

目录

第 1 章 入门准备

01 开篇词：你为什么要学 Python ?

02 我会怎样带你学 Python ?

03 让 Python 在你的电脑上安家落户

04 如何运行 Python 代码 ?

第 2 章 通用语言特性

05 数据的名字和种类—变量和类型

06 一串数据怎么存—列表和字符串 [最近阅读](#)

07 不只有一条路—分支和循环

08 将代码放进盒子—函数

09 知错能改—错误处理、异常机制

10 定制一个模子—类

11 更大的代码盒子—模块和包

12 练习—密码生成器

第 3 章 Python 进阶语言特性

13 这么多的数据结构（一）：列表、元祖、字符串

14 这么多的数据结构（二）：字典、集合

15 Python大法初体验：内置函数

16 深入理解下迭代器和生成器

17 生成器表达式和列表生成式

18 把盒子升级为豪宅：函数进阶

19 让你的模子更好用：类进阶

20 从小独栋升级为别墅区：函数式编程

# 06 一串数据怎么存—列表和字符串

更新时间：2019-08-26 10:09:38



“

知识犹如人体的血液一样宝贵。

——高士其

”

上一节中讲了数据类型，有一个问题，之前所介绍的数据类型大多是用来表示单个数据的。比如整数型，一个整数型的变量只能保存一个整数。又如布尔型，一个布尔型的变量只能保存一个布尔值。浮点型和 None 型也是如此。要是此刻有一系列的数据，那该怎么在程序里保存和使用呢？

举个例子：当我的只有一个电话号码的时候，我可以使用整数型来表示，并保存在变量里：

```
tel = 13011110000
```

但如果如果有十个电话号码，该怎么来表示和使用它们呢？

```
13011110000
18022221111
13433332222
13344443333
17855554444
13866665555
15177776666
13388887777
18799998888
17800009999
```

← 慕课专栏	☰ 你的第一本Python基础入门书 / 06 一串数据怎么存—列表和字符串
目录	
第 1 章 入门准备	
01 开篇词：你为什么要学 Python ？	
02 我会怎样带你学 Python ？	
03 让 Python 在你的电脑上安家落户	
04 如何运行 Python 代码？	
第 2 章 通用语言特性	
05 数据的名字和种类—变量和类型	
06 一串数据怎么存—列表和字符串 <small>最近阅读</small>	<pre>tel01 = 13011110000 tel02 = 18022221111 ... tel10 = 17800009999</pre>
07 不只有一条路—分支和循环	或者「把它们用逗号拼在一起然后放到字符串里」：
08 将代码放进盒子—函数	<pre>tels = '13011110000,18022221111,13433332222,13344443333,17855554444,138666655</pre>
09 知错能改—错误处理、异常机制	是的，看起来这似乎能解决问题。但是这两种办法的弊端也很明显。第一种使用多个变量的方式，在数据量很大的情况下使用起来会十分繁琐；第二种使用字符串的方式，如果我们需要对其中的某些数据做处理，那这种方式就很不方便灵活了。
10 定制一个模子—类	这时我们可以选择使用列表。
11 更大的代码盒子—模块和包	列表（List）
12 练习—密码生成器	列表是一种用于保存批量数据的数据类型。它和整数型、布尔型等数据类型一样都被内置在 Python 中。
第 3 章 Python 进阶语言特性	列表的写法
13 这么多的数据结构（一）：列表、元组、字符串	列表的写法为 <code>[ 数据项1, 数据项2, ..., 数据项N ]</code> ，方括号就代表列表，每个数据项放在方括号中并用逗号分隔。
14 这么多的数据结构（二）：字典、集合	如之前的那一串电话号码可以这样来保存：
15 Python大法初体验：内置函数	<pre>tels = [13011110000, 18022221111, 13433332222, 13344443333, 17855554444, 138666</pre>
16 深入理解下迭代器和生成器	
17 生成器表达式和列表生成式	扩展：为了方便阅读，我们也可以把把这个列表写成多行的形式：
18 把盒子升级为豪宅：函数进阶	<pre>tels = [     13011110000,     18022221111,     13433332222,     13344443333,     17855554444,     13866665555,     15177776666,     13388887777,     18799998888,     17800009999 ]</pre>
19 让你的模子更好用：类进阶	每个数据项一行，这样是不是更好看了！
20 从小独栋升级为别墅区：函数式编程	在解释器的交互模式中输入这样的多行代码时，我们会发现第一行的提示符是 <code>&gt;&gt;&gt;</code> ，之后每行的提示符会变成 <code>...</code> ，直到完成了多行输入则又变回 <code>&gt;&gt;&gt;</code> 。如：

<div>← 慕课专栏</div>	<div>≡ 你的第一本Python基础入门书 / 06 一串数据怎么存—列表和字符串</div>
<div>目录</div>	<div>... 18022221111, ... 13433332222 ... ] &gt;&gt;&gt;</div>
<div>第 1 章 入门准备</div>	
<div>01 开篇词：你为什么要学 Python ？</div>	
<div>02 我会怎样带你学 Python ？</div>	<div>列表中的数据可以是任意类型的。比如整数型、字符串类型和布尔类型等：</div>
<div>03 让 Python 在你的电脑上安家落户</div>	<div>[100, 'about', True]</div>
<div>04 如何运行 Python 代码？</div>	<div>列表索引</div>
<div>第 2 章 通用语言特性</div>	<div>列表中的每个数据项都是有先后次序的，最前面的数据项的位置编号为 0，之后依次是 1，2</div>
<div>05 数据的名字和种类—变量和类型</div>	<div>…… N，这个位置编号在编程中的术语叫做索引（Index）。注意 Python 中索引是从 0 开始计数的，0 即代表第一个位置。</div>
<div>06 一串数据怎么存—列表和字符串 <a href="#">最近阅读</a></div>	<div>可以通过符号 <code>[]</code> 来获取某个索引所对应的数据项。比如：</div>
<div>07 不只有一条路—分支和循环</div>	<div>&gt;&gt;&gt; fruits = [ 'apple' , 'banana' , 'cherry' , 'durian' ] &gt;&gt;&gt; fruits[0] ' apple ' &gt;&gt;&gt; fruits[2] ' cherry '</div>
<div>08 将代码放进盒子—函数</div>	
<div>09 知错能改—错误处理、异常机制</div>	<div>上面的 <code>fruits</code> 有 4 项数据，所以最大的索引是 <code>3</code>。如果我们强行要用更大的索引值去取数据会怎样呢，来试一下：</div>
<div>10 定制一个模子—类</div>	<div>&gt;&gt;&gt; fruits[4]  Traceback (most recent call last):   File " ", line 1, in IndexError: list index out of range</div>
<div>11 更大的代码盒子—模块和包</div>	<div>可以看到代码直接就报错了，具体信息为「list index out of range」，列表索引超出范围。</div>
<div>12 练习—密码生成器</div>	<div>扩展：这是 Python 的典型报错形式，这里有三行内容（也可能会有很多行），前两行是错误定位，描述出错的位置（如某文件的某行），后面是错误描述，指出这是个 <code>IndexError</code> 错误，具体信息为「list index out of range」。</div>
<div>第 3 章 Python 进阶语言特性</div>	<div>若大家在写代码时遇到错误，可以按照这种方法尝试自己分析错误信息。</div>
<div>13 这么多的数据结构（一）：列表、元祖、字符串</div>	<div>除了通过索引去获取值，也可以通过索引去改变列表中某项数据的值。通过赋值的方式来实现：</div>
<div>14 这么多的数据结构（二）：字典、集合</div>	<div>fruits[0] = 'pear'</div>
<div>15 Python大法初体验：内置函数</div>	
<div>16 深入理解下迭代器和生成器</div>	
<div>17 生成器表达式和列表生成式</div>	
<div>18 把盒子升级为豪宅：函数进阶</div>	
<div>19 让你的模子更好用：类进阶</div>	
<div>20 从小独栋升级为别墅区：函数式编程</div>	

<div>← 慕课专栏</div> <div>≡ 你的第一本Python基础入门书 / 06 一串数据怎么存—列表和字符串</div>	
目录	<pre>&gt;&gt;&gt; fruits[0] 'apple'  &gt;&gt;&gt; fruits[0] = 'pear' &gt;&gt;&gt; fruits[0] 'pear'</pre>
第 1 章 入门准备	
01 开篇词：你为什么要学 Python ？	
02 我会怎样带你学 Python ？	
03 让 Python 在你的电脑上安家落户	
04 如何运行 Python 代码？	
第 2 章 通用语言特性	列表的长度
05 数据的名字和种类—变量和类型	列表中数据项的个数，叫做列表（的）长度。
06 一串数据怎么存—列表和字符串 <small>最近阅读</small>	想要获得列表的长度可以使用 <code>len()</code> 这个东西。像这样：
07 不只有一条路—分支和循环	<pre>len(fruits)</pre>
08 将代码放进盒子—函数	<pre>&gt;&gt;&gt; len(fruits) 4</pre>
09 知错能改—错误处理、异常机制	<pre>&gt;&gt;&gt; len([1, 2, 3, 4, 5, 6, 7]) 7</pre>
10 定制一个模子—类	说明： <code>len()</code> 是 Python 中的内置函数。函数的概念会在之后的章节中介绍。
11 更大的代码盒子—模块和包	
12 练习—密码生成器	
第 3 章 Python 进阶语言特性	向列表添加数据
13 这么多的数据结构（一）：列表、元祖、字符串	之前使用时，列表中的数据在一开始就已经被确定下来了，并一直保持着这个长度。但在很多时候，我们需要随时向列表中添加数据。
14 这么多的数据结构（二）：字典、集合	向列表的末尾添加数据可以用 <code>.append()</code> 这个东西，它的写法是：
15 Python大法初体验：内置函数	<pre>列表.append(新数据)</pre>
16 深入理解下迭代器和生成器	看一个示例。这里首先创建了一个空的列表，将其变量命名为 <code>fruits</code> ，然后通过 <code>.append()</code> 向其中添加内容。
17 生成器表达式和列表生成式	<pre>&gt;&gt;&gt; fruits = [] &gt;&gt;&gt; fruits []  &gt;&gt;&gt; fruits.append( 'pear' ) &gt;&gt;&gt; fruits</pre>
18 把盒子升级为豪宅：函数进阶	
19 让你的模子更好用：类进阶	
20 从小独栋升级为别墅区：函数式编程	

<div>← 慕课专栏</div>	<div>≡ 你的第一本Python基础入门书 / 06 一串数据怎么存—列表和字符串</div>
<div>目录</div>	<div>&gt;&gt;&gt; fruits.append('lemon')  &gt;&gt;&gt; fruits [ 'pear', 'lemon' ]</div>
<div>第 1 章 入门准备</div>	
<div>01 开篇词：你为什么要学 Python？</div>	
<div>02 我会怎样带你学 Python？</div>	
<div>03 让 Python 在你的电脑上安家落户</div>	
<div>04 如何运行 Python 代码？</div>	<div>扩展：append() 是列表的方法。「方法」具体是什么我们在之后的面向对象章节中介绍。这里暂且把方法理解为某个数据类型自带的功能，如 append() 是列表自带的功能。</div>
<div>第 2 章 通用语言特性</div>	<div>字符串（String）</div>
<div>05 数据的名字和种类—变量和类型</div>	<div>字符串也可以保存批量数据，只不过其中的数据项只能是字符。</div>
<div>06 一串数据怎么存—列表和字符串 <span>最近阅读</span></div>	<div>我们在前一个章节中介绍过字符串，字符串是用来表示文本的数据类型。字符串以单引号或双引号以及包裹在其中的若干字符组成，如：</div>
<div>07 不只有一条路—分支和循环</div>	<div>'good good study' '100' '江畔何人初见月，江月何年初照人'</div>
<div>08 将代码放进盒子—函数</div>	<div>字符串索引</div>
<div>09 知错能改—错误处理、异常机制</div>	<div>从形式上我们不难看出，字符串中的字符也是有先后次序的。字符串是字符的有序序列，所以也具有索引。也可以根据索引取出其中某一个字符。其索引使用方式和列表相同：</div>
<div>10 定制一个模子—类</div>	<div>'good good study'[3]</div>
<div>11 更大的代码盒子—模块和包</div>	
<div>12 练习—密码生成器</div>	
<div>第 3 章 Python 进阶语言特性</div>	<div>&gt;&gt;&gt; 'good good study'[3]  'd'</div>
<div>13 这么多的数据结构（一）：列表、元组、字符串</div>	<div>也可以先把字符串保存在变量里，然后在变量上使用索引。结果是一样的：</div>
<div>14 这么多的数据结构（二）：字典、集合</div>	<div>words = 'good good study' words[3]</div>
<div>15 Python大法初体验：内置函数</div>	
<div>16 深入理解下迭代器和生成器</div>	
<div>17 生成器表达式和列表生成式</div>	<div>&gt;&gt;&gt; words = 'good good study'  &gt;&gt;&gt; words[3]  'd'</div>
<div>18 把盒子升级为豪宅：函数进阶</div>	<div>有一点需要注意，字符串不能像列表那样通过索引去改变数据项的值。因为字符串类型的值是不可变的（Immutable），我们不能在原地修改它其中的某个字符。</div>
<div>19 让你的模子更好用：类进阶</div>	
<div>20 从小独栋升级为别墅区：函数式编程</div>	

<div>← 慕课专栏</div> <div>三 你的第一本Python基础入门书 / 06 一串数据怎么存—列表和字符串</div>	
目录	
第 1 章 入门准备	
01 开篇词：你为什么要学 Python ？	
02 我会怎样带你学 Python ？	
03 让 Python 在你的电脑上安家落户	
04 如何运行 Python 代码 ？	
第 2 章 通用语言特性	
05 数据的名字和种类—变量和类型	
06 一串数据怎么存—列表和字符串 <a href="#">最近阅读</a>	<pre>&gt;&gt;&gt; words[3] = 'b'  Traceback (most recent call last):    File “ ”, line 1, in  TypeError: ‘str’ object does not support item assignment</pre> <p>上面报出一个 <code>TypeError</code> 错误，具体信息为「<code>‘str’ object does not support item assignment</code>」，其中「<code>‘str’ object</code>」指的就是字符串，它不支持直接为其中某一个项（字符）赋值。</p> <p><b>字符串长度</b></p> <p>字符串中字符的个数也就是<b>字符串的长度</b>（包括空格在内的所有空白符号）。</p> <p>获取字符串长度的方式和列表一样，也是使用 <code>len()</code>：</p> <pre>len('good good study')</pre> <pre>&gt;&gt;&gt; len( 'good good study' )  15</pre>
07 不只有一条路—分支和循环	
08 将代码放进盒子—函数	
09 知错能改—错误处理、异常机制	
10 定制一个模子—类	
11 更大的代码盒子—模块和包	
12 练习—密码生成器	
第 3 章 Python 进阶语言特性	
13 这么多的数据结构（一）：列表、元祖、字符串	
14 这么多的数据结构（二）：字典、集合	
15 Python大法初体验：内置函数	
16 深入理解下迭代器和生成器	
17 生成器表达式和列表生成式	
18 把盒子升级为豪宅：函数进阶	
19 让你的模子更好用：类进阶	
20 从小独栋升级为别墅区：函数式编程	<p><b>总结</b></p> <p>如果我们想要保存和表示批量数据，可以使用 Python 中的列表（List）类型。列表是有序序列，能保存任意类型的数据项，可以通过索引（Index）来获取和修改其中某一个数据项，可以通过 <code>len()</code> 函数来获取列表的长度，也可以通过 <code>.append()</code> 在列表末尾追加数据项。</p> <p>如果数据是文本，那么可以用字符串类型（String）来表示。字符串类型是字符的有序序列，可以通过索引获取某个位置的字符，也可以通过 <code>len()</code> 函数来获取长度。</p> <p>Python 中的列表和字符串还有很多功能，之后讲「数据结构」时为大家一一介绍。</p> <hr/> <p><b>多语言比较：</b></p> <p>数组是保存和表示批量数据的最基本的结构，它也是构造字符串、集合和容器的基石。</p> <p><b>Python 中没有数组概念</b>，取而代之的是列表这种更高级的数据结构，列表涵盖了数组的功能并提供了更多且更强大的功能。</p> <p>Java 中，用 <b>类型[]</b> 的写法来表示数组：</p> <pre>// 定义数组 int numbers[];  // 定义数组并用指定值初始化： int numbers[] = {1, 2, 3};</pre>

← 慕课专栏

三 你的第一本Python基础入门书 / 06 一串数据怎么存—列表和字符串

目录

第 1 章 入门准备

01 开篇词：你为什么要学 Python ？

02 我会怎样带你学 Python ？

03 让 Python 在你的电脑上安家落户

04 如何运行 Python 代码 ？

第 2 章 通用语言特性

05 数据的名字和种类—变量和类型

06 一串数据怎么存—列表和字符串 最近阅读

07 不只有一条路—分支和循环

08 将代码放进盒子—函数

09 知错能改—错误处理、异常机制

10 定制一个模子—类

11 更大的代码盒子—模块和包

12 练习—密码生成器

第 3 章 Python 进阶语言特性

13 这么多的数据结构（一）：列表、元祖、字符串

14 这么多的数据结构（二）：字典、集合

15 Python大法初体验：内置函数

16 深入理解下迭代器和生成器

17 生成器表达式和列表生成式

18 把盒子升级为豪宅：函数进阶

19 让你的模子更好用：类进阶

20 从小独栋升级为别墅区：函数式编程

```
// 定义数组
int numbers[3];

// 定义数组并用指定值初始化：
int numbers[] = {1, 2, 3};
```

Go 语言定义数组：

```
// 定义数组
var numbers [3] int

// 定义数组并用指定值初始化：
var numbers = [3]int {1, 2, 3}
```

← 05 数据的名字和种类—变量和类型

07 不只有一条路—分支和循环 →

精选留言 0

欢迎在这里发表留言，作者筛选后可公开显示

！

目前暂无任何讨论

千学不如一看，千看不如一练