

**ДЖОШ КОГЛИАТИ**

# **PYTHON**

**ДЛЯ НЕПРОГРАММИСТОВ  
САМОУЧИТЕЛЬ  
В ПРИМЕРАХ**

**МОСКВА  
ИЗДАТЕЛЬСТВО АСТ**

УДК 004.43  
ББК 32.973.26-018.1  
К57

*Книга "Non-Programmer's Tutorial for Python 3" на английском языке  
распространяется по лицензии*

**Creative Commons Attribution-ShareAlike License**

*(<https://creativecommons.org/licenses/by-sa/4.0/>).*

*Книга доступна в электронном виде по ссылке:*

*[https://en.wikibooks.org/w/index.php?title=Non-Programmer%27s\\_Tutorial\\_for\\_Python\\_3/Print\\_version&stableid=2694445](https://en.wikibooks.org/w/index.php?title=Non-Programmer%27s_Tutorial_for_Python_3/Print_version&stableid=2694445).*

**В подготовке книги принимали участие:**

Митчелл Эйкенс, Киа Моранте, Элизабет Коглиати,  
Джеймс А. Браун, Джо Оппегаард, Бенджамин Хелл

**Коглиати, Джош**

К57 Python для непрограммистов. Самоучитель в примерах / Дж. Коглиати. — Москва : Издательство АСТ, 2025. — 96 с. : ил. — (Учимся программировать).

ISBN 978-5-17-162198-8.

Эта книга в кратчайшие сроки поможет на базовом уровне освоить Python — идеальный для новичка, доступный и понятный язык программирования, позволяющий легко создавать интересные и креативные приложения. Буквально с первых страниц читатель, даже если он никогда в жизни не писал код, с легкостью начнет постигать Python и на основе простых практических заданий приступит к написанию собственных программ, сначала состоящих из двух-трех строк кода, а затем, по мере усвоения базовых навыков программирования, все более сложных. Информация в книге изложена простым языком, четко, последовательно и логично, с изрядной долей юмора, поэтому учебный материал легко воспринимается и запоминается. Делая упор на самом главном, автор книги предоставляет читателю основные знания о переменных, типах и структурах данных, функциях, циклах, логических выражениях и отладке программ, сразу же иллюстрируя теорию подробно прокомментированными практическими примерами.

**УДК 004.43**

**ББК 32.973.26-018.1**

**ISBN 978-5-17-162198-8**

Перевод на русский язык: ООО «Интеджер».  
Издание на русском языке: ООО «Издательство АСТ».

# Содержание

<b>Глава 1. Предисловие</b> .....	5
Другие ресурсы .....	5
<b>Глава 2. Введение</b> .....	6
Первым делом .....	6
Установка Python .....	7
Установка на Linux, BSD и Unix .....	7
Установка на Mac .....	8
Установка на Windows .....	8
Настройка переменной окружения PATH .....	8
Интерактивный режим .....	9
Создание и запуск программ .....	10
Имена файлов программ .....	10
Использование Python из командной строки .....	11
Запуск программ на Python в *nix .....	11
Где получить помощь .....	11
Документация по Python .....	11
Сообщество пользователей Python .....	11
<b>Глава 3. Здравствуй, мир!</b> .....	12
Что вы должны знать .....	12
Вывод на экран .....	12
Терминология .....	13
Выражения .....	14
Арифметические выражения .....	14
Общение с людьми (и другими разумными существами) .....	15
Примеры .....	15
Упражнения .....	16
<b>Глава 4. Кто идет?</b> .....	17
Ввод и переменные .....	17
Присвоение .....	18
Примеры .....	20
Упражнение .....	21
<b>Глава 5. Сосчитайте до 10</b> .....	22
Циклы While .....	22
Бесконечные циклы .....	23
Примеры .....	24
Последовательность Фибоначчи .....	24
Введите пароль .....	25
Упражнение .....	26
<b>Глава 6. Решения</b> .....	27

---

Оператор if .....	27
Примеры .....	28
Упражнения .....	31
<b>Глава 7. Отладка</b> .....	<b>33</b>
Что такое отладка? .....	33
Что должна делать программа? .....	33
Что делает программа? .....	34
Как исправить мою программу? .....	37
<b>Глава 8. Определение функций</b> .....	<b>38</b>
Создание функций .....	38
Переменные в функциях .....	39
Примеры .....	41
Упражнения .....	43
<b>Глава 9. Пример продвинутых функций</b> .....	<b>45</b>
Рекурсия .....	47
Примеры .....	48
<b>Глава 10. Списки</b> .....	<b>49</b>
Переменные с более чем одним значением .....	49
Дополнительные возможности списков .....	50
Примеры .....	54
Упражнения .....	56
<b>Глава 11. Циклы for</b> .....	<b>58</b>
<b>Глава 12. Булевы выражения</b> .....	<b>61</b>
Заметка о булевых операторах .....	64
Примеры .....	66
Упражнения .....	67
<b>Глава 13. Словари</b> .....	<b>68</b>
<b>Глава 14. Использование модулей</b> .....	<b>74</b>
Упражнения .....	75
<b>Глава 15. Подробнее о списках</b> .....	<b>76</b>
<b>Глава 16. Месть строк</b> .....	<b>80</b>
Нарезка строк (и списков) .....	83
Примеры .....	85
<b>Глава 17. Файловый ввод/вывод</b> .....	<b>86</b>
Продвинутое использование файлов .txt .....	90
Упражнения .....	91
<b>Глава 18. Работа с несовершенным, или Как обрабатывать ошибки</b> .....	<b>93</b>
Закрытие файлов при помощи оператора with .....	93
Перехват ошибок с помощью try .....	93
Упражнения .....	94
<b>Глава 19. Конец</b> .....	<b>94</b>
<b>Глава 20. Часто задаваемые вопросы</b> .....	<b>95</b>

# Предисловие

Все примеры исходного кода Python в этом руководстве являются общественным достоянием, поэтому вы можете изменять его и переиздавать под любой лицензией. Поскольку предполагается, что вы будете изучать программирование, лицензия Creative Commons Attribution-ShareAlike требует, чтобы все программы, созданные на основе исходного кода из этого учебника, были под этой лицензией. Но исходный код Python является общественным достоянием, так что это требование отменяется.

Данный учебник является относительно полной версией «Учебника для непрограммистов по Python 2.6». Более старые его версии и некоторые версии на корейском, испанском, итальянском и греческом языках доступны на сайте <http://jic.freeshell.org/easytut/>.

Данное учебное пособие предоставляет начальные знания о языке программирования Python и предназначено для читателей, не имеющих опыта программирования.

Если вы ранее программировали на других языках, я рекомендую использовать учебник «Python для программистов» (<https://docs.python.org/3/tutorial/index.html>), написанный Гвидо ван Россумом.

Если в процессе изучения книги у вас возникнут вопросы или комментарии, пожалуйста, пишите автору. Я приветствую любые вопросы и комментарии по этому учебнику и постараюсь ответить на них как можно более полно и исчерпывающе.

Спасибо Джеймсу А. Брауну за написание большей части главы об установке Python на Windows. Спасибо также Элизабет Коглиати за то, что она много жаловалась :) на оригинальный учебник (практически непригодный для непрограммиста), за его вычитку, а также за множество идей и комментариев к нему. Спасибо Джо Оппегарду за написание почти всех упреждений. Спасибо всем, кого я не упомянул.

## Другие ресурсы

- Домашняя страница Python (<http://www.python.org>)
- Документация по Python 3 (<https://docs.python.org/3/>)
- «A Byte of Python», пособие, написанное автором С. Н. Swaroop (<http://www.swaroopch.com/notes/python>)
- Переход на Python 3: подробное руководство (<http://python3porting.com/>)

## Глава 2

# Введение

## Первым делом

Итак, вы никогда раньше не программировали. При помощи этого пособия я попытаюсь научить вас создавать программы. На самом деле есть только один способ научиться программировать: вы должны *читать код* и *писать код* (так часто называют компьютерные программы). Я собираюсь показать вам много кода. Вы должны набирать код, который я привожу, чтобы посмотреть, что получится. «Играйте» с этим кодом и вносите в него свои изменения. Худшее, что может случиться, — он не будет работать.

Когда я ввожу код, он форматируется следующим образом:

```
##Python легко изучать
print("Hello, World!")
```

Это нужно для того, чтобы его можно было легко отличить от остального текста.

Если компьютер выводит что-то на экран, это будет отформатировано следующим образом:

```
Привет, мир!
```

(Обратите внимание, что мы применяем оператор печати `print`, но текст выводится на экран, и мы не используем бумагу. До того как у компьютеров появились экраны, результаты работы компьютерных программ печатались на бумаге).

Обратите внимание, что это учебник по Python 3, а это значит, что большинство примеров не будут работать в Python 2.7 и более ранних версиях. Кроме того, некоторые дополнительные библиотеки (библиотеки сторонних разработчиков) еще не конвертированы. Возможно, вы захотите изучить «Учебник для непрограммистов по Python 2.6». Однако различия между версиями не очень существенны, так что если вы изучите одну из них, то сможете читать программы, написанные для другой, без особых проблем.

Иногда текст, который вы вводите по запросу программы, будет выделен жирным шрифтом (но не во всех случаях), и текст, который вы увидите на экране, будет выглядеть следующим образом:

```
Стой!
```

```
Кто идет? Джош
```

```
Ты можешь идти, Джош.
```

Я также познакомлю вас с терминологией программирования — например, с тем, что программирование часто называют *кодингом*. Это не только даст вам возможность понять, о чем говорят программисты, но и поможет в процессе обучения. Теперь перейдем к более важным

вещам. Чтобы программировать на Python, вам необходимо установить на свой компьютер Python 3. Если у вас еще не установлен Python, зайдите на сайт [www.python.org/download](http://www.python.org/download) (<http://www.python.org/download/>) и получите нужную версию для вашей операционной системы. Скачайте ее, прочитайте инструкции и установите.

## Установка Python

Для программирования на Python вам понадобятся рабочая инсталляция Python и текстовый редактор. Python поставляется с собственным редактором, *IDLE*, который довольно неплох и вполне достаточен для начала. По мере углубления в программирование вы, вероятно, перейдете на какой-нибудь другой редактор, например *emacs*, *vi* или какой-либо еще.

Страница загрузки Python находится по адресу <http://www.python.org/download>. Python 2.7 и более старые версии не будут работать с материалами этого учебника. На сайте для скачивания доступны различные установочные файлы для разных компьютерных платформ. Здесь приведены некоторые конкретные инструкции для наиболее распространенных операционных систем.

### Установка на Linux, BSD и Unix

Возможно, вам повезло, и Python уже установлен на вашей машине. Чтобы проверить это, введите `python3` в командной строке. Если вы увидите что-то похожее на то, что показано в следующей главе, значит, все готово.

IDLE может потребоваться установить отдельно, из собственного пакета, например `idle3`, или как часть `python-tools`.

Если нужно установить Python, сначала попробуйте воспользоваться менеджером пакетов операционной системы или зайдите в репозиторий, где доступны ваши пакеты, и поищите Python 3. Python 3.0 был выпущен уже давно, во всех дистрибутивах должен быть доступен Python 3, так что вам, возможно, не придется компилировать его с нуля. В Ubuntu и Fedora доступны бинарные пакеты Python 3, но они еще не установлены по умолчанию, поэтому их нужно устанавливать специально.

Примерно так выглядит компиляция Python из исходного кода в Unix (если эти шаги не совсем понятны, возможно, вам стоит прочитать другое введение в \*nix, например, «Введение в Linux» (<http://tldp.org/LDP/intro-linux/html/index.html>)):

- Скачайте файл `.tgz` (используйте веб-браузер, чтобы получить `gzipped tar`-файл с сайта <https://www.python.org/downloads/release/python-343>)
- Распакуйте `tar`-файл (укажите правильный путь к месту его загрузки):

```
$ tar -xvzf ~/Download/Python-3.4.3.tgz
```

... список файлов в несжатом виде

- Перейдите в каталог и попросите компьютер скомпилировать и установить программу:

```
$ cd Python-3.4/
$ ./configure --prefix=$HOME/python3_install
... много выходных данных. Следите за сообщениями об ошибках здесь ...
$ make
... еще больше выходных данных. Надеюсь, никаких сообщений об ошибках ...
$ make install
```

- Добавьте Python 3 в свой путь. Сначала вы можете проверить его, указав полный путь. Вы должны добавить \$HOME/python3\_install/bin в переменную PATH bash:

```
$ ~/python3_install/bin/python3
Python 3.4.3 (... информация о размере и дате...)
[GCC 4.5.2] на linux2
Введите "help", "copyright", "credits" или "license" для получения
дополнительной информации.
>>>
```

Приведенные выше команды установят Python 3 в ваш домашний каталог, что, вероятно, и нужно, но если пропустить `--prefix=$HOME/python3_install`, он будет установлен в `/usr/local`. Если вы хотите использовать графический редактор кода IDLE, вам нужно убедиться, что в системе установлены библиотеки tk и tcl, а также файлы их разработки. Если они недоступны, то на этапе `make` вы получите предупреждение.

### Установка на Mac

Начиная с Mac OS X Tiger Python поставляется с операционной системой по умолчанию, но вам придется обновить его до Python 3, пока OS X не начнет включать Python 3 (проверьте версию, запустив `python3` в терминале командной строки). Также в стандартной установке может отсутствовать IDLE (редактор Python). Если вы хотите (заново) установить Python, возьмите дистрибутив для MacOS с сайта загрузки Python (<https://www.python.org/downloads/release/python-343/>).

### Установка на Windows

Загрузите соответствующий установщик Windows (установщик MSI для x86 (<https://www.python.org/ftp/python/3.4.3/python-3.4.3.msi>), если у вас не 64-битный процессор AMD или Intel). Запустите программу установки двойным щелчком мыши и следуйте подсказкам.

Дополнительную информацию см. на сайте <https://docs.python.org/3/using/windows.html#installing-python>.

### Настройка переменной окружения PATH

Переменная среды PATH — это список папок, разделенных точками с запятой, в которых Windows будет искать программу, когда вы попытаетесь выполнить ее, набрав ее имя в командной строке. Вы можете узнать текущее значение PATH, набрав эту команду в командной строке:

```
echo %PATH%
```

Самый простой способ постоянно изменять переменные окружения — вызвать встроенный редактор переменных окружения в Windows. В разных версиях Windows этот редактор выглядит по-разному.

**В Windows 8:** Нажмите клавишу Windows и введите *Control Panel*, чтобы найти панель управления Windows. Открыв *Панель управления*, выберите *Вид: Крупные значки*, затем нажмите на *Система*. В появившемся окне щелкните ссылку *Дополнительные параметры системы*, а затем нажмите кнопку *Переменные среды...*

**В Windows 7 или Vista:** Нажмите кнопку *Пуск* в левом нижнем углу экрана, наведите курсор на *Компьютер*, щелкните правой кнопкой мыши и выберите *Свойства* во всплывающем меню. Щелкните ссылку *Дополнительные параметры системы*, а затем нажмите кнопку *Переменные среды...*

**В Windows XP:** Щелкните правой кнопкой мыши значок *Мой компьютер* на рабочем столе и выберите *Свойства*. Выберите вкладку *Дополнительно*, затем нажмите кнопку *Переменные среды...*

Вызвав редактор переменных окружения, вы проделаете одни и те же действия, независимо от того, какая версия Windows у вас установлена. В разделе *Системные переменные* в нижней части редактора найдите переменную PATH. Если она есть, выберите ее и нажмите *Edit*. Полагая, что ваш корень Python — C:\Python34, добавьте эти две папки в свой путь (и убедитесь, что вы правильно расставили точки с запятой; между каждой папкой в списке должна быть точка с запятой):

```
C:\Python34
C:\Python34\Scripts
```

**Примечание:** если вы хотите дважды щелкнуть и запустить свои программы Python из папки Windows, чтобы при этом не исчезало окно консоли, вы можете добавить следующий код в нижнюю часть каждого скрипта:

```
#прекращает выход из консоли
end_prog = ""
while end_prog != "q":
    end_prog = input("введите q для выхода")
```

## Интерактивный режим

Зайдите в IDLE (также называемый графическим интерфейсом Python). Вы должны увидеть окно с текстом, похожим на этот:

```
Python 3.0 (r30:67503, Dec 29 2008, 21:31:07)
[GCC 4.3.2 20081105 (Red Hat 4.3.2-7)] на linux2
Введите "copyright", "credits" или "license()" для получения
дополнительной информации.
```

```
*****
```

```
Программное обеспечение персонального брандмауэра может предупредить
о соединении, которое IDLE устанавливает со своим подпроцессом,
```

используя внутренний интерфейс loopback этого компьютера. Это соединение не видно ни на одном внешнем интерфейсе, и никакие данные не отправляются в Интернет и не принимаются из него.

\*\*\*\*\*

IDLE 3.0

>>>

Символ >>> — это способ Python сообщить вам, что вы находитесь в интерактивном режиме. В интерактивном режиме то, что вы вводите, немедленно запускается. Попробуйте вести 1+1. Python ответит 2. Интерактивный режим позволяет протестировать и посмотреть, что будет делать Python. Если вам захочется «поиграть» с новыми операторами Python, перейдите в интерактивный режим и опробуйте их.

## Создание и запуск программ

Зайдите в IDLE, если вы еще там не находитесь. В меню вверху выберите *File*, затем *New File*.

В появившемся окне введите следующее:

```
print("Hello, World!")
```

Теперь сохраните программу: выберите в меню *File*, затем *Save*. Сохраните ее под именем «hello.py» (вы можете сохранить ее в любой папке). Теперь, когда программа сохранена, ее можно запустить.

Затем запустите программу, перейдя в меню *Run*, затем *Run Module* (или, если у вас старая версия IDLE, используйте *Edit*, затем *Run Script*). Это выведет *Hello, World!* в окно *\*Python Shell\**.

Более подробное введение в IDLE можно найти по адресу [http://hkn.eecs.berkeley.edu/~dyoo/python/idle\\_intro/index.html](http://hkn.eecs.berkeley.edu/~dyoo/python/idle_intro/index.html).

### Имена файлов программ

Очень полезно придерживаться некоторых правил относительно имен файлов программ на Python. Для самих программ эти правила не так важны, но если не соблюдать их для имен модулей (модули будут рассмотрены позже), могут возникнуть неожиданные проблемы.

1. Всегда сохраняйте программу с расширением `.py`. Не ставьте больше нигде точку в имени файла.
2. Для имен файлов используйте только стандартные символы: буквы, цифры, тире (-) и подчеркивание (\_).
3. Пробелы (" ") не должны использоваться вообще (вместо них используйте подчеркивание).
4. Используйте в начале имени файла только буквы (не цифры!).
5. Не используйте «неанглийские» символы (такие как å, ß, ç, ð, é, ò, ù) в именах файлов — или, что еще лучше, вообще не используйте их при программировании.