第 5 章

Python 数据结构

学习目标

- 掌握字符串的格式化、索引和分片的具体方法。
- 掌握 Python 中字符串的基本运算、函数和方法。
- 掌握列表的创建和基本操作方法。
- 了解列表的其他操作方法。
- 掌握元组、字典、集合的创建和操作方法。
- 了解迭代器和生成器、可变对象和不可变对象。

5.1 组合数据类型简介

Python 可以处理的数据类型已经在第 2 章介绍过，如整型、字符串类型和逻辑型等，这些数据类型只能表示一个单一的数据，称为基本数据类型。但是在处理多个有关联的数据时，仅使用基本数据类型是不够的。除了这些简单数据类型外，Python 语言还可以处理一些复杂的数据类型，称为组合数据类型。

组合数据类型能够将多个基本数据类型或组合数据类型组织起来，作用是能够更清晰地反映数据之间的关系，也使得人们管理和操作数据更加方便。Python 中组合数据类型有 3 类：序列类型、映射类型和集合类型。

序列类型是一维元素向量，元素之间存在先后关系，通过序号访问。Python 中的序列类型主要有字符串类型(str)、列表类型(list)和元组类型(tuple)等。只要是序列类型，就可以使用相同的索引体系访问其中的序列元素，索引体系有两种不同的形式，分别是正向递增索引和反向递减索引，如图 5.1 所示。

映射类型是一种键值对，一个键只能对应一个值，但是多个键可以对应相同的值，通过键可以访问值。字典类型(dict)是 Python 中唯一的映射类型。字典中的元素没有特定

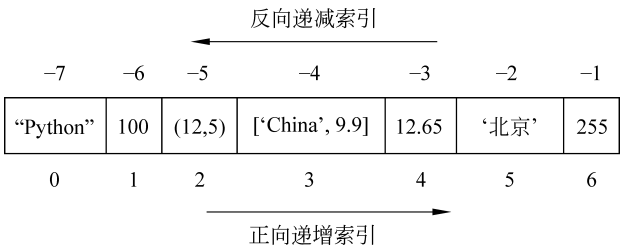

图 5.1　序列类型数据的索引体系

的顺序，每个值都对应一个唯一的键。字典类型的数据和序列类型的数据的区别在于存储和访问方式。另外，序列类型只用整数作为序号，而映射类型可以用整数、字符串或者其他类型的数据作为键，而且键和值有一定关联性，也就是键可以映射到值。

集合类型是通过数学中的集合概念引进的，是一种无序不重复的元素集。集合中包含的元素类型只能是固定的数据类型，例如整型、字符串、元组等，而列表、字典等是可变数据类型，不能作为集合中的数据元素。Python 数据类型的分类如图 5.2 所示。

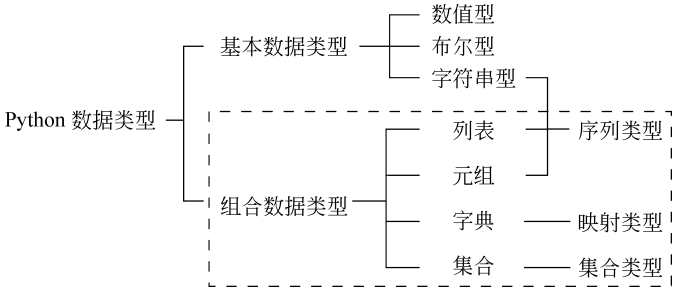

图 5.2　Python 数据类型的分类

5.2　字符串的基本操作

字符串类型与组合数据类型中的列表和字典都属于序列类型，具有很多相似的特点，因此在学习列表和元组之前，先介绍一些关于字符串类型的相关操作。字符串是一种非常重要且应用非常广泛的数据类型，支持丰富的操作和运算。Python 的字符串可以看作一串连续存储的字符的序列，可以通过索引进行顺序访问，属于序列类型。同时，由于字符串类型的单一字符串只表达一个含义，因此被看作是基本数据类型。

5.2.1　字符串的索引与分片

1. 索引

序列类型的索引体系是相同的，因此，字符串的索引也包括正向递增和反向递减两个体系。字符串中的字符按位置进行编号，使用时可以通过编号访问字符串中的特定字符。对于一个长度为 L 的字符串，其第一个字符的编号为 0，最后一个字符编号为 L－1。例如，可以通过下面的方式访问指定字符：

```
>>>str = "God Wants To Check The Air Quality"
>>>str[0],str[1],str[19]
('G', 'o', 'T')
```

Python 同时允许根据索引反向访问字符串，此时字符串的编号从－1 开始。例如：

```
>>>str = "God Wants To Check The Air Quality"
>>>str[-1],str[-13],str[-26]
('y', 'e', 's')
>>>
```

2. 分片

字符串的分片是指通过索引对字符串进行切片的操作。分片操作格式：

```
<字符串名>[i:j:k]
```

这里的 i 表示起始编号，j 表示结束编号，k 表示编号增加步长。注意，切片的位置不包含 j 位置上的字符。例如：

```
>>>str = "God Wants To Check The Air Quality"
>>>str[0:8:2]            #将 str 字符串第 0、2、4、6 的位置进行切片
'GdWn'
```

分片语句中的 i、j、k 均可以省略。i 省略时，表示从 0 或－1 开始；j 省略时，表示到最后一个字符；k 省略时，表示步长为 1。例如：

```
>>>str = "God Wants To Check The Air Quality"
>>>str[4:18]
```

```
'Wants To Check'
>>>str[::2]
'GdWnsT hc h i ult'
>>>str[27::]
'Quality
```

【例 5.1】 利用字符串分片操作，逆序输出字符串。

程序代码如下：

```
#Example5.1
st = input('输入一个字符串：')
print("原字符串：%s"%(st))
print("逆序字符：%s"%(st[-1::-1]))
```

程序的运行结果如下：

```
>>>
================ RESTART:C:/Users/Python/Example5.1.py ================
输入一个字符串：abcdefg
原字符串：abcdefg
逆序字符：gfedcba
>>>
```

说明：

（1）程序用来实现将输入的字符串逆序输出；

（2）通过分片生成逆序字符串，st[－1::－1]表示从末尾进行分片直到完成，步长为－1；

（3）st[－1::－1]也可以写作 st[::－1]。

【例 5.2】 查询月份英文缩写。

程序代码如下：

```
#Example5.2
#查询英文月份
st = '''一月 Jan 二月 Feb 三月 Mar 四月 Apr 五月 May 六月 Jun
七月 Jul 八月 Aug 九月 Sep 十月 Oct 十一 Nov 十二 Dec'''
mon = int(input('input a month:'))
n = (mon-1) * 5
```

```
print('%s 的英文简称为：%s'%(st[n:n+2],st[n+2:n+5]))
```

程序的运行结果如下：

```
>>>
================ RESTART:C:/Users/Python/Example5.2.py ================
input a month:6
六月的英文简称为：Jun
>>>
```

说明：

(1) 程序用来实现在字符串 st 中分片生成月份的英文缩写；

(2) st 中用固定格式顺序存放 12 个月的英文缩写形式；

(3) 为方便定位，将每个月份的信息固定长度为 5(三个英文两个汉字)；

(4) 通过表达式 n=(mon−1) * 5 的计算，得到相应月份所在的起始位置；

(5) 分片 st[n:n+2]生成月份字符串，st[n+2:n+5]生成英文缩写字符串。

5.2.2　字符串的基本运算

Python 支持通过一些运算符实现几个字符串的基本运算，包括字符串的连接、判断子串、字符串的比较等。字符串的基本运算如表 5.1 所示。

表 5.1　字符串基本运算及功能

运　　算	功　　能
s1+s2	连接字符串 s1 和 s2
s1 * n	生成由 n 个 s1 组成的字符串
s1 in s2	如果 s1 是 s2 的子串，返回 True，否则返回 False
>，<，==	比较字符串的 AscII 码，多个字符时从左向右依次比较

字符串的运算实例：

```
>>>s1 = 'Python 程序设计'
>>>s2 = '入门'
>>>n = 2
>>>s1 + s2
'Python 程序设计入门'
```

```
>>> s1 * n
'Python 程序设计 Python 程序设计'
>>> s2 in s1
False
>>> 'a' > 'A'                          #比较单个字符的 AscII 码值,a 为 97,A 为 65
True
>>> 'this is a test' > 'this is '      #多个字符的比较,从左向右依次比较每个字符
True
>>>
```

5.2.3 字符串运算方法

Python 是面向对象的程序设计语言，Python 中的所有数据类型都是一个类。类的方法就是封闭在类中的函数，与内置的功能相似，但是调用方法不同。字符串对象通过一些常用方法完成相应的运算。调用字符串方法的格式如下：

```
<字符串名>.<方法名>(<参数>)
```

字符串常用方法如表 5.2 所示。

表 5.2 字符串常用方法及功能

方　法	功　能
s.lower()	将字符串转换为小写字母
s.upper()	将字符串转换为大写字母
s.capitalize()	将字符串转换为首字母大写
s.tiltle()	将第一个字符转换为大写
s.replace(old,new[,count])	将 s 中的 old 字符串替换为 new，count 为替换的次数
s.split([sep,[maxsplit]])	以 sep 为分隔符将 s 拆分为一个列表，默认分隔符为空格，maxsplit 表示拆分的次数，默认为－1，表示无限制
s.find(s1[,start,[end]])	返回 s1 在 s 中出现的位置。如果没有出现返回－1
s.count(s1[,start,[end]]	返回 s1 在 s 中出现的次数
s.isalnum()	判断 s 是否为全字母和数字，且至少一个字符
s.isalpha()	判断 s 是否为全字母，且至少一个字符

续表

方　法	功　能
s.isupper()	判断 s 是否为全大写字母
s.islower()	判断 s 是否为全小写字母
s.format()	字符串 s 的一种格式化输出格式，5.2.5 节详细介绍
s.join(iterable)	以 s 为分隔符，将可迭代对象 iterable(字符串、列表、元组、字典、集合)的每个元素连接，生成一个新的字符串

字符串大小写转换方法的实例：

```
>>>s = 'this is a test!'
>>>s.upper()
'THIS IS A TEST!'
>>>s.capitalize ()
'This is a test!'
>>>s.title()
'This Is A Test!'
>>>
```

字符串拆分方法的实例：

```
>>>s = 'this is a test!'
>>>s.split()                          #将字符串以空格为分隔符拆分
['this', 'is', 'a', 'test!']
>>>s.split(sep = ' ',maxsplit=1)      #将字符串以空格为分隔符拆分一次
['this', 'is a test!']
>>>
```

字符串替换和查找方法的实例：

```
>>>s = 'this is a test!'
>>>s.find('t')                        #查找字母 t 在字符串 s 中第一次出现的位置
0
>>>s.count('t')                       #查找字母 t 在字符串 s 中出现的次数
3
>>>s.replace('t','T',2)               #将字符串的 t 替换 T，替换两次
```

```
'This is a Test!'
>>>
```

字符串连接方法实例：

```
>>>" ".join('Python')
'P y t h o n'
>>>",".join('Python')
'P,y,t,h,o,n'
>>>",".join(['Python','Java','php','Pascal'])
'Python,Java,php,Pascal'
>>>",".join(('Python','Java','php','Pascal'))
'Python,Java,php,Pascal'
>>>",".join({'Python':98,'Java':80,'php':79,'Pascal':92})
'Python,Java,php,Pascal'
```

【例 5.3】 统计英文语句中某个单词出现的次数。

程序代码如下：

```
#Example5.3
passage = 'Do not trouble trouble till trouble troubles you.'
word = input('input a word:')
n = passage.count(word)
print('%s 出现的次数：%s'%(word,n))
```

程序的运行结果如下：

```
>>>
================ RESTART:C:\Users\Python \Example5.3.py ================
input a word:trouble
trouble 出现的次数：4
>>>
```

【例 5.4】 电文加密程序。

程序代码如下：

```
#Example5.4
#电文加密
```

```
original = input('输入原文：')
cryption = ''
for s1 in original:
      if s1.isalpha():
            i = ord(s1) + 5
            if s1.isupper():
                  ifi > ord('Z'):i -= 26
            else:
                  ifi > ord('z'):i -= 26
            s2 = chr(i)
            cryption += s2
      else:
            cryption+= s1
print('输出密文：%s'%(cryption))
```

程序的运行结果如下：

```
>>>
================ RESTART:C:/Users/Python/Example5.4.py =================
输入原文：Windows!
输出密文：Bnsitbx!
>>>
```

说明如下。

(1) 电文加密规则：将原文中的字母转换为英文字母表中其后面第 5 个字母，例如，A→F。要求保持原文的大小写状态，且除字母外的其他字符不变，不进行加密处理。

(2) 代码中 s1.isalpha()字符串方法判断 s1 是否为字母，若是，返回 True；否则返回 False。

(3) 代码中 s1.isupper()字符串方法判断 s1 是否为大写字母，若是，返回 True；否则返回 False。

(4) 字母在 ASCII 表中分别按 a～z 和 A～Z 的顺序排列，即 26 个字母的 ASCII 码是连续的。a～z 的 ASCII 码是 97～122，A～Z 的 ASCII 码是 65～90。

5.2.4　字符串的格式化

字符串的格式化通常用在 print()函数中，用来实现将字符以特定的样式输出。Python 支持多种字符串的格式化输出方法。在 Python 2.5 以及更低的版本中，只支持使

用字符串格式化输出方法%s。从 Python 3.0 版本开始，可使用 format()方法进行格式化输出，同时兼容%s 形式。Python 3.6 版本中引入了一种新的字符串格式化方式 f-string。

1. 转义字符

在 Python 字符串中可以使用转义字符，即在反斜杠(\)后面紧跟一个字符来表示一些特殊字符，常用的转义字符如表 5.3 所示。

表 5.3 常用的转义字符

转义字符	描　述	转义字符	描　述
\(在行尾时)	续行符	\n	换行
\\	反斜杠符号\	\r	回车
\'	单引号	\t	横向制表符
\"	双引号	\v	纵向制表符

转义字符的使用实例如下：

```
>>>print('It's a book')                          #单引号嵌套错误

SyntaxError: invalid syntax
>>>print('It\'s a book')                         #\'为转义字符，输出时转义为'
It's a book
>>>print("It\'s a book")                         #双引号内的单引号正常输出
It's a book
>>>print("你好!\n 欢迎来到 Python 的世界!")
你好!
欢迎来到 Python 的世界!
```

表 5.3 中，转义字符“\n”“\r”与平常意义上的回车理解不同。“\n”代表换行，换到当前位置的下一行；“\r”代表回车，回到当前行的行首，而不会换到下一行，如果接着输出，本行以前的内容会被逐一覆盖。

【例 5.5】 倒计时程序。

程序代码如下：

```
#Example5.5
```