3.1 Знакомство

*Основное внимание в этой главе уделяется представлению функций, строк и объектов и использованию их для разработки программ.*

Предыдущая глава представила фундаментальные методы программирования и научила вас, как писать простые программы для решения основных задач. В этой главе представлены функции Python для выполнения общих математических операций. О том, как создавать пользовательские функции, вы узнаете в главе 6.

Предположим, вам нужно оценить площадь, заключенную в четыре города, учитывая местоположения GPS (широту и долготу) этих городов, как показано на следующей диаграмме. Как бы вы написали программу для решения этой проблемы? Вы сможете написать такую программу после завершения этой главы.

…

Поскольку все данные в Python являются объектами, полезно вводить объекты на ранней стадии, чтобы вы могли начать использовать их для разработки полезных программ. В этой главе дается краткое введение в объекты и строки; Вы узнаете больше об объектах и строках в главах 7 и 8.

3.2 Общие функции Python

*Python предоставляет множество полезных функций для общих задач программирования.*

*Функция* — это группа операторов, которая выполняет определенную задачу. Python, как и другие языки программирования, предоставляет библиотеку функций. Вы уже использовали функции **eval**, **input**, **print** и **int**. Это встроенные функции, которые всегда доступны в интерпретаторе Python. Вам не нужно импортировать какие-либо модули, чтобы использовать эти функции. Дополнительно можно использовать встроенные функции **abs**, **max**, **min**, **pow** и **round**, как показано в таблице 3.1.

**ТАБЛИЦА 3.1** Простые встроенные функции Python

|  |  |  |
| --- | --- | --- |
| Пример | описания | функции |
| abs(x) | Возвращает абсолютное значение для x. | **abs(-2) = 2** |
| max(x1, x2, ...) | Возвращает наибольшее значение среди x1, x2, ... | **max(1, 5, 2) = 5** |
| min(x1, x2, ...) | Возвращает наименьшее значение среди x1, x2, ... | **min(1, 5, 2) = 1** |
| pow(a, b) | Возвращает ab. То же самое, что и \*\* b. | **pow(2, 3) = 8** |
| round(x) | Возвращает целое число, ближайшее к x. Если x одинаково близок к двум целым числам, возвращается четное число. | **round(5.4) = 5**  **round(5.5) = 6**  **round(4.5) = 4** |
| round(x, n) | Возвращает значение float, округленное до n цифр после десятичной запятой. | **round(5.466, 2) = 5.47**  **round(5.463, 2) = 5.46** |

Многие программы созданы для решения математических задач. модуль **math** Python предоставляет математические функции, перечисленные в таблице 3.2.

Two mathematical constants, **pi** and **e**, are also defined in the **math** module. They can be accessed using **math.pi** and **math.e**. Listing 3.1 is a program that tests some math functions. Because the program uses the math functions defined in the **math** module, the **math** module is imported in line 1.

Две математические константы, **pi** и **e**, также определены в математическом модуле. К ним можно получить доступ с помощью **math.pi** и **math.e**. Листинг 3.1 представляет собой программу, которая тестирует некоторые математические функции. Поскольку программа использует математические функции, определенные в модуле math, модуль math импортируется в строку 1.

**ТАБЛИЦА 3.2** Математические функции

…

**LISTING 3.1** MathFunctions.py

…

Математические функции можно использовать для решения многих вычислительных задач. Например, для трех вершин треугольника можно вычислить углы с помощью следующей формулы:

…

Не пугайтесь математической формулы. Как мы обсуждали в начале листинга 2.8, ComuteLoan.py, вам не нужно знать, как выводится математическая формула, чтобы написать программу для вычисления платежей по кредиту. Здесь, в этом примере, учитывая длину трех сторон, вы можете использовать эту формулу для написания программы для вычисления углов без необходимости знать, как получена формула. Чтобы вычислить длины сторон, нам нужно знать координаты трех угловых точек и вычислять расстояния между точками.

В листинге 3.2 приведен пример программы, которая предлагает пользователю ввести координаты x и y трех угловых точек треугольника, а затем отображает углы рисунка.

**LISTING 3.2** ComputeAngles.py

…

Программа предлагает пользователю ввести три точки (строка 3). Это сообщение с подсказкой неясно. Вы должны дать пользователю явные инструкции о том, как ввести эти пункты следующим образом:

input(**"Enter six coordinates of three points separated by commas**\

**like x1, y1, x2, y2, x3, y3: "**)

Программа вычисляет расстояния между точками (линии 5–7) и применяет формулу для вычисления углов (линии 9–11). Углы округляются для отображения до двух цифр после десятичной запятой (строки 13–14).

3.3 Строки и символы

*Строка (описанная в главе 1) представляет собой последовательность символов. Python обрабатывает символы и строки одинаково.*

Помимо обработки числовых значений, в Python можно обрабатывать *строки*. *Строка* представляет собой последовательность символов и может включать текст и числа. Строковые значения должны быть заключены в соответствующие *одинарные кавычки* (') или *двойные кавычки* ("). Python не имеет типа данных для символов. Односимвольная строка представляет символ. Например:

letter = **'A'** # Same as letter = "A"

numChar = **'4'** # Same as numChar = "4"

message = **"Good morning"** # Same as message = 'Good morning'

The first statement assigns a string with the character **A** to the variable **letter**. The second statement assigns a string with the digit character **4** to the variable **numChar**. The third statement assigns the string **Good morning** to the variable **message**.

Первая инструкция присваивает строку с символом **A** переменной **letter**. Второй оператор присваивает переменную **numChar** строку с символом цифры **4**. Третья инструкция присваивает строку **Good morning** переменному **message**.

**Заметка**

Для согласованности в этой книге используются двойные кавычки для строки с более чем одним символом и одинарные кавычки для строки с одним символом или пустой строкой. Это соглашение согласуется с другими языками программирования, поэтому вам будет легко преобразовать программу Python в программу, написанную на других языках.

3.3.1 ASCII Code

Компьютеры используют двоичные числа внутри (см. Раздел 1.2.2). Символ хранится в компьютере в виде последовательности 0 и 1. Сопоставление символа с его двоичным представлением называется *кодировкой символов* *(Character encoding)*. Существуют различные способы кодирования персонажа. Способ кодирования символов определяется схемой кодирования. Одним из популярных стандартов является ASCII (American Standard Code for Information Interchange), 7-битная схема кодирования для представления всех прописных и строчных букв, цифр, знаков препинания и управляющих символов. ASCII использует числа от 0 до 127 для представления символов. В приложении B, Набор символов ASCII, показан код ASCII для символов.

3.3.2 Unicode Code

Python also supports Unicode. Unicode is an encoding scheme for representing international characters. ASCII is a small subset of Unicode. Unicode was established by the Unicode Consortium to support the interchange, processing, and display of written texts in the world’s diverse languages. A Unicode starts with **\u**, followed by four hexadecimal digits that run from **\u0000** to **\uFFFF**. (For information on hexadecimal numbers, see Appendix C.) For example, the word “welcome” is translated into Chinese using two characters, and . The Unicode representations of these two characters are **\u6B22\u8FCE**.

Python также поддерживает Юникод. Юникод — это схема кодирования для представления международных символов. ASCII — это небольшое подмножество Юникода. Юникод начинается с **\u**, за которым следуют четыре шестнадцатеричные цифры, которые выполняются от **\u0000** до **\uffff**. (Информацию о шестнадцатеричных числах см. в Приложении C.) Например, слово “welcome” переводится на китайский язык с использованием двух иероглифов 欢 и 迎. Представления этих двух символов в Юникоде — **\u6B22\u8FCE**.

Программа в листинге 3.3 отображает два китайских иероглифа и три греческие буквы, как показано на рисунке 3.1.

**LISTING 3.3** DisplayUnicode.py

…

If no Chinese font is installed on your system, you will not be able to see the Chinese characters. In this case, delete **\u6B22\u8FCE** from your program to avoid errors. The Unicode codes for the Greek letters a, b, and g are **\u03b1**, **\u03b2**, and **\u03b3**.

Если в вашей системе не установлен китайский шрифт, вы не сможете увидеть китайские иероглифы. В этом случае удалите **\u6B22\u8FCE** из вашей программы, чтобы избежать ошибок. Коды Юникода для греческих букв α, β и γ: **\u03b1**, **\03b2** и **\03b3**.

3.3.3 Функции ord и chr