

Комментарии переводчика

Весь материал книги протестирован в среде Windows 10. При тестировании исходного кода за основу взят Python версии 3.9.5. Электронный материал к книге дополнен сведениями о функциональном программировании (см. главу 15 в электронном архиве).

Прилагаемый к книге адаптированный и скорректированный исходный код примеров следует разместить в подпапке домашней папки пользователя (/home/python_projects или C:\Users\ИМЯ_ПОЛЬЗОВАТЕЛЯ\python_projects).

Представление чисел

Для представления десятичных чисел в данной книге в качестве разделителя используется точка.

Для форматирования чисел в локальное для России представление (когда в качестве разделителя разрядов используется запятая) в Python имеется встроенная библиотека `locale`, которая позволяет настраивать форму представления чисел и соответствующим образом их форматировать. В общем случае для форматирования чисел следует начинать программу с приведенных ниже строк кода:

```
import locale # импортировать библиотеку по работе с локалями
loc = locale.getlocale() # запомнить текущую локаль
locale.setlocale(locale.LC_ALL, "Russian_Russia.1251") # изменить локаль
```

В последней строке задается форма написания чисел, принятая в России. Далее можно разместить собственный программный код, к примеру:

```
salary = 50000.0 # зарплата в рублях
print('Ваша ежемесячная зарплата составляет ',
      locale.format('%.2f', salary, grouping=True), sep='')
```

А в конце программы нужно вернуть локаль:

```
locale.setlocale(locale.LC_ALL, loc) # вернуть локаль
```

Если выполнить этот фрагмент кода, то получится результат, в котором представление чисел будет соответствовать принятому у нас стандарту, т. е. с разделением групп разрядов пробелом и использованием в качестве десятичного разделителя запятой:

```
Ваша ежемесячная зарплата составляет 50 000,00
```

Базовый набор сторонних библиотек

Хотя в книге установка и применение сторонних библиотек не рассматриваются, тем не менее, ниже приведена информация о базовом наборе инструментов, необходимых для дальнейшей работы. Эта информация ни к чему не обязывает, но избавит от затруднений и

послужит прекрасной отправной точкой для всех, кто интересуется программированием на Python.

Основополагающими являются следующие библиотеки:

- ◆ NumPy — базовая библиотека, необходимая для научных вычислений на Python;
- ◆ matplotlib — библиотека для работы с двумерными графиками;
- ◆ Pandas — инструмент для анализа структурных данных и временных рядов. Требует наличия `numpy` и некоторых других;
- ◆ Scikit-learn — интегратор классических алгоритмов машинного обучения. Требует наличия `numpy+mk1`;
- ◆ Scikit-image — коллекция алгоритмов для обработки изображений. Требует наличия `numpy+mk1`;
- ◆ SciPy — библиотека, используемая в математике, естественных науках и инженерном деле; требует наличия `numpy+mk1`;
- ◆ PyQt6 — библиотека инструментов для программирования графического интерфейса пользователя, требуется для работы инструментальной среды программирования Spyder.

Порядок установки библиотек

Для того чтобы работать с более свежими версиями библиотек и держать процесс установки под своим контролем, далее предлагается список команд установки библиотек. В случае нескольких библиотек может потребоваться установка из файлов `whl`, которые можно взять из репозитория `whl`-файлов (<http://www.lfd.uci.edu/~gohlke/pythonlibs/>).

```
pip install --upgrade pip
pip install numpy

pip install pandas
pip install scipy

pip install Scikit-learn

pip install scikit-image

pip install matplotlib
pip install jupyter
pip install pyqt6
```

Установка библиотек Python из `whl`-файлов

Библиотеки для Python можно разрабатывать не только на чистом Python. Довольно часто библиотеки программируют на C (динамические библиотеки), и для них пишут обертку Python, либо библиотеку пишут на Python, но для оптимизации узких мест часть кода — на C. Такие библиотеки получаются очень быстрыми, однако библиотеки с вкраплениями кода на C программисту на Python тяжелее устанавливать ввиду отсутствия соответствующих знаний либо необходимых компонентов и настроек в рабочей среде (в особенности в Windows). Для решения описанных проблем разработан специальный формат (файлы с расширением `whl`) для распространения библиотек, который содержит заранее скомпилирован-

ную версию библиотеки со всеми ее зависимостями. Формат WHL поддерживается всеми основными платформами (Mac OS X, Linux, Windows).

Установка производится с помощью менеджера библиотек `pip`. В отличие от обычной установки командой `pip install <имя_библиотеки>` вместо имени библиотеки указывается полное имя `whl`-файла (т. е. путь к файлу и собственно имя файла):

```
pip install <путь_к_whl-файлу>
```

Например,

```
pip install C:\temp\scipy-*****.whl
```

Откройте окно командой строки и при помощи команды `cd` перейдите в каталог, где размещен ваш `whl`-файл. В этом случае путь указывать не понадобится. Например,

```
pip install scipy-*****.whl
```

При выборе библиотеки важно, чтобы разрядность устанавливаемой библиотеки и разрядность интерпретатора совпадали. Пользователи Windows могут брать `whl`-файлы на веб-странице <http://www.lfd.uci.edu/~gohlke/pythonlibs/> Кристофа Голька из Лаборатории динамики флуоресценции Калифорнийского университета в г. Ирвайн. Библиотеки там постоянно обновляются, и в архиве содержатся все, какие только могут понадобиться.