

在模块 `random` 中，另一个很有用的函数是 `choice()`。它将一个列表或元组作为参数，并随机返回其中的一个元素：

```
>>> from random import choice
>>> players = ['charles', 'martina', 'michael', 'florence', 'eli']
>>> first_up = choice(players)
>>> first_up
'florence'
```

在创建与安全相关的应用程序时，不要使用模块 `random`，但它能用来创建众多有趣的项目。

注意：还可以从其他地方下载外部模块。第二部分的每个项目都需要使用外部模块，届时你将看到很多这样的示例。

动手试一试

练习 9.13：骰子 创建一个 `Die` 类，它包含一个名为 `sides` 的属性，该属性的默认值为 6。编写一个名为 `roll_die()` 的方法，它打印位于 1 和骰子面数之间的随机数。创建一个 6 面的骰子并掷 10 次。

创建一个 10 面的骰子和一个 20 面的骰子，再分别掷 10 次。

练习 9.14：彩票 创建一个列表或元素，其中包含 10 个数和 5 个字母。从这个列表或元组中随机选择 4 个数或字母，并打印一条消息，指出只要彩票上是这 4 个数或字母，就中大奖了。

练习 9.15：彩票分析 可以使用一个循环来理解中前述彩票大奖有多难。为此，创建一个名为 `my_ticket` 的列表或元组，再编写一个循环，不断地随机选择数或字母，直到中大奖为止。请打印一条消息，报告执行多少次循环才中了大奖。

练习 9.16：Python 3 Module of the Week 要了解 Python 标准库，一个很不错的资源是网站 `Python 3 Module of the Week`。请访问该网站并查看其中的目录，找一个你感兴趣的模块进行探索，从模块 `random` 开始可能是个不错的选择。

9.6 类的编程风格

你必须熟悉有些与类相关的编程风格问题，在编写的程序较复杂时尤其如此。

类名应采用大驼峰式命名法，即将类名中的每个单词的首字母都大写，并且不使用下划线。实例名和模块名都采用全小写格式，并在单词之间加上下划线。