所示。



```
colors=['red','orange','yellow','green','indigo','blue','purple']
```

图 5-3 七彩蟒蛇



程序资源

本单元配套了“列表遍历示例.py”“七彩蟒蛇绘制.py”等程序，供授课教师选择演示，以激发孩子的学习兴趣。





5.7 教学组织安排

教学环节	教学过程	建议时长
知识导入	使用一个小火车玩具引入列表的概念	1 课时
列表的创建与删除	通过提问、讨论、测试、动手操作等互动及实践掌握列表创建与删除的方法	
列表的索引和访问	通过和字符串的对比、采用代码演示、动手操作、提问等互动熟悉列表的索引和访问方法	1 课时
列表元素的控制引入	通过猜拳游戏引入列表元素的使用	1 课时
列表元素的控制	采用代码演示、动手操作、提问等互动熟练掌握列表元素控制的方法	
单元总结	以提问方式总结本次课所学内容，布置课后作业	

5.8 教学过程设计



1. 游戏式知识导入

用一个火车玩具，和学生玩一个火车运货物的游戏，顺带引入列表的概念。



2. 知识点一：列表的创建

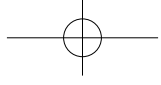
- (1) 通过主教材小实例的演示介绍列表创建的方法。
- (2) 问答式完成主教材第 5 单元上“想一想”中[问题 5-1](#)，介绍空列表。



3. 知识点二：列表的删除

- (1) 通过火车卸货演示比拟列表元素的删除。





- (2) 通过命令操作演示并讲解 `del` 语句的使用方法。
- (3) 通过命令操作演示并讲解 `pop()` 函数的使用方法。
- (4) 通过命令操作演示并讲解 `remove()` 函数的使用方法。
- (5) 通过命令操作演示并讲解 `clear()` 函数的使用方法。
- (6) 测试方式完成“来找茬”问题，对比 `pop()` 函数和 `remove()` 函数的异同，强化学生对这两个函数使用方法的理



知识点三：列表的索引和访问

- (1) 以提问的方式带领学生回忆字符串的索引概念，以对比方式引入列表的索引概念。
- (2) 用提问的方式让学生说出主教材第5单元中列表引用命令的结果。
- (3) 运行主教材例5-1的程序，掌握列表在实际中的运用方式。
- (4) 让学生完成主教材第5单元“练一练”中问题5-2，并和其他同学分享自己的程序代码。
- (5) 简单介绍扩充知识：使用 `for-in` 语句实现列表元素的遍历。
- (6) 运行主教材例5-2的程序，掌握列表的截取方法。



知识点四：增加列表元素

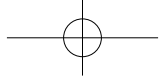
- (1) 讲解主教材例5-3的程序代码，并运行。
- (2) 选两个学生来玩猜拳游戏，并让学生仿照主教材例5-3编写程序代码，记录游戏过程及双方得分。
- (3) 通过命令操作演示及提问的方式讲解 `insert()`、`extend()` 函数和“+”运算符的使用方法。
- (4) 以课堂讨论的方式让学生提出使用其余函数或方法（除 `append()` 外）修改主教材例5-3的方法。
- (5) 以课堂练习的方式让学生完成主教材第5单元“练一练”中问题5-3，并作分享讲解。



知识点五：修改列表元素

通过命令操作演示及提问的方式讲解列表修改的方法。





知识点六：列表的其他操作

- (1) 简单介绍主教材第 5 单元表 5-2 里的列表函数或方法。
- (2) 运行主教材例 5-4 程序，讲解 sort() 函数的使用方法。
- (3) 以课堂练习的方式让学生完成主教材第 5 单元“练一练”中问题 5-4，并作分享讲解。
- (4) 通过命令操作演示及提问的方式讲解求最大、最小等函数的使用方法。



单元总结

小结本次课的内容，布置课后作业。



5.9 拓展练习



成绩统计

设计一个程序，输入全班 10 个人的语文成绩，统计并输出该课程的总分、平均分、最高分及最低分，程序运行结果如图 5-4 所示。

```
请输入成绩: 76
请输入成绩: 78
请输入成绩: 86
请输入成绩: 90
请输入成绩: 93
请输入成绩: 92
请输入成绩: 90
请输入成绩: 90
请输入成绩: 85
请输入成绩: 91
语文考试的总分为871，平均分为87.1，最高分为93，最低分为76
```

图 5-4 成绩统计运行结果



名字搜索

在经过一轮初赛决定了进入演讲比赛决赛的同学名单，编写一个程序查

