

## 第五课 Python中可变类型的数据集合

# 课时介绍

◆ 列表简单介绍

◆ 列表的应用

◆ 字典的基本操作

◆ 列表常用操作

◆ 字典的简单介绍

◆ 字典的应用

# 课程目标

- ◆ 掌握列表的特性与使用方法
- ◆ 掌握字典的特性与使用方法

# 数据结构

◆ 列表(List)

◆ 元组(Tuple)

◆ 数据结构就是指从计算机存储、组织数据的结构

◆ 字典(Dictionary)

◆ 集合(Set)

# 列表(List)

- ◆ 列表中的数据按顺序排列
- ◆ 列表有正序与倒序两种索引
- ◆ 列表可存储任意类型数据，且允许重复

# 列表(List)

-7	-6	-5	-4	-3	-2	-1
a	b	c	1	2	3	4
0	1	2	3	4	5	6

# 创建列表



## 列表的取值

-6	-5	-4	-3	-2	-1
张三	李四	王五	赵六	钱七	孙八
0	1	2	3	4	5



# for..in语句

- ◆ for..in语句专门用于遍历列表、元组等数据结构

for 迭代变量 in 可迭代对象

循环体(必须缩进)

# 列表的反转与排序



# 冒泡排序

原始值

28	32	14	12	53	42
----	----	----	----	----	----

28	32	14	12	53	42
28	32	14	12	53	42
28	14	32	12	53	42
28	14	12	32	53	42
28	14	12	32	53	42
28	14	12	32	53	42

第一次循环

28	14	12	32	42	53
14	28	12	32	42	53
14	12	28	32	42	53
14	12	28	32	42	53
14	12	28	32	42	53
14	12	28	32	42	53

第二次循环

最终结果

12	14	28	32	42	53
----	----	----	----	----	----

# 列表的新增、修改、删除操作

函数名	描述
<code>list.append(新元素)</code>	在列表末端追加新元素
<code>list.insert(索引,新元素)</code>	在指定索引插入新元素
<code>list[索引] = 新值</code>	更新指定索引位置数据
<code>list[起始索引:结束索引] = 新列表</code>	更新指定范围数据
<code>list.remove(元素)</code>	删除指定元素
<code>list.pop(索引)</code>	按索引删除指定元素

# 列表的其他常用方法



# 项目中的使用场景



# 列表存储数据的问题

- ◆ 列表在表达结构化数据时语义不明确
- ◆ 结构化数据是指有明确属性，明确表示规则的数据

# 列表存储数据的问题

[ '王峰' , '男' , '1997-10-20' , 'A' , '销售' ,1000,100 ]

[ '李玲' , '女' , '2000-10-20' , 'B' , '客服' ,800 , 200 ]



# 什么是字典

- ◆ 字典(Dictionary)是Python中的内置数据结构
- ◆ 字典非常适合表达结构化数据

# 什么是字典

```
{  
    '姓名': '王峰', '性别': '男',  
    '绩效评级': 'A', '岗位': '销售',  
    '工资': 1000, '话费补贴': 100  
}
```

# 字典的特点

- ◆ 字典采用**键(key):值(value)**形式表达数据
- ◆ 字典中key不允许重复,value允许重复
- ◆ 字典是可修改的,运行时动态调整存储空间

# 创建字典的两种方式

◆ 使用`{}`创建字典

◆ 使用`dict`函数创建字典

# 字典的取值操作



# 字典更新与删除操作



# 字典的常用操作

- ◆ 为字典设置默认值
- ◆ 字典的视图
- ◆ 字典的格式化输出

# 散列值(Hash)

- ◆ 字典也称为“ 哈希(Hash)” ,对应“ 散列值”
- ◆ 散列值是从任何一种数据中创建数字“ 指纹”
- ◆ Python中提供了**hash()**函数生成散列值



# 字典的存储原理

键(Key)
name
sex
age
salary

转换

hash()
837181093
983801213
327198330
377193810

存储

内存地址	值(Value)
837181093	张三
377193810	1500
327198330	23
983801213	男

# 字典在项目中的应用



## 第五课 Python中可变类型的数据集合

# 课程总结

- ◆ 列表有序存储数据,按索引值进行提取
- ◆ 字典采用键值方式存储数据,数据无序存储

# 列表(List)

- ◆ 列表中的数据按顺序排列
- ◆ 列表有正序与倒序两种索引
- ◆ 列表可存储任意类型数据，且允许重复

# 列表(List)

-7	-6	-5	-4	-3	-2	-1
a	b	c	1	2	3	4
0	1	2	3	4	5	6

# for..in语句

- ◆ for..in语句专门用于遍历列表、元组等数据结构

for 迭代变量 in 可迭代对象

循环体(必须缩进)

# 冒泡排序

原始值

28	32	14	12	53	42
----	----	----	----	----	----

28	32	14	12	53	42
28	32	14	12	53	42
28	14	32	12	53	42
28	14	12	32	53	42
28	14	12	32	53	42
28	14	12	32	53	42

第一次循环

28	14	12	32	42	53
14	28	12	32	42	53
14	12	28	32	42	53
14	12	28	32	42	53
14	12	28	32	42	53
14	12	28	32	42	53

第二次循环

最终结果

12	14	28	32	42	53
----	----	----	----	----	----



# 什么是字典

- ◆ 字典(Dictionary)是Python中的内置数据结构
- ◆ 字典非常适合表达结构化数据

# 什么是字典

```
{  
    '姓名': '王峰', '性别': '男',  
    '绩效评级': 'A', '岗位': '销售',  
    '工资': 1000, '话费补贴': 100  
}
```

# 字典的特点

- ◆ 字典采用**键(key):值(value)**形式表达数据
- ◆ 字典中key不允许重复,value允许重复
- ◆ 字典是可修改的,运行时动态调整存储空间

# 创建字典的两种方式

◆ 使用`{}`创建字典

◆ 使用`dict`函数创建字典

# 散列值(Hash)

- ◆ 字典也称为“ 哈希(Hash)” ,对应“ 散列值”
- ◆ 散列值是从任何一种数据中创建数字“ 指纹”
- ◆ Python中提供了**hash()**函数生成散列值

# 字典的存储原理

键(Key)
name
sex
age
salary

转换

hash()
837181093
983801213
327198330
377193810

存储

内存地址	值(Value)
837181093	张三
377193810	1500
327198330	23
983801213	男