Алисе посвящается

Содержание

Предисловие 12

Часть I. Введение в Flask 21

Глава 1. Установка 22
Использование виртуальных окружений 23
Установка пакетов Python с помощью pip 25

Глава 2. Структура простого приложения 26
Инициализация 26
Маршруты и функции представлений 26
Запуск сервера 28
Законченное приложение 29
Цикл запрос–ответ 31
Контексты приложения и запроса 31
Обработка запросов 33
Обработчики событий жизненного цикла 34
Ответы 35
Расширения Flask 37
Поддержка параметров командной строки с помощью Flask-Script 37

Глава 3. Шаблоны 40
Механизм шаблонов Jinja2 41
Отображение шаблонов 41
Переменные 42
Управляющие структуры 43
Интеграция Twitter Bootstrap с помощью Flask-Bootstrap 45
Нестандартные страницы с сообщениями об ошибках 49
Ссылки 52
Статические файлы 53
Локализация дат и времени с помощью Flask-Moment 54

Глава 4. Веб-формы 57
Защита от подделки межсайтовых запросов 57
Классы форм 58
Отображение форм в формат HTML 60

Обработка форм в функциях представления....62
Переадресация и сеансы....65
Всплывающие сообщения....67

Глава 5. Базы данных....70
Базы данных SQL....70
Базы данных NoSQL....71
SQL или NoSQL?....72
Фреймворки на Python поддержки баз данных....72
Интеграция с фреймворком Flask....74
Управление базой данных с помощью Flask-SQLAlchemy....74
Определение модели....75
Отношения....78
Операции с базами данных....80
Создание таблиц....80
Вставка строк....80
Изменение строк....82
Удаление строк....82
Извлечение строк....82
Операции с базой данных в функциях представления....85
Интеграция с интерактивной оболочкой Python....86
Миграция базы данных с помощью Flask-Migrate....87
Создание репозитория миграции....88
Создание сценария миграции....88
Обновление базы данных....89

Глава 6. Электронная почта....91
Поддержка электронной почты с помощью Flask-Mail....91
Отправка электронной почты из интерактивной оболочки Python....93
Интеграция поддержки электронной почты в приложение....93
Асинхронная отправка электронной почты....95

Глава 7. Структура больших приложений....97
Структура проекта....97
Параметры настройки....98
Пакет приложения....100
Фабричная функция приложения....100
Реализация функциональности приложения в виде макета....101
Сценарий запуска....104

Файл зависимостей ... 105
Модульные тесты ... 106
Настройка базы данных ... 108

Часть II. Пример: приложение социального блогинга ... 109

Глава 8. Аутентификация пользователей ... 110
Расширения аутентификации для Flask ... 110
Защита паролей ... 111
Хэширование паролей с помощью Werkzeug ... 111
Создание макета для поддержки аутентификации ... 114
Аутентификация пользователя с помощью Flask-Login ... 115
Подготовка модели User для аутентификации ... 115
Защита маршрутов ... 117
Добавление формы аутентификации ... 118
Аутентификация ... 119
Выход пользователя ... 121
Тестирование процедуры аутентификации ... 122
Регистрация нового пользователя ... 122
Добавление формы регистрации пользователя ... 123
Регистрация ... 125
Подтверждение создания учетной записи ... 126
Создание маркера подтверждения с помощью itsdangerous ... 126
Отправка электронных писем с инструкциями для подтверждения ... 128
Управление учетными записями ... 133

Глава 9. Роли пользователей ... 135
Представление ролей в базе данных ... 135
Присваивание ролей ... 138
Проверка роли ... 139

Глава 10. Профили пользователей ... 143
Информация для профиля ... 143
Страница профиля пользователя ... 144
Редактор профиля ... 147
Редактор профиля уровня пользователя ... 147

Редактор профиля уровня администратора ... 149
Аватары пользователей ... 152

Глава 11. Блогинг ... 156

Отправка и отображение сообщений ... 156
Сообщения из блогов на страницах профилей ... 159
Постраничный вывод длинных списков сообщений ... 160
Создание фиктивных сообщений ... 161
Постраничное отображение данных ... 163
Виджет постраничного отображения ... 164
Форматирование текста сообщений с помощью Markdown и Flask-PageDown ... 167
Flask-PageDown ... 168
Обработка форматированного текста на сервере ... 169
Постоянные ссылки на сообщения ... 171
Редактор сообщений ... 173

Глава 12. Читающие и читаемые ... 176

Пересмотр отношений в базе данных ... 176
Отношение «многие ко многим» ... 177
Самоссылочные отношения ... 179
Усовершенствованные отношения «многие ко многим» ... 180
Читающие и читаемые на странице профиля ... 183
Запрос сообщений читаемых пользователей с помощью операции соединения ... 186
Отображение сообщений читаемых пользователей на главной странице ... 189

Глава 13. Комментарии пользователей ... 194

Представление комментариев в базе данных ... 194
Отправка и отображение комментариев ... 196
Модерирование комментариев ... 198

Глава 14. Прикладные программные интерфейсы ... 204

Введение в REST ... 204
Все сущее является ресурсами ... 205
Методы запросов ... 206
Содержимое запросов и ответов ... 207
Поддержка версий ... 208

Веб-службы RESTful на основе Flask ... 209
Создание макета API ... 209
Обработка ошибок ... 210
Аутентификация пользователей с помощью Flask-HTTPAuth ... 212
Аутентификация на основе маркеров ... 214
Преобразование ресурсов в формат JSON и обратно ... 217
Реализация конечных точек ресурсов ... 220
Разбивка больших коллекций ресурсов на страницы ... 223
Тестирование веб-служб с помощью HTTPie ... 224

Часть III. Последняя миля ... 226

Глава 15. Тестирование ... 227
Получение отчета о степени охвата кода тестированием ... 227
Тестовый клиент Flask ... 231
Тестирование веб-приложений ... 231
Тестирование веб-служб ... 235
Сквозное тестирование с помощью Selenium ... 237
Насколько это необходимо? ... 241

Глава 16. Производительность ... 243
Регистрация медленных запросов к базе данных ... 243
Профилирование исходного кода ... 245

Глава 17. Развертывание ... 247
Порядок развертывания ... 247
Журналирование ошибок во время эксплуатации ... 248
Развертывание в облаке ... 249
Платформа Heroku ... 250
Подготовка приложения ... 250
Тестирование с помощью Foreman ... 256
Включение безопасного протокола HTTP с помощью Flask-SSLify ... 257
Развертывание командой git push ... 260
Просмотр журналов ... 260
Развертывание и обновление ... 261
Традиционный хостинг ... 261
Настройка сервера ... 261

Импортирование переменных окружения 262
Настройка журналирования 263

Глава 18. Дополнительные ресурсы 264
Использование интегрированной среды разработки 264
Поиск расширений для Flask 265
Участие в разработке Flask 266

Предметный указатель 267

Об авторе 270

Выходные данные 271

Предисловие

Flask отличается от других фреймворков тем, что позволяет разработчику сесть на место водителя и получить полный контроль над его приложением. Возможно, вам уже доводилось слышать фразу: «бороться с фреймворком». Такое происходит с большинством фреймворков при попытке реализовать нестандартное решение. Это может быть попытка использовать другой механизм управления базами данных или иной способ аутентификации пользователей. Отклонение от пути, предусмотренного разработчиками фреймворка, приносит массу неприятностей.

Фреймворк Flask не такой. Хотите использовать реляционную базу данных? Отлично, Flask поддерживает их. Предпочитаете базу данных NoSQL? Нет проблем, Flask способен работать и с ними. Хотите использовать механизм хранения данных собственной разработки или вообще решили обойтись без базы данных? Замечательно. Используя Flask, можно выбирать, какие его компоненты будут применяться в приложении, и даже писать собственные. Все в ваших руках!

Такая свобода объясняется тем, что фреймворк Flask изначально задумывался расширяемым. Он включает надежное ядро, обеспечивающее основные функциональные возможности, необходимые в любых веб-приложениях, и предполагает, что остальное будет реализовано сторонними разработчиками в форме расширений и, конечно же, вами.

В этой книге я расскажу о моих подходах к разработке веб-приложений с применением фреймворка Flask. Я не претендую на истину в последней инстанции, и вы должны рассматривать мои слова как рекомендации, а не как непреложное руководство к действию.

В большинстве книг, посвященных программированию, приводятся короткие фрагменты кода, демонстрирующие разные особенности обсуждаемых технологий по отдельности, оставляя за рамками «связывающий» код, необходимый для объединения этих разных фрагментов в действующее приложение. Я предпочел избрать иной подход. Все примеры, представленные в этой книге, являются частями единого приложения – сначала очень простого, а затем постепенно усложняющегося в каждой последующей главе. В начале пути это приложение состоит всего из нескольких строк кода, а к концу оно превращается в полноценное приложение социальных сетей и блогинга.

Кому адресована эта книга

Чтобы извлечь выгоду из этой книги, необходимо иметь некоторый опыт программирования на языке Python. Эта книга не предполагает предварительного знакомства с фреймворком Flask, но вы должны быть знакомы с такими понятиями языка Python, как пакеты, модули, функции, декораторы и объектно-ориентированное программирование. Нелишними будут также знакомство с исключениями и умение диагностировать проблемы по трассировке стека.

Следуя за примерами в книге, вы проведете немало времени в командной строке, поэтому вам также потребуются навыки работы в командной строке своей операционной системы.

Современные веб-приложения немыслимы без использования HTML, CSS и JavaScript. Приложение, разрабатываемое на протяжении всей книги, также использует их, но в самом тексте книги не приводятся подробности, касающиеся этих технологий. Знакомство с этими языками совершенно необходимо, если предполагаете писать законченные приложения, не прибегая к помощи разработчика, искушенного в клиентских технологиях.

Исходные тексты приложения, написанного для этой книги, я выложил в открытый доступ на сайте GitHub. Несмотря на то что GitHub позволяет загружать приложения в виде обычных ZIP- или TAR-архивов, я настоятельно рекомендую установить клиента Git и познакомиться с системой управления версиями, хотя бы с основными командами, позволяющими извлекать различные версии приложения непосредственно из репозитория. Короткий список команд, которые вам понадобятся, представлен в разделе «Использование программного кода примеров» ниже. Вы наверняка пожелаете применить систему управления версиями для собственных проектов, поэтому используйте эту книгу как повод изучить Git!

Наконец, не следует рассматривать эту книгу как полное и исчерпывающее руководство по фреймворку Flask. Здесь охватываются многие его особенности, но не упускайте из виду официальную документацию с описанием фреймворка[1].

Структура книги

Эта книга делится на три части.

---

[1] http://flask.pocoo.org/.

В первой части «Введение в Flask» исследуются основы разработки веб-приложений с применением фреймворка Flask и некоторых его расширений:

- глава 1 описывает установку и настройку фреймворка Flask;
- глава 2 погружает читателя в разработку простого приложения с помощью Flask;
- глава 3 знакомит с поддержкой шаблонов в приложениях Flask;
- глава 4 – с поддержкой веб-форм;
- глава 5 – с поддержкой баз данных;
- глава 6 – с поддержкой электронной почты;
- глава 7 описывает типичную структуру крупных и средних приложений.

Во второй части «Пример: приложение социального блогинга» описывается разработка открытого приложения социальных сетей и блогинга на основе фреймворка Flask, которое я создал для этой книги:

- глава 8 описывает реализацию системы аутентификации пользователей;
- глава 9 – реализацию системы ролей и привилегий пользователей;
- глава 10 – реализацию страниц профилей пользователей;
- глава 11 – создание интерфейса для блогинга;
- глава 12 – реализацию поддержки последователей;
- глава 13 – реализацию поддержки комментариев пользователей;
- глава 14 описывает реализацию прикладного программного интерфейса (Application Programming Interface, API).

В третьей части «Последняя миля» раскрываются некоторые важные задачи, не связанные с разработкой приложений непосредственно, которые необходимо решать перед публикацией приложения:

- глава 15 подробно описывает разные стратегии модульного тестирования;
- глава 16 представляет обзор приемов анализа производительности;
- глава 17 описывает варианты развертывания приложений на основе Flask в традиционных облачных окружениях;
- глава 18 перечисляет дополнительные источники информации.

Как работать с примерами программного кода

Примеры программного кода, описываемые в этой книге, доступны на сайте GitHub по адресу: **https://github.com/miguelgrinberg/flasky**.

История изменений в этом репозитории в точности соответствует порядку представления понятий в этой книге. Рекомендуется извлекать код из репозитория, начиная с самых ранних версий, и затем двигаться вперед по списку изменений, по мере чтения книги. Желающим сайт GitHub предоставляет возможность загружать каждое изменение в виде ZIP- или TAR-архива.

Если кто-то предпочтет извлекать исходный код примеров из репозитория Git, ему придется установить программу-клиента Git, которую можно бесплатно загрузить по адресу: **http://git-scm.com**. Ниже приводится команда, которая загрузит исходный код примеров из репозитория:

```
$ git clone https://github.com/miguelgrinberg/flasky.git
```

Команда `git clone` загрузит исходный код в каталог *flasky*, который будет создан в текущем каталоге. Этот каталог содержит не только исходный код; в него будет скопирован весь репозиторий Git с полной историей изменений приложения.

В первой главе будет предложено *извлечь* (*check out*) первоначальную версию приложения, и в соответствующих местах в книге будет предлагаться переходить дальше по истории изменений. Для извлечения исходного кода и перемещения по истории изменений используется команда `git checkout`, например:

```
$ git checkout 1a
```

Здесь `1a` в команде – это *тег*, именованная точка в истории развития проекта. Репозиторий размечен такими точками в соответствии с главами в книге, то есть тег `1a` соответствует начальной версии файлов приложения, описываемой в главе 1. Большинству глав соответствует более одного тега. Например, в репозитории имеются теги `5a`, `5b` и т. д., соответствующие последовательности версий, представленных в главе 5.

Помимо извлечения файлов из репозитория, может также потребоваться выполнить некоторые настройки. Например, иногда бывает необходимо установить дополнительные пакеты Python или внести

изменения в базу данных. Я буду сообщать об этом в соответствующие моменты.

В процессе чтения вам не придется изменять исходные файлы приложения, но если вы сделаете это, Git не позволит вам извлечь из репозитория следующую версию, чтобы не затереть локальных изменений. В этой ситуации, чтобы перейти к следующей версии, необходимо вернуть файлы в исходное состояние. Проще всего это сделать с помощью команды `git reset`:

```
$ git reset --hard
```

Эта команда затрет все локальные изменения, поэтому, если вы не желаете потерять их, сохраните резервные копии файлов перед запуском этой команды.

Иногда бывает желательно обновлять локальную копию репозитория из внешнего, расположенного на сервере GitHub, куда разработчики могут добавлять улучшения, расширения и исправления ошибок. Сделать это можно следующей последовательностью команд:

```
$ git fetch --all
$ git fetch --tags
$ git reset --hard origin/master
```

Команды `git fetch` обновят историю изменений и список тегов в локальной копии репозитория в соответствии с удаленным репозиторием GitHub, но ни одна из них не оказывает влияния на сами файлы с исходным кодом. Обновление этих файлов осуществляется командой `git reset`. И снова имейте в виду, что команда `git reset` затрет все локальные изменения, которые вы могли выполнить.

Еще одной полезной операцией является получение различий между двумя версиями приложения. Она может пригодиться всем, кто пожелает лучше разобраться в изменениях. Выполняется эта операция командой `git diff`. Например, чтобы увидеть различия между ревизиями `2a` и `2b`:

```
$ git diff 2a 2b
```

Различия будут сохранены в формате файла «заплаты» (patch). Этот формат не очень удобен для просмотра изменений, особенно если прежде вам не приходилось сталкиваться с подобными файлами. Гораздо удобнее в этом отношении инструменты с графическим интерфейсом, предлагаемые проектом GitHub. Например, различия между ревизиями `2a` и `2b` можно посмотреть на странице GitHub: **https://github.com/miguelgrin berg/flasky/compare/2a…2b**.

Использование программного кода примеров

Данная книга призвана оказать вам помощь в решении ваших задач. Вы можете свободно использовать примеры программного кода из этой книги в своих приложениях и в документации. Вам не нужно обращаться в издательство за разрешением, если вы не собираетесь воспроизводить значительные по объему части программного кода. Например, если вы разрабатываете программу и используете в ней несколько отрывков программного кода из книги, вам не нужно обращаться за разрешением. Однако в случае продажи или распространения компакт-дисков с примерами из этой книги вам необходимо получить разрешение от издательства O'Reilly. Если вы отвечаете на вопросы, цитируя данную книгу или примеры из нее, получение разрешения не требуется. Но при включении существенных объемов программного кода примеров из этой книги в вашу документацию вам необходимо будет получить разрешение издательства.

Мы приветствуем, но не требуем добавлять ссылку на первоисточник при цитировании. Под ссылкой на первоисточник мы подразумеваем указание авторов, издательства и ISBN. Например: «Flask Web Development by Miguel Grinberg (O'Reilly). Copyright 2014 Miguel Grinberg, 978-1-449-3726-2».

Если вам кажется, что использование примеров из книги нарушает правила добросовестного применения или условия, сформулированные выше, обращайтесь по адресу **permissions@oreilly.com**.

Типографские соглашения

В книге приняты следующие соглашения:

Курсив
: Применяется для выделения новых терминов, имен файлов и их расширений.

`Моноширинный шрифт`
: Применяется для представления листингов программного кода, а также элементов программ, таких как имена переменных и функций, баз данных, типов данных, переменных окружения, инструкций и ключевых слов.

Моноширинный жирный

Используется для выделения команд или других фрагментов текста, которые вводятся пользователем.

Моноширинный курсив

Используется для выделения текста, который должен замещаться значениями, предоставляемыми пользователями, или значениями, определяемыми контекстом.

Так выделяются советы и предложения.

Так выделяются примечания общего характера.

Так выделяются предупреждения и предостережения.

Safari® Books Online

Safari Books Online (**www.safaribooksonline.com**) – это виртуальная библиотека, содержащая авторитетную информацию в виде книг и видеоматериалов, созданных ведущими специалистами в области технологий и бизнеса.

Профессионалы в области технологий, разработчики программного обеспечения, веб-дизайнеры, а также бизнесмены и творческие работники используют Safari Books Online как основной источник информации для проведения исследований, решения проблем, обучения и подготовки к сертификационным испытаниям.

Библиотека Safari Books Online предлагает широкий выбор продуктов и тарифов для организаций, правительственных учреждений и физических лиц. Подписчики имеют доступ к поисковой базе данных, содержащей информацию о тысячах книг, видеоматериалов и рукописей от таких издателей, как O'Reilly Media, Prentice Hall Professional, Addison-Wesly Professional, Microsoft Press, Sams, Que, Peachpit Press, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, Course Technology, и десятков других. За подробной информацией о Safari Books Online обращайтесь по адресу: **http://www.safaribooksonline.com/**.

Ждем ваших отзывов

Направляйте свои комментарии и вопросы, касающиеся этой книги, по обычной почте:

O'Reilly Media
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (в Соединенных Штатах Америки или в Канаде)
707-829-0515 (международный)
707-829-0104 (факс)

Список опечаток, файлы с примерами и другую дополнительную информацию вы найдете на сайте книги: **http://www.bit.ly/flask-web-dev**.

Свои пожелания и вопросы технического характера отправляйте по адресу: **bookquestions@oreilly.com**.

Дополнительную информацию о наших книгах, курсах, конференциях и новостях вы найдете на веб-сайте издательства O'Reilly: **http://www.oreilly.com**.

Ищите нас на Facebook: **http://facebook.com/oreilly**.

Читайте нас в Твиттере: **http://twitter.com/oreillymedia**.

Смотрите нас на YouTube: **http://www.youtube.com/oreillymedia**.

Благодарности

Едва ли я смог бы написать эту книгу в одиночку. Огромную помощь оказали мне моя семья, коллеги, старые друзья и новые, с которыми я познакомился попутно.

Я хотел бы поблагодарить Брендана Кехлера (Brendan Kohler) за подробный технический отзыв и за его помощь в работе над главой о прикладных программных интерфейсах. Я также многим обязан Дэвиду Баумголду (David Baumgold), Тодду Бранхоффу (Todd Brunhoff), Сесиль Рок (Cecil Rock) и Мэтью Хьюгасу (Matthew Hugues), просматривавшим рукопись на разных стадиях готовности и дававшим мне очень полезные советы относительно того, о чем писать и как организовать материал.

Разработка примеров для этой книги потребовала значительных усилий. Я благодарен Даниэлю Хофманну (Daniel Hofmann), который просмотрел весь код приложения и внес множество советов по его улучшению. Я также хочу сказать спасибо моему сыну-подростку Дилану Гринбергу (Dylan Grinberg), который забросил свою игру на несколько выходных и помогал мне тестировать код на разных платформах.

Издательство O'Reilly запустило замечательную программу под названием «Early Release» (ранний выпуск), позволяющую нетерпе-

ливым читателям получить доступ к книге в процессе ее создания. Некоторые из читателей первых выпусков моей книги приняли участие в полезных обсуждениях и обмене опытом, что помогло существенно улучшить ее. Мне хотелось бы поблагодарить Сандипа Гупту (Sundeep Gupta), Дэна Кэрона (Dan Caron), Брайана Уисти (Brian Wisti) и Коди Скотта (Cody Scott) за их помощь в работе над этой книгой.

Сотрудники O'Reilly Media всегда благоволили мне. Прежде всего я хотел бы поблагодарить моего прелестного редактора Меган Бланшетт (Meghan Blanchette) за ее поддержку, советы и помощь с самой первой нашей встречи. Благодаря Мег я получил незабываемый опыт работы над моей первой книгой.

В заключение я хотел бы поблагодарить замечательное сообщество пользователей Flask.

Часть I

Введение в Flask

Глава 1

Установка

Flask[1] – это очень маленький фреймворк, такой маленький, что его часто называют «микрофреймворк». Он настолько мал, что после непродолжительного знакомства с ним вы сможете читать и понимать его исходный код.

Но быть маленьким не означает давать меньше, чем дают другие фреймворки. Flask изначально проектировался как расширяемый фреймворк – он имеет монолитное ядро, реализующее основные службы, а все остальное поддерживается посредством расширений. Поскольку вы можете выбирать только необходимые пакеты, в результате получается достаточно ограниченный комплект программных средств, не поддающийся неконтролируемому разбуханию и в точности соответствующий вашим потребностям.

Фреймворк Flask имеет две основные зависимости. Подсистемы маршрутизации, отладки и интерфейса WSGI (Web Server Gateway Interface) заимствованы из проекта Werkzeug, а поддержка шаблонов – из проекта Jinja2. Проекты Werkzeug и Jinja2 были созданы основными разработчиками Flask.

Flask не имеет встроенной поддержки доступа к базам данных, проверки веб-форм, аутентификации пользователей или других высокоуровневых задач. Существует и множество иных ключевых служб, необходимых большинству веб-приложений и доступных в виде расширений, интегрируемых с основными пакетами. Как разработчик вы можете выбирать расширения, лучше всего подходящие для вашего проекта, или даже писать собственные, если чувствуете, что вам это удастся лучше. Этим Flask отличается от крупных фреймворков, где выбор уже сделан за вас, который очень сложно изменить, если вообще возможно.

В данной главе вы узнаете, как установить Flask. Единственное предварительное условие – наличие компьютера с установленным языком Python.

Код примеров был протестирован с Python 2.7 и Python 3.3, поэтому рекомендуется использовать одну из этих двух версий.

---

[1] http://flask.pocoo.org.

Использование виртуальных окружений

Самый удобный способ установки Flask – воспользоваться виртуальным окружением. Виртуальное окружение – это отдельная копия интерпретатора Python, в которую можно установить пакеты, не оказывая влияния на глобальный интерпретатор Python, установленный в системе.

Виртуальные окружения удобны тем, что предотвращают конфликты между версиями и захламление системного интерпретатора Python посторонними пакетами. Создание виртуального окружения для каждого приложения гарантирует доступность только необходимых пактов, при этом системная установка интерпретатора остается чистой и служит всего лишь ресурсом для создания виртуальных окружений. Как дополнительное преимущество виртуальные окружения не требуют наличия у вас прав администратора.

Виртуальные окружения можно создавать с помощью сторонней утилиты `virtualenv`. Чтобы проверить наличие утилиты в системе, введите следующую команду:

```
$ virtualenv --version
```

Если в ответ вы получите сообщение об ошибке, значит, вам придется установить утилиту.

Python 3.3 включает встроенную поддержку виртуальных окружений в виде модуля venv и команды `pyvenv`. Команда `pyvenv` может использоваться вместо утилиты `virtualenv`, но имейте в виду, что виртуальные окружения, созданные с помощью `pyvenv` из установки Python 3.3, не включают утилиту `pip`, которую необходимо установить вручную. Это ограничение устранено в Python 3.4, где `pyvenv` можно использовать как полноценную замену `virtualenv`.

Большинство дистрибутивов Linux позволяет установить `virtualenv` в виде отдельного пакета. Например, пользователи Ubuntu могут установить эту утилиту командой:

```
$ sudo apt-get install python-virtualenv
```

В Mac OS X ее можно установить командой `easy_install`:

```
$ sudo easy_install virtualenv
```

Пользователям Microsoft Windows и других операционных систем, официально не поддерживающих пакета `virtualenv`, может потребоваться пройти более сложную процедуру установки.

Откройте в браузере адрес **https://bitbucket.org/pypa/setuptools** – домашнюю страницу дистрибутива `setuptools`. На этой странице най-

Конец ознакомительного фрагмента.
Приобрести книгу можно
в интернет-магазине «Электронный универс»
(e-Univers.ru)