



学习目标

1. 掌握集合的使用;
2. 循环遍历字典;
3. 掌握从字典中读取键和值的方法;
4. 掌握使用 for 语句遍历字符串;
5. 了解字符串的常用方法;
6. 了解字符串切片;

传智播客-黑马程序员



目录

第1章 变量进阶-----集合.....	3
第2章 变量进阶-----字典.....	4
第3章 变量进阶-----字符串.....	6
第4章 变量进阶-----字符串的切片.....	9
第5章 变量进阶-----多维列表.....	12

传智播客-黑马程序员



第 1 章 变量进阶-----集合

- 集合用{}定义

- 创建空集合 变量名 = set()

```
# 定义一个空集合 set1  
set1 = set()
```

- -----不能通过 变量名 = {}来创建空集合;

- 集合和列表的区别:

- 列表是有序的对象集合 ;
- 集合是无序的对象集合 ;
- 同一个集合内值不允许重复。

```
# 定义一个集合 set1  
set1 = {"张飞", 33, 1.75}
```

一、集合常用操作

分类	方法	说明
增加	add(值)	增加值
删除	pop()	删除最后一个值
	remove(值)	删除指定值
	clear()	清空集合

1. 课堂练习---

```
# 定义一个空集合变量，通过 input 函数，向集合里输入任意 5 个整数  
  
# 显示集合中的最小值
```



二、循环遍历集合

遍历就是依次从集合中获取所有值

```
set1 = {"张飞", 33, 1.75}
for n in set1:
    print(n)
```

2. 课堂练习---

```
# 定义一个空集合变量，通过input函数，向集合里输入任意3个字符串
# 遍历集合，显示集合中所有的字符串
```

第2章 变量进阶-----字典

- dictionary（字典）通常用于存储“键值对”数据，键与值之间用冒号分隔。
 - 键 key 是索引，同一个字典内，键名不能重复；
 - value 是数据；
- 字典用{}定义；
 - 通过变量名 = {}创建一个空字典

```
# 定义一个空字典
dict1 = {}
```

- 字典与集合的区别：
 - 集合中只有值；
 - 字典是包含键与值的键值对。

```
# 定义一个字典，包含三个键值对
```



```
dict1 = {"name": "刘备", "age": 32, "height": 1.75}
```

一、字典常用操作

分类	方法	说明
增加/修改	字典["键"] = 值	如果键存在，代表修改已有键的值 如果键不存在，代表新增键值对
删除	pop("键")	删除指定键
	clear()	清空字典
得到值	字典["键"]	返回指定键的值

1. 课堂练习---

```
dict1 = {"name": "周瑜", "age": 32, "id": "001"}  
# 字典中增加一个键值对 'sex': '男' ;  
  
# 删除键 'id' ;  
  
# 将键 'age' 的值修改为 26。
```

二、循环遍历字典一

遍历就是依次从字典中获取所有键值对

```
# 定义一个字典 dict1，包含三个键值对  
dict1 = {"name": "刘备", "age": 32, "height": 1.75}  
# n 为键，通过 dict1[n] 可以得到键对应的值  
for n in dict1:  
    print("键=%s, 值=%s" % (n, str(dict1[n])))
```

三、循环遍历字典二---拆包方式

- 字典的 items 方法获取字典中的键值对；



- items 方法返回包含字典键值对的元组。

```
# 定义一个字典 dict1，包含三个键值对
dict1 = {"name": "刘备", "age": 32, "height": 1.75}
# n 获取 dict1 中的每个键值对
for n in dict1.items():
    print(n)
```

- 通过对元组拆包的方式获取键和值

```
# 定义一个字典 dict1，包含三个键值对
dict1 = {"name": "刘备", "age": 32, "height": 1.75}
# a 为键，b 为值
for a, b in dict1.items():
    print("键=%s · 值=%s" % (a, str(b)))
```

1. 课堂练习---

```
dict1 = {"a": 23, "b": 4, "c": 9, "d": 3, "e": 12}
# 循环遍历字典，显示字典每个键和键对应的值
```

第 3 章 变量进阶-----字符串

字符串就是一串字符，是编程语言中表示文本的数据类型。

在 Python 中可以使用一对双引号 " 或者一对单引号 ' 定义一个字符串。

虽然可以使用 \" 或者 \' 做字符串的转义，但是在实际开发中：如果字符串内部需要使用 "，可以使用 ' 定义字符串。

如果字符串内部需要使用 '，可以使用 " 定义字符串。

可以使用[索引]获取一个字符串中指定位置的字符，索引计数从 0 开始。

一、遍历字符串中每个字符

可以使用 for 循环遍历 字符串中每一个字符

```
# 定义一个字符串 str1
str1 = "我爱 python"
for n in str1:
    print(n)
```

二、字符串的常用方法

分类	方法	说明
	字符串[索引]	得到指定索引位置的字符
	isalpha()	判断字符串是否为文字构成
	isdigit()	判断字符串是否为数字构成
	islower()	判断字符串中所有字母是否都为小写
	isupper()	判断字符串中所有字母是否都为大写
查找和替换	find("子串")	查找子串在字符串中出现的位置，找不到返回 -1
	replace("子串", "新子串")	查找子串，并用新的子串替代
	count("子串")	返回子串在字符串中出现的次数
大小写转换	upper()	将小写字母转化为大写
	lower()	将大写字母转化为小写
	swapcase()	将大小写字母反转
去除空格	lstrip()	去除左侧空格
	rstrip()	去除右侧空格
	strip()	去除左右两侧空格
拆分	split("子串")	根据子串拆分字符串，返回由拆分后字符串组成的 list

三、字符串课堂练习

1. 课堂练习---

通过 `input` 函数，输入一个字符串，判断字符串是否可以转化为整数，如果不可以转化，显示"请输入数字"

2. 课堂练习---

```
# "明日复明日 明日何其多 我生待明日 万事成蹉跎"  
  
# 去掉字符串中，中间的空格
```

四、格式化字符串

% 被称为格式化操作符，专门用于处理字符串中的格式。

包含 % 的字符串，被称为格式化字符串。

% 和不同的字符连用，不同类型的数据需要使用不同的格式化字符

格式化字符	说明
%s	字符串
%d	有符号十进制整数，%06d 表示输出 6 位整数，不足用 0 补全
%x	无符号十六进制整数，%06x 表示输出 6 位整数，不足用 0 补全
%X	无符号十六进制整数，用大写字母表达
%f	浮点数，%.2f 表示只显示小数点后两位
%%	输出%

● 语法格式

```
str1 = "姓名：%s，年龄%d" % ("鲁肃", 31)
```




1. 课堂练习---

```
id = 1
name = "刘备"
weight = 80.2
tel = "13912345678"

# 以上变量，输出结果如下
```

```
*****
编号000001
姓名:刘备
体重:80.200
电话:13912345678
*****
```

第 4 章 变量进阶-----字符串的切片

切片方法适用于字符串、列表、元组；

切片使用索引值来限定范围，从一个大的字符串中切出小的字符串；

字符串、列表和元组都是有序的集合，都能够通过[索引]获取到对应的数据。

一、切片语法

字符串[开始索引:结束索引:步长]

注意：

1. 指定的区间属于左闭右开型 [开始索引, 结束索引) => 开始索引 >= 范围

< 结束索引

从起始位开始，到结束位的前一位结束（不包含结束位本身）

2. 从头开始，开始索引数字可以省略，冒号不能省略

3. 到末尾结束，结束索引数字可以省略，冒号不能省略

4. 步长默认为 1，如果连续切片，数字和冒号都可以省略

- 索引的顺序和倒序

在 Python 中不仅支持顺序索引，同时还支持倒序索引

所谓倒序索引就是从右向左计算索引

- 最右边的索引值是 -1，依次递减

```
# 定义一个字符串 str1
str1 = "我爱python"
s = str1[-1]
print(s)
```

二、切片演练

1. 截取从 2~5 位置的字符串

```
# 定义一个字符串 str1
str1 = "我爱python"
s = str1[2:6]
print(s)
```

2. 截取从 2~ 末尾的字符串

```
# 定义一个字符串 str1
str1 = "我爱python"
```



```
s = str1[2:]  
print(s)
```

3. 截取从开始 ~ 5 位置的字符串

```
# 定义一个字符串 str1  
str1 = "我爱 python"  
s = str1[:6]  
print(s)
```

4. 截取完整的字符串

```
# 定义一个字符串 str1  
str1 = "我爱 python"  
s = str1[:]  
print(s)
```

5. 从开始位置，每隔一个字符截取字符串

```
# 定义一个字符串 str1  
str1 = "我爱 python"  
s = str1[::2]  
print(s)
```

6. 从索引 1 开始，每隔一个取一个

```
# 定义一个字符串 str1  
str1 = "我爱 python"  
s = str1[1::2]  
print(s)
```

7. 截取从 2 到末尾 -1 的字符串

```
# 定义一个字符串 str1  
str1 = "我爱 python"  
s = str1[2:-1]  
print(s)
```

8. 截取字符串末尾两个字符



```
# 定义一个字符串 str1
str1 = "我爱python"
s = str1[-2:]
print(s)
```

9. 字符串的逆序（面试题）

```
# 定义一个字符串 str1
str1 = "我爱python"
s = str1[::-1]
print(s)
```

第5章 变量进阶-----多维列表

一、二维列表

列表中的数据又是另一个列表，有如下列表结构：

刘备	30
张飞	20

用列表存放以上二维表格

```
# 定义一个二维列表
list1 = [['刘备', 30],
         ['张飞', 20]]
# list1[0][0]代表第一行第一列的值
# list1[1][1]代表第二行第二列的值
```

二、多维列表

python 中不但可以有二维列表，还可以有三维等多维列表，此处不再展开赘述。



黑马程序员
www.itheima.com

传智播客旗下
高端IT教育品牌

改变中国IT教育，我们正在行动

传智播客-黑马程序员