

# ОГЛАВЛЕНИЕ

<b>Отзывы о книге</b>	<b>5</b>
<b>Введение. Что такое программирование и почему оно件лезно для детей?</b>	<b>10</b>
Почему дети должны изучать программирование? .....	11
Где дети могут научиться писать код? .....	12
Как пользоваться этой книгой .....	13
<b>Глава 1. Основы Python: Знакомство со средой</b>	<b>16</b>
Начало работы с Python. ....	18
Написание программ на Python. ....	20
Запуск программ на Python. ....	21
Что вы узнали .....	22
Задачи по программированию .....	23
<b>Глава 2. «Черепашья» графика: рисование с Python</b>	<b>25</b>
Наша первая программа turtle. ....	25
Черепашка в ударе. ....	30
Черепашка закругляется. ....	31
Добавим красок. ....	33
Одна переменная, управляющая всей программой. ....	39
Что вы узнали .....	42
Задачи по программированию .....	43
<b>Глава 3. Числа и переменные: Python делает подсчеты</b>	<b>45</b>
Переменные: место, где мы храним данные. ....	45
Числа и математика в Python .....	48
Строки: реальные символы в Python. ....	57
Улучшим нашу спираль с помощью текста .....	59
Списки: храним все в одном месте .....	61
Python делает ваше домашнее задание. ....	64
Что вы узнали .....	66
Задачи по программированию .....	68
<b>Глава 4. Циклы — это весело (повторите пару раз)</b>	<b>69</b>
Создание собственных циклов for. ....	71
Улучшение программы с розеткой с помощью пользовательского ввода .....	75
Игровые циклы и циклы while .....	78
Семейная спираль. ....	81
Сведем все вместе: спираль уходит в народ .....	85
Что вы узнали .....	91
Задачи по программированию .....	92
<b>Глава 5. Условия (Что если?)</b>	<b>94</b>
Выражение if .....	96

Встречаем булевы выражения .....	98
Выражения <code>else</code> .....	103
Выражения <code>elif</code> .....	110
Сложные условия: <code>if</code> , <code>and</code> , <code>or</code> , <code>not</code> .....	111
Секретные послания .....	114
Что вы узнали .....	122
Задачи по программированию .....	124

## **Глава 6. Случайное веселье и игры: на удачу! 126**

Игра на угадывание .....	127
Цветные случайные спирали .....	130
Камень, ножницы, бумага .....	138
Выберите карту, любую карту .....	141
Кидаем кубики: игра в кости в стиле яцзы .....	149
Калейдоскоп .....	157
Что вы узнали .....	161
Задачи по программированию .....	164

## **Глава 7. Функции: да, у этого есть название 166**

Соберем все вместе с функциями .....	167
Параметры: покормите свою функцию .....	171
<b>Return</b> : важно не то, что ты получаешь, важно то, что ты возвращаешь .....	179
Прикосновение интерактивности .....	184
<code>ClickKaleidoscope</code> .....	195
Что вы узнали .....	199
Задачи по программированию .....	201

## **Глава 8. Таймеры и анимация: как поступил бы Дисней? 202**

Использование графического интерфейса <code>Pygame</code> .....	202
Правильный тайминг: двигайся и прыгай .....	213
Что вы узнали .....	231
Задачи по программированию .....	233

## **Глава 9. Взаимодействие с пользователем: подключаемся к игре 237**

Добавление интерактивности: щелчки и перетаски .....	238
Улучшенная интерактивность: взрыв из смайликов .....	245
<code>SmileyPop</code> , версия 1.0 .....	256
Что вы узнали .....	260
Задачи по программированию .....	261

## **Глава 10. Программирование игр: кодирование для развлечения 263**

Создание каркаса игры: <code>Smiley Pong</code> , версия 1.0 .....	264
Усложнение и конец игры: <code>Smiley Pong</code> , версия 2.0 .....	278
Добавление новых функций: <code>SmileyPop 2.0</code> .....	285
Что вы узнали .....	294
Задачи по программированию .....	296

## **Приложение А. Установка Python в среде Windows, macOS и Linux 298**

Python для Windows .....	298
--------------------------	-----

Python для macOS .....	307
Python для Linux .....	313
<b>Приложение Б. Установка и настройка Pygame в среде Windows, macOS и Linux</b>	<b>315</b>
Pygame для Windows .....	315
Pygame для macOS .....	320
Pygame для Linux .....	324
<b>Приложение В. Создание ваших собственных модулей</b>	<b>326</b>
Создание модуля <code>colorspiral</code> .....	327
Дополнительные ресурсы .....	331
<b>Приложение Г. Установка Pygame для Python 3 в среде macOS и Linux</b>	<b>332</b>
Pygame для Python 3.4 в среде macOS .....	332
Pygame для Python 3 в среде Linux .....	341
<b>Глоссарий</b>	<b>343</b>
<b>Об авторе</b>	<b>347</b>
<b>Об иллюстраторе</b>	<b>347</b>
<b>Благодарности</b>	<b>348</b>
<b>Предметный указатель</b>	<b>349</b>

# Введение

## ЧТО ТАКОЕ ПРОГРАММИРОВАНИЕ И ПОЧЕМУ ОНО ПОЛЕЗНО ДЛЯ ДЕТЕЙ?

Компьютерное программирование, или *написание программного кода*, — один из важнейших навыков, которым должен обладать каждый ребенок. Мы используем программы для решения математических задач, игр, они помогают нам повысить эффективность труда, выполнять однообразные задания, хранить и повторно использовать информацию, создавать что-то новое, а также поддерживать связь с друзьями и всем миром. Понимание принципов программирования делает всю эту мощь компьютеров легкодоступной.

Каждый может научиться программировать: это аналогично решению головоломки или загадки. Все, что нужно, — использовать логику, опробовать решение, еще немного поэкспериментировать — и наконец решить задачу. Время научиться программировать настало уже *сейчас!* Мы живем в совершенно особый исторический период: никогда раньше миллиарды людей не могли ежедневно общаться друг с другом, как мы можем общаться сегодня с помощью компьютеров. Мы живем в мире, полном новых возможностей: от электромобилей и роботов-сиделок до радиоуправляемых квадрокоптеров, доставляющих посылки и даже пиццу.

Если ваши дети начнут обучаться программированию уже сегодня, то они смогут оказать помощь в формировании облика этого быстро изменяющегося мира.

## Почему дети должны изучать программирование?

Существует множество веских причин изучать компьютерное программирование, но для меня эти две самые важные.

1. Программирование — это весело.
2. Программирование — это ценный профессиональный навык.

### Программирование — это весело

Новые технологии становятся частью нашей повседневной жизни. Каждая компания, благотворительная организация и любое дело могут ощутить на себе преимущества новых технологий. На рынке представлены приложения, помогающие пользователям покупать, передавать, вступать, играть, быть волонтером, общаться и делиться — программы помогают во всем, что вы только можете себе представить.

Хотели ли ваши дети создать собственный уровень любимой видеоигры? Программисты делают это! А как насчет создания их собственного приложения на телефон? Они могут осуществить этот замысел, написав программный код такого приложения на компьютере! Любая программа, игра, система или приложение, которые им довелось увидеть, были созданы с помощью блоков программного кода, аналогичных тем, что представлены в этой книге. Когда дети программируют, они принимают активное участие в развитии технологий: они не просто *развлекаются*, они делают технологии *увлекательными*!

### Программирование — это ценный профессиональный навык

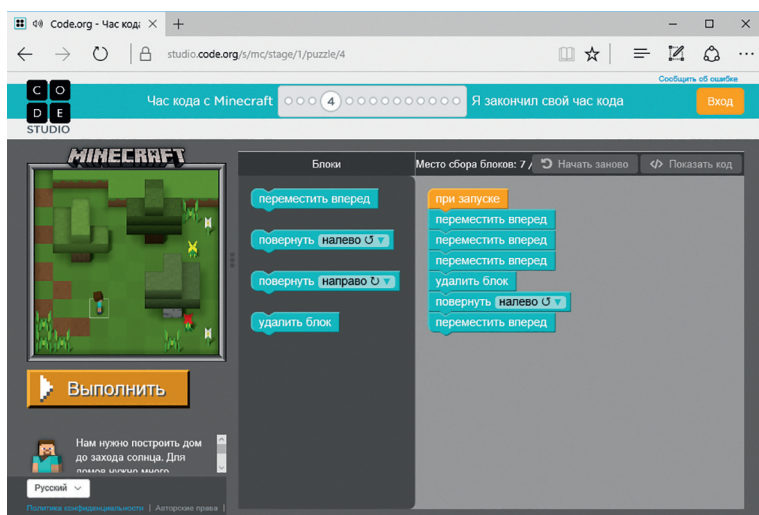
Программирование — это *важнейший* навык XXI века. Сегодня работа как никогда ранее требует способности решать поставленные задачи, новые технологии становятся неотъемлемой частью строительства карьеры во все большем количестве отраслей.

Бюро трудовой статистики США предсказывает, что в следующие пять лет на рынке будет создано более 8 миллионов рабочих мест, так или иначе связанных с новыми технологиями. Согласно справочнику занятости за 2014–2015 годы, семь из десяти быстро набирающих популярность должностей, для которых не требуется наличие степени магистра или доктора, приходится на компьютерную науку и информационные технологии (ИТ).

Марк Цукерберг был лишь студентом колледжа, работавшим в своей комнате в общежитии, когда создал первую версию Facebook в 2004 году. Спустя всего 10 лет 1,39 миллиарда людей *в месяц* использовали Facebook (источник: [newsroom.fb.com/company-info/](https://newsroom.fb.com/company-info/)). Еще никогда в истории идея, продукт или услуга не могли обзавестись аудиторией более миллиарда человек менее чем за 10 лет. Facebook демонстрирует уникальную способность новых технологий достигать большого количества людей так быстро, как никогда ранее.

## Где дети могут научиться писать код?

Эта книга — только лишь начало. Сегодня есть небывалое количество мест, где дети могут обучиться программированию. Сайты, такие как Code.org (рис. 1), Codecademy и бесчисленное количество подобных им, обучают основам и более углубленным аспектам написания программного кода на любом языке программирования. Когда вы с детьми завершите изучение этой книги, ваши дети могут записаться на бесплатные курсы с помощью сайтов, таких как EdX, Udacity или Coursera, для более углубленного обучения.



**Рис. 1.** Code.org в игровой форме обучит вас программированию на большом количестве языков

«Клубы программистов» — отличный способ сделать обучение развлекательным вместе с друзьями. Получение степени в колледже по одному из соответствующих профилей все же является одним из лучших способов

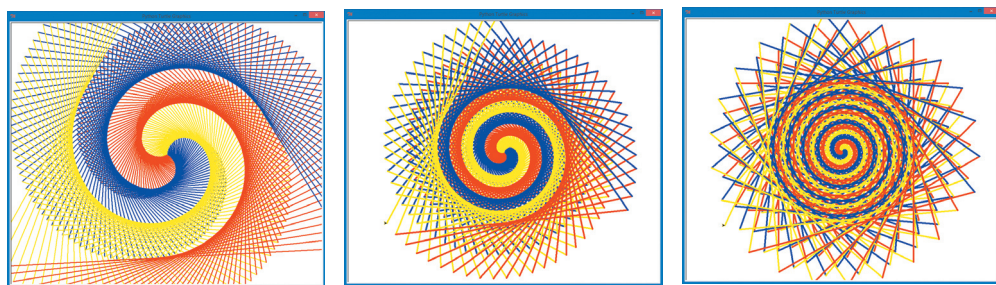
подготовки к началу карьеры. Но, если вы сейчас не рассматриваете колледж как один из доступных вариантов, ваши дети могут начать составлять свое портфолио программиста и демонстрировать навыки написания программного кода и решения поставленных задач уже сегодня.

## Как пользоваться этой книгой

Эта книга написана не только для детей, но и для родителей, учителей, студентов и взрослых, которые хотели бы изучить основы компьютерного программирования, как для развлечения, так и для получения доступа к новым рабочим местам в высокотехнологической экономике. Не имеет значения, сколько вам лет, вы все можете получить удовольствие, изучая основы программирования. Наилучший подход в данном случае заключается в сочетании работы и экспериментов.

### Исследуйте!

Изучение программирования может оказаться увлекательным, если вы желаете пробовать что-то новое. По мере следования программам, излагаемым в данной книге, попробуйте изменять цифры и текст в коде, чтобы увидеть, что произойдет с программой. Даже если вы выведете ее из строя, исправляя программу, вы научитесь чему-то новому. В худшем случае все, что вам понадобится сделать, — это снова ввести пример кода из книги или открыть последнюю рабочую версию программы. Цель изучения кода — попробовать что-то новое, обзавестись новым навыком — и получить возможность решать задачи новым способом. Убедитесь, что ваши дети играют с программой: тестируют код, изменяя что-либо в нем, сохраняя программу, запуская ее, наблюдая за результатами и исправляя ошибки при необходимости.



**Рис. 2.** Три спиральных узора, которые я создал, просто пробуя ввести разные значения в строке кода одной из программ

Так, например, я написал небольшой программный код для отображения спирального узора (рис. 2), затем вернулся, изменил некоторые цифры там и сям и попробовал запустить программу снова. Так я получил второй узор, абсолютно не похожий на первый, но от этого не менее потрясающий. Я снова менял еще несколько цифр кода — и получил взамен еще один красивый уникальный узор. Видите, что можно сделать, просто экспериментируя с кодом программы?

## Делайте это вместе!

Эксперименты с кодом — это хороший способ понять, как работают программы, но еще более эффективно работать совместно с кем-то. Учите ли вы ребенка, студента или обучаетесь самостоятельно, всегда гораздо *веселее* экспериментировать вместе, кроме того, это еще и более *эффективно*.

Например, особенность методики преподавания музыки Судзуки в том, что родители посещают занятия вместе с детьми и даже изучают программу немного вперед, чтобы помогать ребенку. Раннее начало обучения также является еще одной визитной карточкой метода Судзуки: дети могут приступить к формальному обучению в возрасте трех или четырех лет.

Я начал знакомить своих сыновей с программированием, когда им было два и четыре года, я поощрял их веселье, разрешал изменять небольшие кусочки каждой программы, например параметры цветов, форм или размеров форм.

Я изучил программирование, когда мне было 13. Я просто набирал на компьютере примеры из книг, затем изменял их, заставляя программы выполнять новые действия. Теперь, преподавая курсы по компьютерной науке, я часто даю студентам программу и поощряю их эксперименты с кодом, чтобы они могли создать нечто новое.

Если вы используете эту книгу для самостоятельного обучения, вы можете работать и вместе с другими людьми, например найдя друга, с которым можно было бы прорабатывать примеры, или открыв клуб по программированию после занятий, или среди знакомых (на сайтах [coderdojo.com](https://www.coderdojo.com) и [codeacademy.com/afterschool/](https://www.codeacademy.com/afterschool/) вы можете получить советы или почерпнуть несколько идей). Программирование в команде — это спорт!

## Ресурсы во Всемирной паутине

Файлы с кодом всех программ из этой книги можно скачать по адресу [https://eksmo.ru/files/Python\\_deti.zip](https://eksmo.ru/files/Python_deti.zip). Для более углубленного изучения

скачайте эти программы и поэкспериментируйте с ними. Если вы зашли в тупик, воспользуйтесь примером решения. Заходите на сайт!

## **Программирование = Решение задач**

Сколько бы ни было лет вашему ребенку: два, и он только учится считать, или 22, и он ищет новые вызовы, — эта книга и концепции, излагаемые в ней, освещают дорогу к достойному и вдохновляющему времяпрепровождению и лучшим карьерным перспективам. Люди, умеющие программировать и, соответственно, решать задачи быстро и эффективно, очень ценятся в современном мире, поэтому им достается интересная и благодарная работа.

Конечно, не все проблемы современного мира могут быть решены с помощью технологий, однако технология способствует общению, сотрудничеству, повышает информированность, позволяет действовать, причем делает это с неслыханными раньше скоростью и размахом. Если вы умеете писать программный код, вы можете решать задачи. Люди, способные решать поставленные перед ними задачи, обладают возможностью сделать этот мир лучше, поэтому начните программировать уже сегодня.



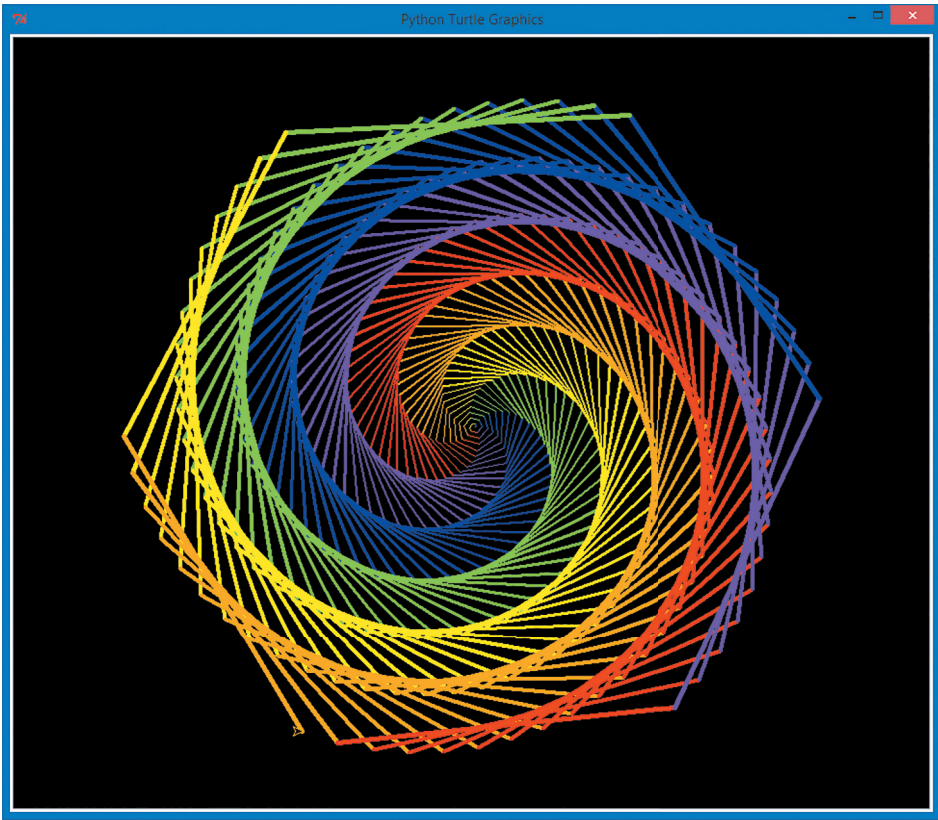
# Глава 1

## ОСНОВЫ PYTHON: ЗНАКОМСТВО СО СРЕДОЙ

Практически внутри всего может находиться компьютер: будь то телефон, автомобиль, часы, игровая консоль, тренажер, медицинское устройство, промышленное оборудование, поздравительная открытка или робот. С помощью компьютерного программирования, или *написания программного кода*, мы даем компьютеру инструкцию выполнить ту или иную задачу. Понимание принципов написания кода делает всю мощь компьютера легкодоступной для вас.

Компьютерные программы, также называемые *приложениями*, сообщают компьютеру, какие действия от него требуются. Так, веб-приложение может приказывать компьютеру отслеживать вашу любимую музыку, игровое приложение сообщает компьютеру, как отображать эпическое поле боя с реалистичной графикой, простое приложение может дать инструкцию компьютеру, как нарисовать красивую спираль, такую как шестиугольный орнамент на рис. 1.1.





**Рис. 1.1.** Цветной спиралевидный орнамент

Некоторые приложения содержат тысячи строк кода, а другие — всего несколько, как, например, программа *NiceHexSpiral.py*, код которой показан на рис. 1.2.

```
File Edit Format Run Options Windows Help
#NiceHexSpiral.py
import turtle
colors=['red', 'purple', 'blue',
        'green', 'yellow', 'orange']
t=turtle.Pen()
turtle.bgcolor('black')
for x in range(360):
    t.pencolor(colors[x%6])
    t.width(x/100+1)
    t.forward(x)
    t.left(59)
Ln: 11 | Col: 14
```

**Рис. 1.2.** *NiceHexSpiral.py* — это короткая программа на языке Python, рисующая спираль, наподобие показанной на рис. 1.1

Эта короткая программа рисует цветную спираль, показанную на рис. 1.1. В качестве иллюстрации для этой книги мне хотелось использовать какую-то приятную картинку, поэтому я решил эту задачу с помощью компьютерной программы. Сначала я набросал идею, а затем начал писать программный код.

В этой главе мы скачаем, установим и научимся использовать программы, которые помогут нам писать код и создавать различные приложения, любые, какие вы только можете себе представить.

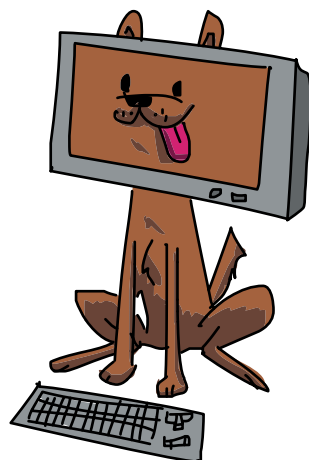
## Начало работы с Python

Чтобы начать писать программный код, вам необходимо начать говорить на компьютерном языке. Компьютерам требуются пошаговые инструкции, и они могут понимать только определенные языки. Точно так, как человек из России может не понимать по-английски, компьютеры понимают только языки, специально созданные для них. Компьютерный, или программный, код пишется на языках программирования, таких как Python, C++, Ruby или JavaScript. Эти языки позволяют нам «разговаривать» с компьютером и сообщать ему команды. Представьте, что вы обучаете свою собаку трюкам: когда вы произносите команду «сидеть», она садится, говорите «голос» — она лает. Собака понимает эти простые команды, однако остальные ваши слова — нет.

Также у компьютеров есть определенные ограничения, однако они готовы выполнить все, что вы попросите на их языке. Язык, которым мы будем пользоваться в этой книге, называется *Python* — простой и мощный язык программирования. Язык Python преподается на вводных курсах по компьютерной науке в старших классах и колледжах, кроме того, он используется для создания одних из самых мощных приложений на свете, таких как Gmail, Google Maps и YouTube.

Чтобы вы могли пользоваться языком Python на своем компьютере, давайте совместно сделаем следующие три шага.

1. Скачаем Python.
2. Установим Python на компьютер.
3. Протестируем Python, написав одну или две простые программы.



## 1. Скачивание Python

Python — это бесплатная программа, которую легко загрузить на компьютер с официального веб-сайта, изображенного на рис. 1.3.

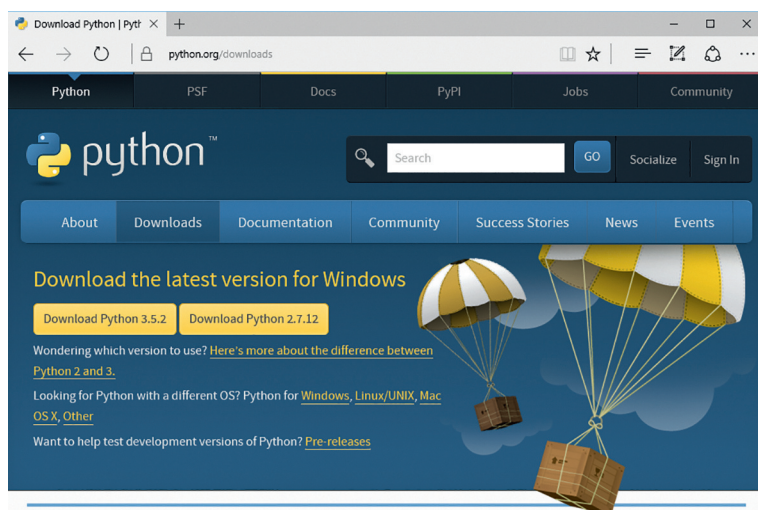


Рис. 1.3. Сайт Python упрощает загрузку Python на компьютер

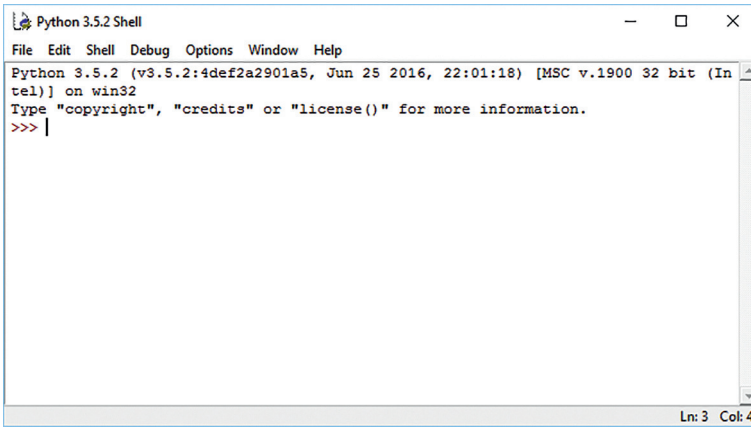
С помощью своего браузера перейдите на сайт [python.org](http://python.org), наведите указатель мыши на кнопку меню **Downloads** и щелкните по кнопке, текст которой начинается с **Python 3**.

## 2. Установка Python

Найдите только что загруженный файл (возможно, он расположен в папке *Загрузки (Downloads)*), дважды щелкните мышью по этому файлу, чтобы запустить программу и установить Python и редактор IDLE на компьютер. IDLE — это программа, которую мы будем использовать для ввода и запуска наших программ на языке Python. Более подробные инструкции по установке программного обеспечения см. Приложение А.

## 3. Тестирование Python

В меню **Пуск (Start)** или папке *Приложения (Applications)* найдите программу IDLE и запустите ее. Вы увидите текстовое окно для ввода команд, наподобие того, что изображено на рис. 1.4. Эта программа называется оболочкой Python. *Оболочка* — это окно или экран, позволяющие пользователю вводить команды и строки кода.



**Рис. 1.4.** Оболочка Python IDLE — наш командный пункт для изучения Python

Строка `>>>` называется *приглашением*, и ее наличие означает, что компьютер готов принять вашу первую команду. Компьютер просит вас сказать ему, что требуется сделать. Введите:

---

```
print("Здравствуй, мир!")
```

---

и нажмите на клавиатуре клавишу **Enter** или **Return**. На экране вы должны увидеть, что Python напечатал текст, который вы указали в скобках: *Здравствуй, мир!* Ну вот: вы написали свою первую программу!

## Написание программ на Python

Обычно вам потребуется писать программы длиной более чем одна строка, поэтому Python поставляется вместе с *редактором*, позволяющим писать более длинные программы. В окне IDLE перейдите к меню **File** (Файл) и выберите команду **New Window** (Новое окно) или **New File** (Новый файл). На экране появится пустое окно с заголовком *Untitled*.

Давайте напишем чуть более длинную программу на Python. В новом, пустом, окне введите следующие три строки кода:

---

```
# YourName.py
name = input("Как тебя зовут?\n")
print("Привет, ", name)
```

---

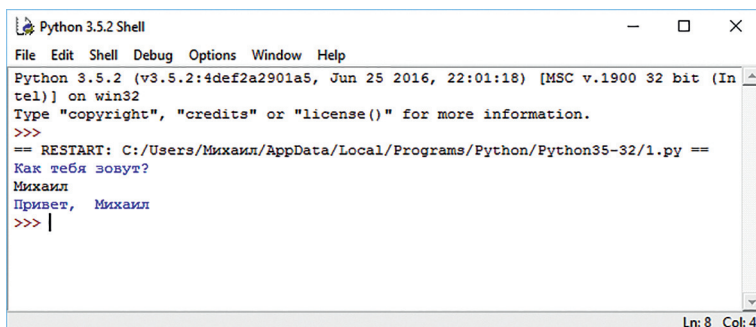
Первая строка называется *комментарием*. Комментарии, начинающиеся с символа «решетка» (`#`), представляют собой заметки программиста или

какие-то напоминания, игнорируемые компьютером. В данном примере комментарий — это лишь заметка, напоминающая нам об имени программы. Вторая строка просит пользователя ввести свое имя и запоминает его как `name`. Третья строка печатает слово "Привет, ", после которого следует имя, введенное пользователем. Обратите внимание, что текст "Привет, " и `name` разделяет запятая (,).

## Запуск программ на Python

Перейдите к пункту **Run** (Запуск) строки меню, расположенной над текстом вашей программы, и выберите команду **Run Module** (Запуск модуля). Данное действие *запустит*, или передаст в обработку, инструкции вашей программы. Сначала система попросит вас сохранить программу. Давайте назовем наш файл `YourName.py`. Данное действие сообщит вашему компьютеру, что программу нужно сохранить как файл с именем `YourName.py`, где расширение `.py` обозначает, что перед нами программа на языке Python.

После сохранения и запуска файла вы увидите, что окно оболочки Python запустило программу, выведя на экран вопрос Как тебя зовут?. Введите свое имя на следующей строчке и нажмите клавишу **Enter**. Программа напечатает слово Привет, после которого будет следовать введенное вами имя. Так как вы не просили программу сделать что-либо еще, то программа будет завершена — и вы опять увидите приглашение `>>>`, как показано на рис. 1.5.



```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSVC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/Михаил/AppData/Local/Programs/Python/Python35-32/1.py ==
Как тебя зовут?
Михаил
Привет, Михаил
>>> |
```

**Рис. 1.5.** Компьютер знает мое имя!

Маленьким ученикам, таким как мой трехлетний сын, очень интересно объяснить, что программа просит их ввести свое имя. Макс уже знает буквы своего имени, поэтому он набирает на клавиатуре `м-а-к-с`, и ему очень нравится, когда я говорю, что программа ответила ему Привет, макс. Спросите своего ребенка, не хотел ли бы он, чтобы программа ответила ему что-то другое. Макс