



# ОГЛАВЛЕНИЕ

Вступление. Книга о языке программирования Python . . . . .	6
Язык Python . . . . .	6
Особенности книги . . . . .	8
Программное обеспечение . . . . .	9
Об авторе . . . . .	15
Обратная связь . . . . .	15
Благодарности . . . . .	15
Глава 1. Знакомство с Python . . . . .	16
Первая программа . . . . .	16
Использование различных сред разработки . . . . .	22
Среда разработки PyCharm . . . . .	23
Среда разработки Wing . . . . .	28
Среда разработки PyScripter . . . . .	33
Знакомство с переменными . . . . .	35
Ввод значения в программу . . . . .	39
Функция <b>eval ()</b> . . . . .	47
Знакомство со списками . . . . .	48
Знакомство с условным оператором . . . . .	56
Знакомство с оператором цикла . . . . .	59
Знакомство с функциями . . . . .	64
Резюме . . . . .	68
Задания для самостоятельной работы . . . . .	69
Глава 2. Основные операции . . . . .	71
Оператор цикла <b>while</b> . . . . .	71
Оператор цикла <b>for</b> . . . . .	81
Условный оператор <b>if</b> . . . . .	89
Тернарный оператор . . . . .	103
Обработка исключительных ситуаций . . . . .	108
Резюме . . . . .	120
Задания для самостоятельной работы . . . . .	122
Глава 3. Списки и кортежи . . . . .	124
Знакомство с кортежами . . . . .	124
Основные операции со списками и кортежами . . . . .	130
Создание выборки на основе списков и кортежей . . . . .	139
Вложенные списки и кортежи . . . . .	147
Копирование списков и кортежей . . . . .	153
Функции и методы для работы со списками . . . . .	158

Резюме . . . . .	167
Задания для самостоятельной работы . . . . .	169
Глава 4. Множества и словари . . . . .	171
Знакомство с множествами . . . . .	171
Операции с множествами . . . . .	176
Примеры использования множеств . . . . .	186
Знакомство со словарями . . . . .	192
Операции со словарями . . . . .	200
Резюме . . . . .	208
Задания для самостоятельной работы . . . . .	209
Глава 5. Работа с текстом . . . . .	211
Текстовые литералы . . . . .	211
Основные операции с текстом . . . . .	226
Методы для работы с текстом . . . . .	230
Примеры работы с текстом . . . . .	242
Резюме . . . . .	247
Задания для самостоятельной работы . . . . .	248
Глава 6. Функции . . . . .	250
Объявление и вызов функции . . . . .	250
Именованные аргументы функции . . . . .	261
Механизм передачи аргументов . . . . .	262
Значения аргументов по умолчанию . . . . .	266
Функции с произвольным количеством аргументов . . . . .	270
Локальные и глобальные переменные . . . . .	274
Вложенные функции . . . . .	277
Лямбда-функции . . . . .	279
Функция как аргумент и результат . . . . .	282
Рекурсия . . . . .	286
Декораторы функций . . . . .	289
Функции-генераторы . . . . .	292
Аннотации и документирование в функциях . . . . .	297
Резюме . . . . .	301
Задания для самостоятельной работы . . . . .	303
Глава 7. Файлы и данные . . . . .	305
Числовые данные . . . . .	305
Логические значения . . . . .	319
Дата и время . . . . .	322
Работа с файлами . . . . .	331
Резюме . . . . .	344
Задания для самостоятельной работы . . . . .	345
Глава 8. Классы и объекты . . . . .	347
Концепция классов и объектов . . . . .	347
Описание классов и создание объектов . . . . .	350

---

Конструкторы и деструкторы . . . . .	358
Объект реализации класса . . . . .	361
Операции с атрибутами классов и объектов . . . . .	372
Копирование объектов . . . . .	378
Документирование и декораторы . . . . .	382
Использование классов и объектов . . . . .	388
Резюме . . . . .	400
Задания для самостоятельной работы . . . . .	402
Глава 9. Наследование и специальные методы . . . . .	405
Знакомство с наследованием . . . . .	405
Множественное наследование . . . . .	413
Переопределение методов при наследовании . . . . .	417
Приведение типов . . . . .	432
Перегрузка операторов . . . . .	436
Доступ к атрибутам . . . . .	448
Индексирование объектов . . . . .	459
Вызов объекта . . . . .	463
Итераторы . . . . .	466
Резюме . . . . .	474
Задания для самостоятельной работы . . . . .	475
Глава 10. Обработка исключений и потоки . . . . .	477
Принципы обработки исключений . . . . .	477
Обработка исключений разных типов . . . . .	483
Использование объекта исключения . . . . .	484
Вложенные блоки для обработки исключений . . . . .	487
Искусственное генерирование исключений . . . . .	490
Создание классов исключений . . . . .	494
Использование исключений . . . . .	496
Знакомство с потоками . . . . .	506
Взаимодействие потоков . . . . .	517
Примеры использования потоков . . . . .	527
Резюме . . . . .	534
Задания для самостоятельной работы . . . . .	536
Глава 11. Программы с графическим интерфейсом . . . . .	538
Создание простого окна . . . . .	538
Окно с меткой и кнопкой . . . . .	540
Использование текстового поля . . . . .	543
Раскрывающийся список . . . . .	549
Опции, переключатели и другие компоненты . . . . .	557
Использование меню . . . . .	576
Работа с графикой . . . . .	596
Резюме . . . . .	611
Задания для самостоятельной работы . . . . .	612
Заключение. Python и программирование . . . . .	614

# Вступление

## КНИГА О ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON

Пошли, Скрипач, в открытый космос.

*Из к/ф «Кин-дза-дза»*

Сегодня существует много разных языков программирования. Некоторые из них популярны, другие не очень. Обычно популярность языка определяют по количеству программистов, которые на постоянной основе используют его в своей работе, или по запросам работодателей, ищущих сотрудников-программистов. Долгие годы традиционно популярными являются языки программирования Java, C++, C#, JavaScript и PHP. В последнее время в этой великолепной компании все чаще упоминается язык программирования Python. Более того, согласно некоторым опросам, язык Python уже занимает лидирующие позиции. Именно ему посвящена эта книга.

## Язык Python

Обо мне придумано столько небылиц, что я устаю их опровергать.

*Из к/ф «Формула любви»*

Тенденции таковы, что даже если язык Python и не является самым популярным на сегодня, то все равно нет сомнений в том, что масштабы его применения постоянно растут. Соответственно, увеличивается спрос на программистов, работающих с языком Python. Такая возрастающая популярность языка во многом объясняется его простотой, красотой и эффективностью. Спектр задач, решаемых с использованием Python, довольно внушителен. Поэтому изучение Python — выбор вполне разумный и многообещающий.

Чем же замечателен Python? Что в нем особенного? Ответы на эти вопросы не такие уж и простые. Тем более что многое зависит от того, с каким языком мы будем его сравнивать. Среди наиболее важных характеристик языка Python можно выделить следующие.

- Язык *интерпретируемый*. При первом запуске программы на выполнение для нее создается промежуточный код. Именно промежуточный код используется при выполнении программы. Если впоследствии в программу вносятся изменения, то при очередном запуске программы создается новый промежуточный код.



### НА ЗАМЕТКУ

---

Языки программирования бывают интерпретируемыми и компилируемыми. Если программа компилируется, то на основе исходного кода создается исполнительный (машинный) код, который и выполняется при запуске программы. Если речь идет об интерпретируемом языке, то программа, написанная на нем, выполняется построчно, без предварительной компиляции. Существует и промежуточный вариант — нечто среднее между компилированием и интерпретированием. В таком случае исходный код программы преобразуется в промежуточный код, который уже затем интерпретируется при выполнении.

Интерпретируемые языки позволяют больше вольности в описании и обработке данных. Программы, написанные на компилируемых языках, характеризуются относительно высокой скоростью выполнения.

- В плане синтаксиса язык Python прост и лаконичен. Он не содержит избыточных конструкций. С другой стороны, язык очень строгий: даже лишний пробел в программном коде может привести к ошибке.
- Язык Python поддерживает парадигму объектно-ориентированного программирования (ООП). Тем не менее он позволяет создавать программы, не использующие классы и объекты.



### НА ЗАМЕТКУ

---

Концепция ООП, реализуемая в языке Python, может стать сюрпризом для читателей, знакомых с такими языками программирования, как Java, C++ и C#. Напротив, те, кто знаком с языком JavaScript, обнаружат для себя некоторые знакомые моменты.

- Язык Python удобен для создания приложений с графическим интерфейсом.

- Еще одним фактором, способствующим популярности языка Python, является большое и дружное сообщество разработчиков, использующих этот язык. Нет недостатка и в свободно распространяемых программных продуктах (включая среды разработки), облегчающих знакомство и использование языка Python.

Выше представлен лишь очень общий и краткий перечень достоинств и особенностей языка. В детали мы погрузимся в основной части книги, когда будем рассматривать конкретные примеры и синтаксические конструкции.

## Особенности книги

Пацак пацака не обманывает. Это некрасиво, родной...

*Из к/ф «Кин-дза-дза»*

Цель этой книги — научить читателя программировать на языке Python. Но учиться можно по-разному. Скажем, можно слушать лекции в университете, можно посещать курсы по программированию, а можно пытаться научиться самостоятельно. Последний вариант — самый трудный, поскольку обычно рядом нет советчика, который мог бы подсказать или объяснить сложный момент. Вот именно для этого «сложного» случая в первую очередь и предназначена книга. Понятно, что совсем исключить «крутые повороты» при «прокладке маршрута» по изучению языка Python не получится. Но мы попытаемся свести к минимуму их количество.

Опыт показывает, что легче всего усвоить различные концепции программирования и подходы, когда они проиллюстрированы примерами. Как раз такая методика использована в этой книге. Принципиальная задача, которая при этом решается, — донести до читателя основную идею, причем не просто на некотором абстрактном уровне, а на уровне ее прикладной реализации с помощью программного кода. Теоретические сведения приводятся в объеме минимальном, но вместе с тем достаточном для качественного усвоения материала.

Структура книги такова, что в первой главе дается краткий обзор основных синтаксических конструкций языка Python. Это позволит читателю практически сразу, еще до завершения чтения книги, приступить к созданию несложных, но вполне функциональных программных

кодов. Этот прием применялся в книгах, посвященных другим языкам программирования, и получил неплохие отзывы читателей. Так что есть основания полагать, что он будет полезен и при изучении языка Python.

Главы после первой посвящены более детальному рассмотрению вопросов, связанных с эффективным программированием в Python. Среди рассмотренных тем: работа с данными разных типов, управляющие инструкции, списки и кортежи, множества и словари, работа с текстом, создание функций, операции с файлами, работа с классами и объектами, наследование и специальные методы, обработка исключительных ситуаций, создание потоков и многое другое. Последняя глава книги содержит полезную информацию, касающуюся создания приложений с графическим интерфейсом (с использованием библиотеки Tkinter). Для удобства усвоения материала каждая глава заканчивается кратким обобщением, в которое вынесены основные положения, рассмотренные и обсуждаемые в соответствующей главе. Также каждая глава содержит список заданий для самостоятельной работы.



#### **НА ЗАМЕТКУ**

---

Материал от главы к главе усложняется постепенно. Некоторые важные моменты достаточно часто повторяются (в разном контексте), особенно в начальных главах. Иногда одни и те же (или похожие) задачи решаются разными методами. Все это сделано намеренно. Цель простая — облегчить процесс усвоения информации и сформировать основы для понимания принципов программирования в Python.

## **Программное обеспечение**

Показывай свою гравицапу. Если фирменная вещь — возьмем!

*Из к/ф «Кин-дза-дза»*

Для составления программных кодов мало знать язык программирования (в данном случае Python). Понадобится также определенное программное обеспечение. Какое именно? Не помешала бы программа-редактор для набора кода. Хотя собственно программный код мы можем набирать хоть в текстовом редакторе, вроде Notepad. Для этого нам достаточно создать пустой текстовый документ, внести в него команды в соответствии с правилами языка Python и сохранить файл



с расширением `.py` (стандартное расширение для файлов с программами на языке Python).



### НА ЗАМЕТКУ

---

Помимо расширения `.py` файлы с Python-программами могут иметь расширение `.pyw`, если мы имеем дело с программами, в которых используется графический интерфейс (в операционной системе Windows). У файлов, связанных с Python-проектами, могут быть и другие расширения. Например расширение `.pyc` имеют файлы со скомпилированным промежуточным кодом (файлы с байт-кодом). Оптимизированный байт-код сохраняется в файле с расширением `.pyo`, а расширение `.pyd` используется для файлов с бинарным кодом динамических `dll`-библиотек в операционной системе Windows.

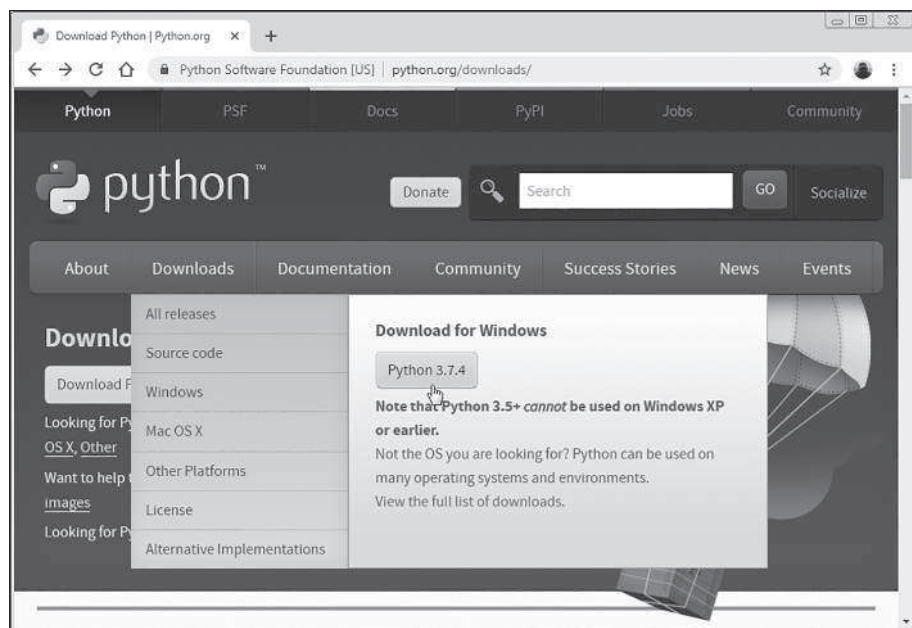
Но даже если мы так поступим, этого все же будет недостаточно. Нам еще как минимум понадобится программа-интерпретатор, которая сможет выполнить команды, написанные на языке Python. Другими словами, нам понадобится специальная программа, которая сможет понять код, который мы написали на языке Python, и исполнит этот код. Как отмечалось выше, такие программы называются *интерпретаторами*. Поэтому обойтись совсем без специального программного обеспечения мы не сможем. А поскольку программное обеспечение все равно придется устанавливать, то разумно воспользоваться всем спектром возможностей, доступных разработчику на языке Python. Тем более что предлагаемые для программирования на Python средства разработки довольно эффективны и часто бесплатны.

Самый разумный подход при создании программ на языке Python состоит в том, чтобы использовать *интегрированную среду разработки* (сокращенно *IDE* от *Integrated Development Environment*). Среда разработки — это специальное приложение, которое позволяет набирать, отлаживать и запускать на выполнение программные коды. Фактически среда разработки объединяет в себе сразу несколько программ. Это очень удобно, поскольку самые разные задачи, начиная с набора кода и до отладки приложения и запуска его на выполнение, реализуются через одну универсальную программу. Использовать среду разработки — разумно и удобно. Поэтому общая рекомендация состоит в том, чтобы использовать ее. Вопрос лишь, какую именно.

Существует довольно много сред разработки для языка Python. Здесь мы кратко остановимся лишь на некоторых, наиболее популярных

(и бесплатных). Но прежде чем перейти к обсуждению сред разработки, мы сделаем несколько замечаний относительно всего процесса установки программного обеспечения, необходимого для программирования на Python.

В первую очередь необходимо установить программу-интерпретатор (и некоторые сопутствующие утилиты). Для этого имеет смысл перейти на страницу поддержки языка `www.python.org`. Эта страница содержит много полезной информации. Там, кроме прочего, в разделе загрузок **Downloads** (адрес `www.python.org/downloads`) можно найти предназначенное для программирования на Python программное обеспечение. Веб-страница с ресурсами, предназначенными для загрузки, представлена на рис. В.1.



**Рис. В.1.** Страница `www.python.org/downloads` для загрузки программного обеспечения для программирования на Python

Следует загрузить соответствующие файлы и выполнить установку. Процесс установки простой и интуитивно понятный, поэтому особых комментариев не требует и обычно проходит без проблем. Стоит заметить, что в этом случае автоматически будет установлена и среда разработки, называемая IDLE. Это простая и надежная среда, которая вполне подойдет для эффективной работы с программными кодами на языке Python.



## НА ЗАМЕТКУ

Методы работы со средой IDLE, равно как и с другими средами разработки, кратко описываются в первой главе.

Если читателя по каким-либо причинам среда IDLE не устроит, можно воспользоваться другой средой. Благо, выбор достаточно большой.



## НА ЗАМЕТКУ

Обычно среды разработки устанавливаются без интерпретатора, поэтому рекомендуется сначала установить интерпретатор (например, загрузив файлы с сайта [www.python.org](http://www.python.org)), а уже после этого устанавливать среду разработки. В таком случае настройки среды, связанные с интерпретатором, скорее всего, будут выполнены автоматически.

Компания JetBrains предлагает для разработчиков на Python среду разработки, которая называется PyCharm. Информация об этой среде разработки (а также о других многочисленных разработках компании JetBrains) представлена на сайте [www.jetbrains.com](http://www.jetbrains.com). На рис. В.2 на странице открыт раздел **Tools**, в котором есть ссылка для загрузки установочных файлов среды разработки PyCharm.

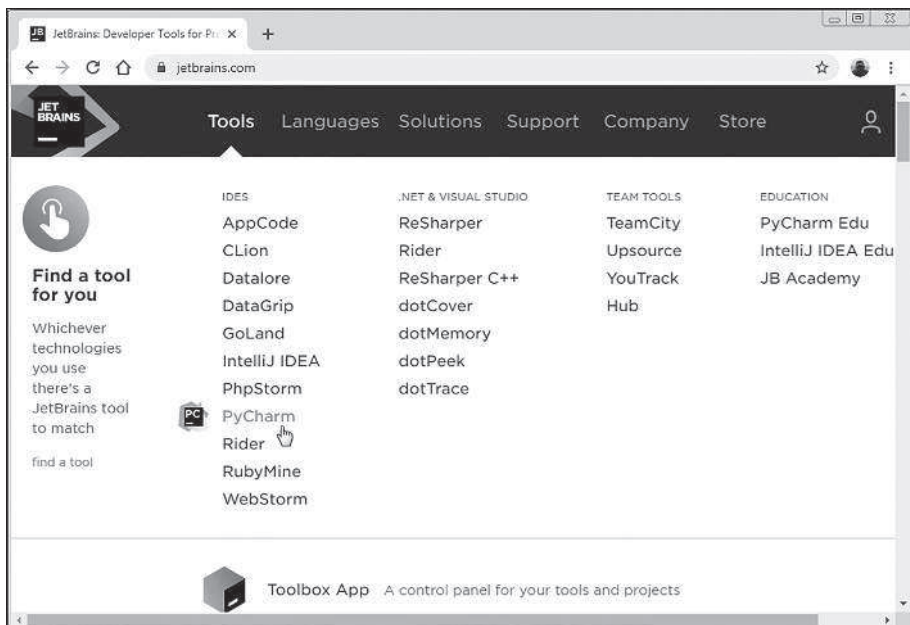
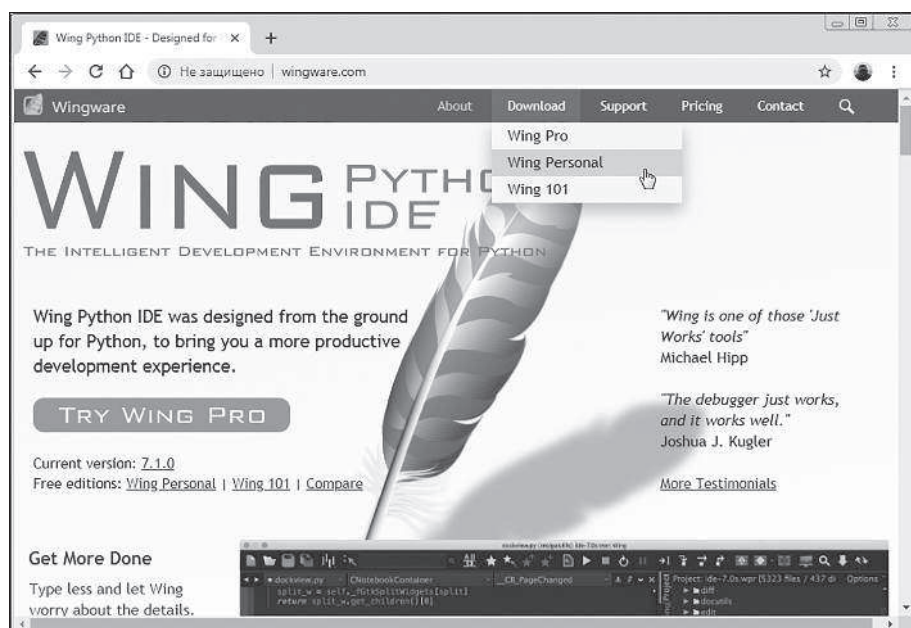


Рис. В.2. Страница [www.jetbrains.com](http://www.jetbrains.com) со ссылкой для загрузки установочных файлов среды разработки PyCharm

Процесс установки среды PyCharm достаточно простой. Это удобная и эффективная среда разработки. Правда, процесс создания приложений (по сравнению с тем, как это происходит при использовании других сред) может показаться немного запутанным, хотя это, конечно субъективное мнение. Вместе с тем среда PyCharm является, на мой взгляд, оптимальным выбором при работе с Python.

Достаточно удобной и функциональной является среда разработки Wing (продукт компании Wingware). На рис. В.3 показано окно браузера, в котором открыта страница [www.wingware.com](http://www.wingware.com) поддержки проекта.



**Рис. В.3.** Страница [www.wingware.com](http://www.wingware.com) со ссылкой для загрузки установочных файлов среды разработки Wing

Среда разработки Wing проста в использовании и содержит все основные утилиты, необходимые для эффективного программирования на Python.

Наконец, стоит упомянуть среду разработки PyScripter. На рис. В.4 показано окно браузера, открытое на странице с адресом <http://sourceforge.net/projects/pyscripter/>.

Это страница ресурса SourceForge, с помощью которого можно загрузить последнюю версию среды разработки PyScripter.

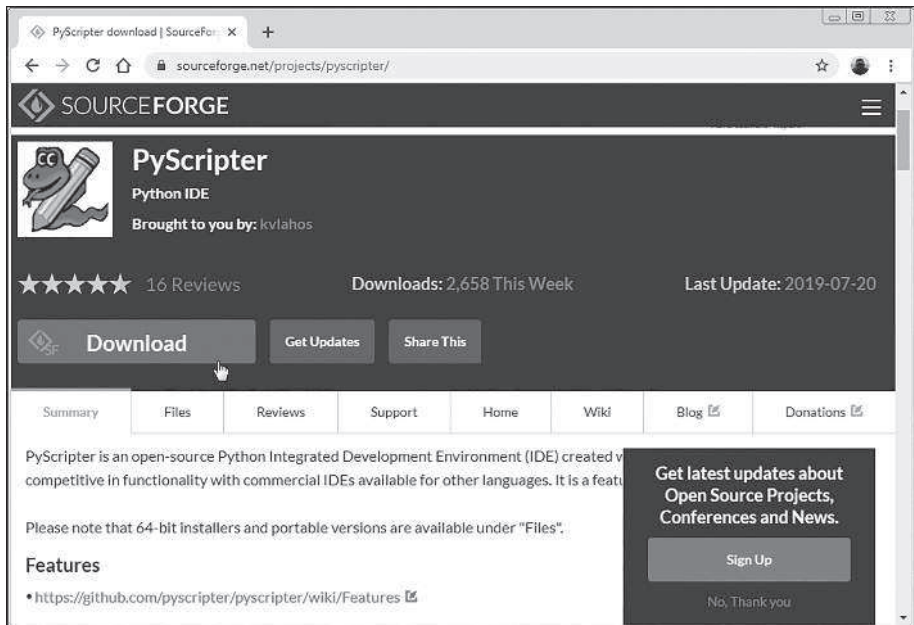


Рис. В.4. Страница *sourceforge.net/projects/pyscripter/* со ссылкой для загрузки установочных файлов среды разработки PyScripter

### **И** НА ЗАМЕТКУ

Поскольку страница поддержки среды PyScripter время от времени меняет свой адрес, то перед загрузкой установочных файлов стоит предварительно уточнить актуальный адрес для загрузки.

Характеризуя ситуацию в целом, стоит заметить, что большинство сред разработки предоставляют пользователю практически одинаковый «набор услуг». По крайней мере на начальном этапе, когда читатель только будет знакомиться с языком программирования Python, нет принципиальной разницы в том, какую именно среду разработки использовать. Это скорее вопрос эстетики, а не эффективности.

### **И** НА ЗАМЕТКУ

Ситуация со средами разработки довольно изменчива: какие-то среды становятся популярными, другие отходят на второй план. Поэтому следует понимать, что перечень доступных или предпочтительных сред разработки, приведенный выше, достаточно условный. Читатель вполне может использовать и иную среду.

## Об авторе

Товарищ, там человек говорит, что он — инопланетянин. Надо что-то делать...

*Из к/ф «Кин-дза-дза»*

Автор книги — *Васильев Алексей Николаевич*, доктор физико-математических наук, профессор кафедры теоретической физики физического факультета Киевского национального университета имени Тараса Шевченко. Автор книг по программированию и математическому моделированию. Сфера его научных интересов: физика жидкостей и жидких кристаллов, фазовые переходы и критические явления, биофизика, синергетика, математическая экономика, моделирование социально-политических процессов и математическая лингвистика.

## Обратная связь

— Слово лечит, разговор мысль отгоняет... Хотите беседовать, сударь?

— О чем?

— О чем прикажете.

*Из к/ф «Формула любви»*

Высказать свои замечания и предложения относительно этой и других книг автора можно по адресу электронной почты [alex@vasilev.kiev.ua](mailto:alex@vasilev.kiev.ua). Автор заранее благодарен своим читателям за конструктивную критику. Информацию об уже вышедших книгах, а также некоторые полезные материалы, касающиеся этих книг (например, программные коды примеров), можно найти на сайте [www.vasilev.kiev.ua](http://www.vasilev.kiev.ua).

## Благодарности

Вельми понеже... Весьма вами благодарен!

*Из к/ф «Иван Васильевич меняет профессию»*

Книги пишутся для того, чтобы их читали. Лучший стимул — осознание того, что твой труд кому-то нужен. Пользуясь случаем, хочу выразить самую искреннюю благодарность своим читателям: за интерес к книгам, за критические замечания, за желание становиться лучше и умнее.