



3.10.0



Ийти

Учебник по Python

Python - это простой в освоении мощный язык программирования. Он имеет эффективные высокоуровневые структуры данных и простой, но эффективный подход к объектно-ориентированному программированию. Элегантный синтаксис и динамическая типизация Python вместе с его интерпретируемой природой делают его идеальным языком для написания сценариев и быстрой разработки приложений во многих областях на большинстве платформ.

Интерпретатор Python и обширная стандартная библиотека находятся в свободном доступе в исходной или двоичной форме для всех основных платформ на веб-сайте Python <https://www.python.org/> и могут свободно распространяться. Этот же сайт также содержит дистрибутивы и указатели на многие бесплатные сторонние модули, программы и инструменты Python, а также дополнительную документацию.

Интерпретатор Python легко расширяется новыми функциями и типами данных, реализованными на C или C++ (или других языках, вызываемых из C). Python также подходит в качестве языка расширения для настраиваемых приложений.

Это руководство неформально знакомит читателя с основными концепциями и функциями языка и системы Python. Помогает иметь интерпретатор Python под рукой для практического опыта, но все примеры являются самодостаточными, поэтому учебное пособие также можно читать в автономном режиме.

Описание стандартных объектов и модулей см. [В Стандартной библиотеке Python](#). [Справочник по языку Python](#) дает более формальное определение языка. Чтобы написать расширения на C или C++, прочтите [Справочное руководство по расширению и](#)



3.10.0



Ийти

Это руководство не пытается быть исчерпывающим и охватывать каждую функцию или даже каждую часто используемую функцию. Вместо этого он представляет многие из наиболее примечательных функций Python и даст вам хорошее представление об особенностях и стиле языка. Прочитав его, вы сможете читать и писать модули и программы Python и будете готовы узнать больше о различных модулях библиотеки Python, описанных в [Стандартной библиотеке Python](#).

[Глоссарий](#) также стоит идти до конца.

- [1. Разжигание аппетита](#)
- [2. Использование интерпретатора Python](#)
 - [2.1. Вызов переводчика](#)
 - [2.1.1. Передача аргумента](#)
 - [2.1.2. Интерактивный режим](#)
 - [2.2. Интерпретатор и его окружение](#)
 - [2.2.1. Кодирование исходного кода](#)
- [3. Неформальное введение в Python](#)
 - [3.1. Использование Python в качестве калькулятора](#)
 - [3.1.1. Числа](#)
 - [3.1.2. Струны](#)
 - [3.1.3. Списки](#)
 - [3.2. Первые шаги к программированию](#)
- [4. Дополнительные инструменты управления потоком](#)
 - [4.1. ifЗаявления](#)
 - [4.2. forЗаявления](#)
 - [4.3. range\(\) Функция](#)
 - [4.4. breakи continueутверждения, и elseпункты о циклах](#)
 - [4.5. passЗаявления](#)
 - [4.6. matchЗаявления](#)
 - [4.7. Определение функций](#)
 - [4.8. Подробнее об определении функций](#)
 - [4.8.1. Значения аргументов по умолчанию](#)
 - [4.8.2. Аргументы ключевых слов](#)



3.10.0



Идти

[слова](#)

- [4.8.3.2. Только позиционные параметры](#)
- [4.8.3.3. Аргументы только для ключевых слов](#)
- [4.8.3.4. Примеры функций](#)
- [4.8.3.5. Резюме](#)
- [4.8.4. Списки произвольных аргументов](#)
- [4.8.5. Распаковка списков аргументов](#)
- [4.8.6. Лямбда-выражения](#)
- [4.8.7. Строки документации](#)
- [4.8.8. Аннотации функций](#)
- [4.9. Интермеццо: стиль кодирования](#)
- [5. Структуры данных](#)
 - [5.1. Подробнее о списках](#)
 - [5.1.1. Использование списков как стеков](#)
 - [5.1.2. Использование списков в качестве очередей](#)
 - [5.1.3. Составить список](#)
 - [5.1.4. Вложенные описания списков](#)
 - [5.2. del-заявление](#)
 - [5.3. Кортежи и последовательности](#)
 - [5.4. Наборы](#)
 - [5.5. Словари](#)
 - [5.6. Методы зацикливания](#)
 - [5.7. Подробнее об условиях](#)
 - [5.8. Сравнение последовательностей и других типов](#)
- [6. Модули](#)
 - [6.1. Подробнее о модулях](#)
 - [6.1.1. Выполнение модулей как скриптов](#)
 - [6.1.2. Путь поиска модуля](#)
 - [6.1.3. «Скомпилированные» файлы Python](#)
 - [6.2. Стандартные модули](#)
 - [6.3. dir\(\) Функция](#)
 - [6.4. Пакеты](#)
 - [6.4.1. Импорт * из пакета](#)
 - [6.4.2. Ссылки внутри пакета](#)
 - [6.4.3. Пакеты в нескольких каталогах](#)



3.10.0



Ийти

- 7.1.1. Форматированные строковые литералы
- 7.1.2. Метод String format ()
- 7.1.3. Форматирование строк вручную
- 7.1.4. Старое форматирование строки
- 7.2. Чтение и запись файлов
 - 7.2.1. Методы файловых объектов
 - 7.2.2. Сохранение структурированных данных с json
- 8. Ошибки и исключения
 - 8.1. Ошибки синтаксиса
 - 8.2. Исключения
 - 8.3. Обработка исключений
 - 8.4. Вызов исключений
 - 8.5. Цепочка исключений
 - 8.6. Пользовательские исключения
 - 8.7. Определение действий по очистке
 - 8.8. Предусмотренные действия по очистке
- 9. Занятия
 - 9.1. Несколько слов об именах и объектах
 - 9.2. Области действия и пространства имен Python
 - 9.2.1. Пример областей и пространств имен
 - 9.3. Первый взгляд на классы
 - 9.3.1. Синтаксис определения класса
 - 9.3.2. Объекты класса
 - 9.3.3. Экземплярные объекты
 - 9.3.4. Объекты метода
 - 9.3.5. Переменные класса и экземпляра
 - 9.4. Случайные замечания
 - 9.5. Наследование
 - 9.5.1. Множественное наследование
 - 9.6. Частные переменные
 - 9.7. Шансы и концы
 - 9.8. Итераторы
 - 9.9. Генераторы
 - 9.10. Генераторные выражения
- 10. Краткая экскурсия по стандартной библиотеке.



3.10.0



Идти

- 10.3. Аргументы командной строки
- 10.4. Перенаправление вывода ошибок и завершение программы
- 10.5. Соответствие строковому шаблону
- 10.6. Математика
- 10.7. Доступ в Интернет
- 10.8. Даты и время
- 10.9. Сжатие данных
- 10.10. Измерение производительности
- 10.11. Контроль качества
- 10.12. Батарей в комплекте
- 11. Краткий обзор стандартной библиотеки - часть II.
 - 11.1. Форматирование вывода
 - 11.2. Шаблоны
 - 11.3. Работа с макетами записи двоичных данных
 - 11.4. Многопоточность
 - 11.5. логирование
 - 11.6. Слабые ссылки
 - 11.7. Инструменты для работы со списками
 - 11.8. Десятичная арифметика с плавающей запятой
- 12. Виртуальные среды и пакеты
 - 12.1. Вступление
 - 12.2. Создание виртуальных сред
 - 12.3. Управление пакетами с помощью pip
- 13. Что теперь?
- 14. Интерактивное редактирование ввода и подстановка истории
 - 14.1. Завершение вкладки и редактирование истории
 - 14.2. Альтернативы интерактивному переводчику
- 15. Арифметика с плавающей запятой: проблемы и ограничения
 - 15.1. Ошибка представления
- 16. Приложение
 - 16.1. Интерактивный режим
 - 16.1.1. Обработка ошибок
 - 16.1.2. Исполняемые скрипты Python
 - 16.1.3. Интерактивный файл запуска