

25 фреймворков Python, которыми нужно овладеть в 2023 году

11.05.2023

Python быстро стал одним из самых популярных языков программирования благодаря своей простоте, универсальности и обилию ресурсов, доступных для изучающих этот язык. Часто это первый язык, который изучают начинающие программисты, и он также широко используется в промышленности для самых разных целей – от веб-разработки до анализа данных. Сообщество Python огромно, и, по данным GitHub, Python продолжает расти: в 2022 году его популярность увеличится на 22,5%. Кроме того, в 2023 году он займет третье место среди самых используемых языков программирования. Это основная причина, по которой мы имеем доступ к широкому разнообразию фреймворков, большинство из которых с открытым исходным кодом. В этой статье мы рассмотрим самые популярные и полезные фреймворки Python, с которыми вы должны познакомиться в 2023 году.

Что такое фреймворк Python?

Фреймворк Python – это набор модулей Python, предоставляющих набор общих функций, которые могут быть использованы в качестве структуры для создания приложений любого типа. Фреймворки предназначены для упрощения процесса разработки путем предоставления общих рекомендаций по созданию программного обеспечения и абстрагирования от некоторых более сложных или повторяющихся задач. Это позволяет вам сосредоточиться на написании уникальной и индивидуальной логики для ваших приложений, а не изобретать велосипед.

Примером повторяющейся задачи может служить обработка HTTP-запросов. Поскольку большинство веб-приложений должны обрабатывать этот тип запросов, разработчики используют

существующие фреймворки, которые облегчают эту функцию, вместо того чтобы писать все с нуля или повторно использовать один и тот же код в разных проектах. Теперь, когда у нас есть понятие фреймворков, давайте рассмотрим некоторые типы фреймворков Python.

Типы фреймворков Python

В Python существует множество фреймворков для различных типов разработки. Давайте рассмотрим некоторые из них.

Полнофункциональный фреймворк

Полнофункциональный фреймворк Python – это набор инструментов, который предоставляет все необходимое разработчику для создания полноценного веб-приложения от начала до конца. Сюда входит способ создания фронтенда – например, системы шаблонов и подхода к отображению информации для пользователя – и бэкенда, включая такие общие функции, как создание записей в базе данных, обработка HTTP-запросов и контроль безопасности приложения.

Микрофреймворк

Микрофреймворк – это минималистичный фреймворк, который предоставляет только основные компоненты, необходимые для создания какого-либо приложения. Он разработан, чтобы быть легким и легко расширяемым, что делает его хорошим выбором для небольших проектов или для разработчиков, которые хотят больше контроля над своим кодом.

Асинхронные фреймворки

Асинхронный фреймворк предназначен для обработки параллелизма и параллельности, позволяя разработчикам создавать приложения, которые могут выполнять несколько задач одновременно.

Менеджер пакетов Python (pip)

pip – это менеджер пакетов для пакетов Python. Если вы уже установили Python, у вас есть доступ к интерфейсу командной строки pip, который позволяет устанавливать пакеты Python. Пакет – это изолированная и многократно используемая коллекция кода, которая обеспечивает определенную функциональность для решения общей задачи (веб-разработка, аутентификация, GUI). Все фреймворки, которые мы рассмотрим далее, являются частью стандартной библиотеки Python, что означает, что они доступны в виде пакета pip.

Вы можете установить любой пакет, доступный в Python Package Index (PyPI), с помощью следующей команды:

```
pip install package-name
```

Вы также можете установить пакеты из текстового файла – например, файла требований для проекта – с помощью следующей команды.

```
pip install -r requirements.txt
```

Python Фреймворк против Python Библиотеки

Библиотека Python – это набор функций и методов, которые можно использовать для выполнения определенных задач, таких как разбор данных или генерация случайного числа. Библиотека обычно более ограничена по объему и предназначена для использования в качестве инструмента в рамках более крупного приложения. Фреймворк Python, с другой стороны, предоставляет полный набор инструментов и функций, которые могут быть использованы для создания целого приложения. Обычно он определяет рабочий процесс, которому следуют разработчики программного обеспечения при создании проекта. Это не означает, что возможности фреймворка нельзя расширить с помощью других пакетов, которые часто называют плагинами.

Зачем использовать фреймворк Python?

Существует несколько преимуществ использования фреймворка Python при разработке приложений. Давайте рассмотрим некоторые из них:

- **Фреймворки облегчают работу разработчика программного обеспечения:** Предоставляя структуру для кода и набор инструментов и функций, которые упрощают процесс разработки, фреймворки могут облегчить и сделать более эффективным создание приложений для разработчиков.
- **Фреймворки способствуют организации кода:** Хорошо разработанный фреймворк помогает обеспечить организованность и сопровождаемость кода, что облегчает понимание и изменение исходного кода в будущем.
- **Фреймворки могут повысить производительность:** Предоставляя готовые компоненты и инструменты и придерживаясь отраслевых стандартов, фреймворки позволяют разработчикам сосредоточиться на уникальных аспектах своих приложений, а не тратить время на выполнение базовых задач.

Теперь, когда у вас достаточно теоретической базы, давайте рассмотрим 25 лучших фреймворков Python в 2023 году.

25 лучших фреймворков Python в 2023 году

Здесь представлены 25 самых популярных фреймворков Python, разделенных на разделы по функциональности.

1. Микрофреймворки Python
2. Веб-фреймворки Python
3. Фреймворки графического интерфейса пользователя

4. Системы машинного обучения Python
5. Python Frameworks для научных вычислений
6. Python Frameworks для тестирования
7. Асинхронные фреймворки

Микрофреймворки Python

Ниже представлены легкие микрофреймворки, полезные, когда вы хотите быстро создать простое приложение или не потреблять много памяти на вашем сервере.

Flask

Hello Kinsta User



This is a hello world with Flask!"

Flask – это легкий микрофреймворк, используемый для быстрого создания простых веб-приложений. Он включает в себя поддержку шаблонов Jinja (способ повторного использования HTML-кода), обработку запросов и сигнализацию приложений.

Для запуска приложения Flask требуется менее шести строк кода:

```
from flask import Flask
app = Flask(__name__)
```

```
@app.route("/")
def hello_world():
    return "Hello World"
```

Особенности:

- Легкий и простой в использовании
- Высокая гибкость
- Построен поверх стандартной библиотеки Python
- Поддержка маршрутизации и представлений (контроллеров)
- Поддержка шаблонов с помощью Jinja
- Отличная документация и сообщество
- Поддержка обработки cookies и аутентификации пользователей

Вы также получите доступ к некоторым пакетам расширений, таким как Flask-RESTful, который добавляет поддержку для создания мощных REST API, и Flask-SQLAlchemy, удобный способ использования SQLAlchemy в вашем приложении flask.

Bottle



Хотите создавать до смешного легкие веб-приложения без каких-либо других зависимостей? Bottle – это легкий Python-

микрофреймворк, разработанный для легкого создания небольших или средних веб-приложений. Он не включает никаких внешних зависимостей, кроме стандартной библиотеки Python, Bottle построен на основе стандарта WSGI (Web Server Gateway Interface) и совместим с большинством веб-серверов и версий Python.

Особенности:

- Простота в изучении и использовании
- Встроенная система шаблонов
- Подходит для создания небольших веб-приложений и API
- Очень низкое использование памяти
- Поддержка HTTP, форм и маршрутизации

CherryPy



CherryPy – это минималистичный фреймворк для разработки веб-приложений на языке Python. Это объектно-ориентированная среда (ООП), которая позволяет создавать веб-приложения так же, как и любая другая ООП на Python. Он существует с 2002 года и широко применяется в производственных средах на самых разных веб-сайтах – от простых функциональных до требующих высокой

сложности. Например, оба гигантских программных сервиса Netflix и Hulu используют CherryPy как часть своей инфраструктуры. Основная задача CherryPy – обрабатывать HTTP-запросы и сопоставлять их с адекватной логикой, написанной разработчиками. Это означает, что по умолчанию CherryPy не предоставляет доступ к базе данных или шаблонизацию HTML, оставляя всю логику приложения на ваше усмотрение.

Особенности:

- Зрелый фреймворк
- Обработка HTTP-запросов
- Гибкая и расширяемая система плагинов
- Фреймворк только для бэкендов
- Объектно-ориентированная разработка
- Стабильный API

Веб-фреймворки Python

Ниже представлены полноценные веб-фреймворки, которые включают в себя компоненты для таких распространенных задач веб-разработки, как доступ к базе данных, обработка форм и безопасность. В отличие от микрофреймворков, веб-фреймворки Python предоставляют все необходимое из коробки. При создании веб-приложения необходимо учитывать свои потребности. Иногда более чем достаточно просто использовать WordPress.

Django



The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.



[Django Documentation](#)

Topics, references, & how-to's



[Tutorial: A Polling App](#)

Get started with Django



[Django Community](#)

Connect, get help, or contribute

Django – один из самых популярных и широко используемых веб-фреймворков в экосистеме Python. Это надежный полнофункциональный фреймворк, который включает в себя все необходимое для создания полноценного веб-приложения. Впервые он был представлен в 2005 году как Pythonic способ создания сайта для рассылки новостей. Сегодня это одна из самых используемых библиотек Python для веб-разработки, с почти 8 миллионами загрузок в месяц. Некоторые из крупнейших сайтов, созданных когда-либо, были изначально созданы с помощью Django, например, Instagram, Discus, Pinterest, Bitbucket и Chess.com.

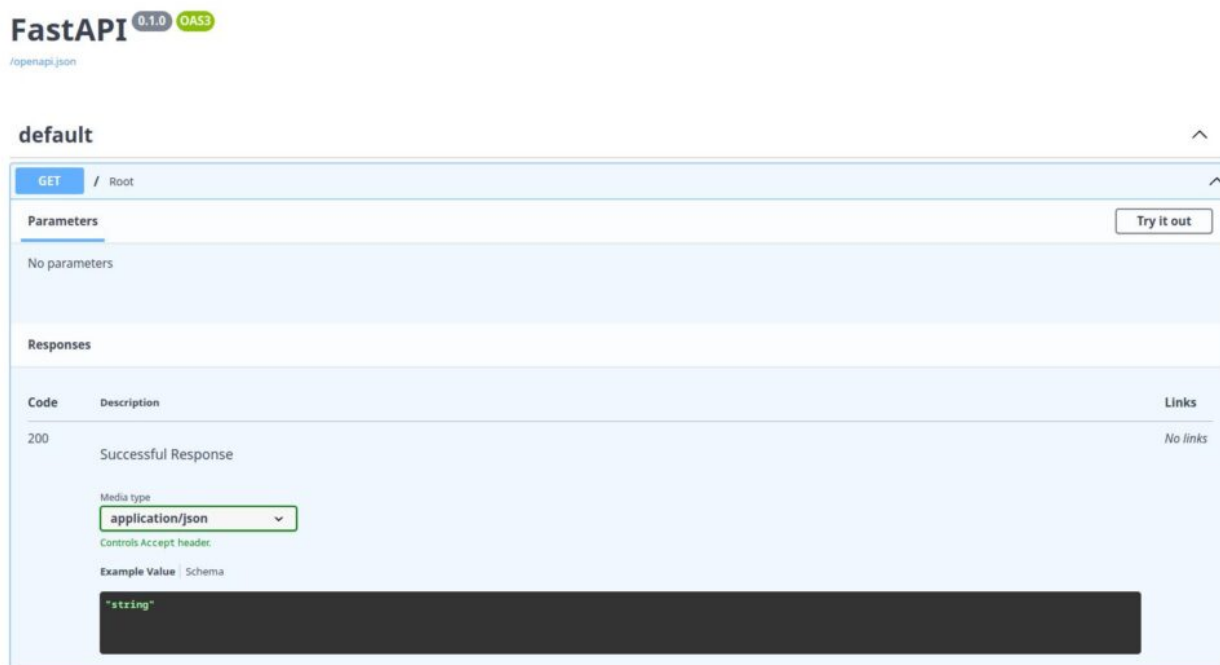
Django следует архитектуре модель-вид-шаблон (MVT) и включает встроенные функции, такие как шаблонизация, управление базой данных, панель администратора, сигналы и представления (для управления логикой вашего бэкенда). Для освоения Django желательно иметь сильные основы объектно-ориентированного программирования на Python, а также функционального программирования. Вы не ошибетесь, выбрав Django для своего следующего веб-проекта. Это мощный веб-фреймворк, который предоставляет все необходимое для создания быстрых и надежных веб-сайтов. А если вам нужны дополнительные возможности –

скажем, возможность создания REST API для использования с современными фронтенд-фреймворками, такими как React или Angular – вы можете использовать расширения, такие как Django REST framework.

Особенности:

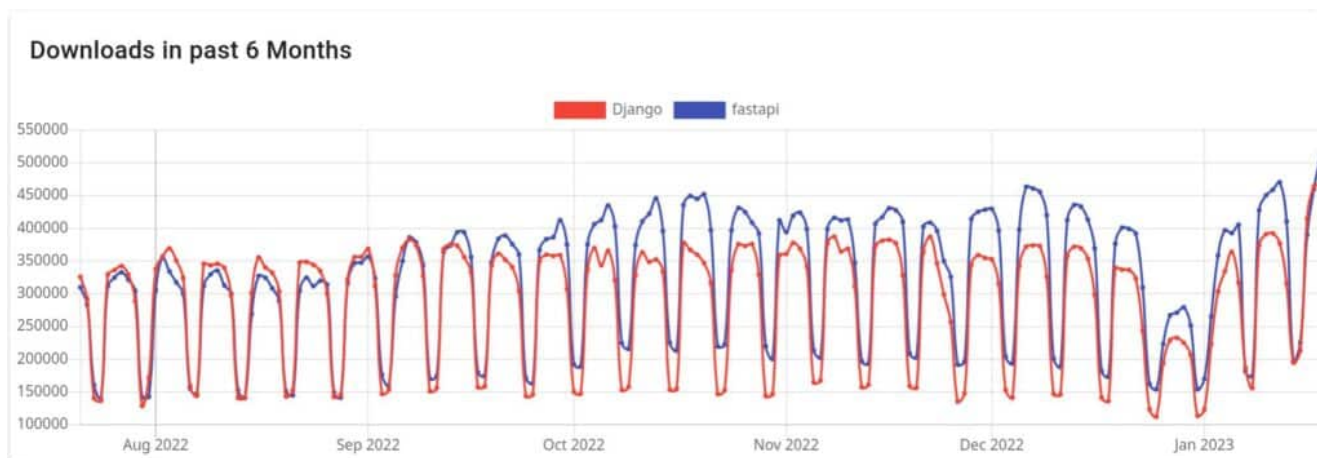
- Очень безопасный фреймворк
- Встроенная панель администратора
- Собственный язык шаблонов, который динамически отображает HTML
- Object-relational-mapper (ORM), создание базы данных с помощью классов Python
- Запрос к базе данных с помощью Python API
- Отличные инструменты CLI
- Широкий выбор расширений

FastAPI



FastAPI – это современный веб-фреймворк, который потрясающе хорошо делает одну вещь: строит API. С момента своего выхода в 2018 году он быстро набрал популярность благодаря отличной производительности и простоте. На самом деле, согласно PyPi

Stats, FastAPI имеет более 9 миллионов ежемесячных загрузок, превосходя даже такие полнофункциональные фреймворки, как Django.



Имейте в виду, что единственная цель FastAPI – создание бэкенд API. Это означает, что для отображения вашего сайта для пользователей вам необходимо иметь фронтенд-фреймворк (например, Vue.js). При этом фреймворк невероятно прост в работе, и вы даже можете развернуть приложение с FastAPI через Kinsta и ваш репозиторий на GitHub за считанные минуты.

Особенности:

- Следование открытым стандартам для API, таким как OpenAPI и схема JSON.
- Валидация и сериализация данных
- Автоматическое документирование вашего API
- Современный веб-фреймворк
- Основан на возможностях Python 3.6+, таких как объявления типов
- Валидация типов благодаря Pydantic
- Поддержка асинхронного программирования

Pyramid

Pyramid

The Start Small, Finish Big Stay Finished Framework

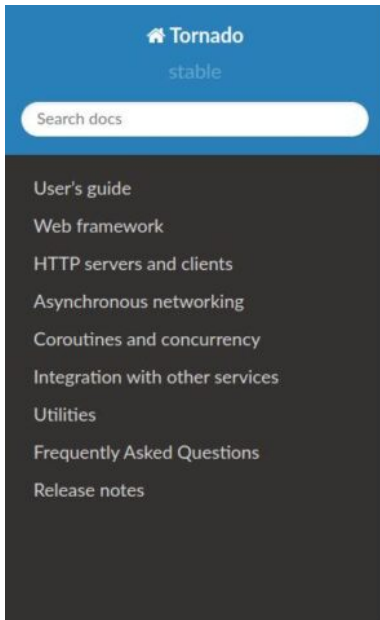
Projects with ambition start small but finish big and must stay finished.
You need a Python web framework that supports your decisions,
by artisans for artisans.

Pyramid – это гибкий и расширяемый веб-фреймворк, который хорошо подходит как для небольших, так и для крупных приложений. Это нечто среднее между микрофреймворком типа Flask и полнофункциональным фреймворком типа Django. Pyramid включает в себя наиболее распространенные функции, необходимые при разработке веб-приложений на Python, но позволяет сосредоточиться только на тех компонентах, которые нужны для вашего проекта.

Особенности:

- Легкий и гибкий
- Поддержка языка шаблонов Мако
- Простая реализация аутентификации и авторизации
- Объекты запросов и ответов, совместимые с WSGI
- Простое тестирование и отладка с помощью встроенных помощников тестирования и интерактивного отладчика
- Библиотеки и плагины сторонних разработчиков с легкой интеграцией

Tornado



Hello, world

Here is a simple "Hello, world" example web app for Tornado:

```
import asyncio

import tornado.web

class MainHandler(tornado.web.RequestHandler):
    def get(self):
        self.write("Hello, world")

def make_app():
    return tornado.web.Application([
        (r"/", MainHandler),
    ])

async def main():
    app = make_app()
    app.listen(8888)
    await asyncio.Event().wait()

if __name__ == "__main__":
    asyncio.run(main())
```

Tornado – это асинхронный веб-фреймворк с открытым исходным кодом и сетевая библиотека для создания веб-приложений с использованием Python. Первоначально он был разработан в компании FriendFeed, агрегаторе социальных сетей, который впоследствии был приобретен Facebook. Сейчас он широко используется в различных приложениях, включая веб-сервисы, аналитику в реальном времени и другие высокоскоростные приложения. Tornado оптимизирован для обработки большого количества одновременных соединений, что делает его подходящим для приложений, требующих длительного соединения с каждым пользователем (например, чат-приложения, боты и парсеры).

Особенности:

- Масштабируемость (может обрабатывать тысячи открытых соединений)
- Пользовательские шаблоны торнадо
- Встроенная аутентификация пользователей
- WebSockets и возможности длительного опроса
- Опции аутентификации сторонних производителей
- Поддержка интернационализации

Фреймворки графического интерфейса пользователя

Следующие платформы графического интерфейса пользователя (GUI) Python предоставляют широкий спектр инструментов и возможностей для создания настольных приложений, включая поддержку событийно-управляемого программирования, виджетов и графики.

PyQt



PyQt – это один из самых популярных наборов связок Python для кроссплатформенного фреймворка приложений Qt. Этот фреймворк идеально сочетает в себе простоту Python как языка общего назначения и мощную прикладную среду Qt, построенную на C++.

Информация

Связка – это API (считайте, что это мост) между двумя языками программирования. В данном случае, привязка Python для Qt означает, что вы можете использовать инструменты Qt, изначально написанные на C++, в вашем коде на Python.

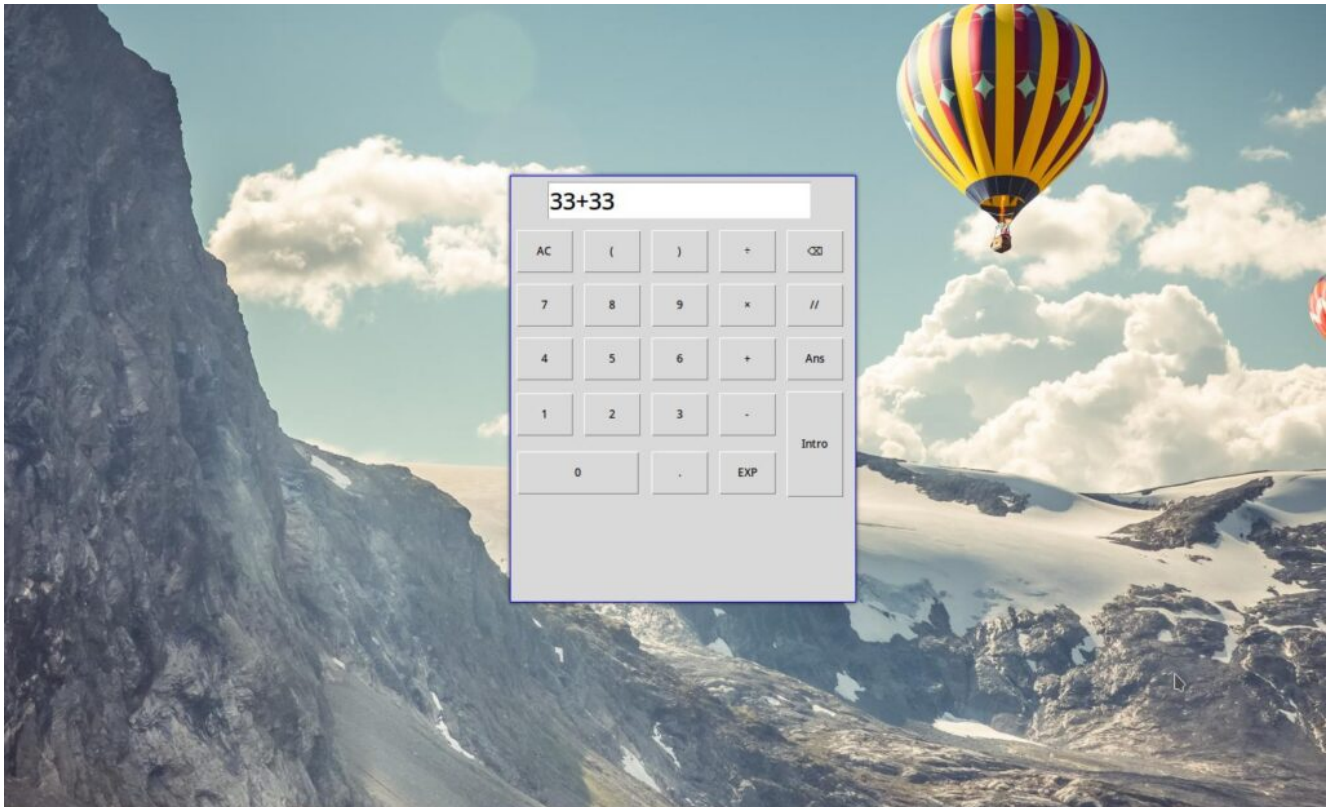
Имейте в виду, что PyQt поставляется с двумя вариантами лицензирования:

- **GPLv3:** Свободно для использования, но любой, кому вы его распространяете, может его перераспространять (вероятно, бесплатно).
- **Коммерческая:** Позволяет вам сохранить свой код в тайне, чтобы вы могли монетизировать свою работу, но вам придется платить за лицензию для каждого разработчика, который ее использует (в настоящее время \$550 на одного разработчика).

Особенности:

- Поддерживается Riverbank
- Событийно-ориентированное программирование для обработки взаимодействия с пользователем
- Интеграция с другими библиотеками и фреймворками
- Поддержка интернационализации и локализации
- Доступ к большому количеству функциональных возможностей из библиотеки Qt
- Поддержка мультимедиа, WebKit и WebEngine, интеграция баз данных
- Кросс-платформенная совместимость с Windows, Linux и macOS

Tkinter



Tkinter – это встроенный фреймворк графического интерфейса Python. Он является частью стандартной библиотеки, поэтому вам не придется устанавливать никаких дополнительных зависимостей. Подобно PyQt (который использует Qt), Tkinter использует инструментарий графического пользовательского интерфейса под названием Tk. Он также предоставляет набор инструментов для создания различных виджетов, таких как кнопки, ярлыки, текстовые поля и меню. В целом, Tkinter – это отличный способ быстро создавать небольшие программы с графическим интерфейсом пользователя, но из-за его устаревшего внешнего вида и функциональности вам лучше выбрать более мощный фреймворк для больших проектов.

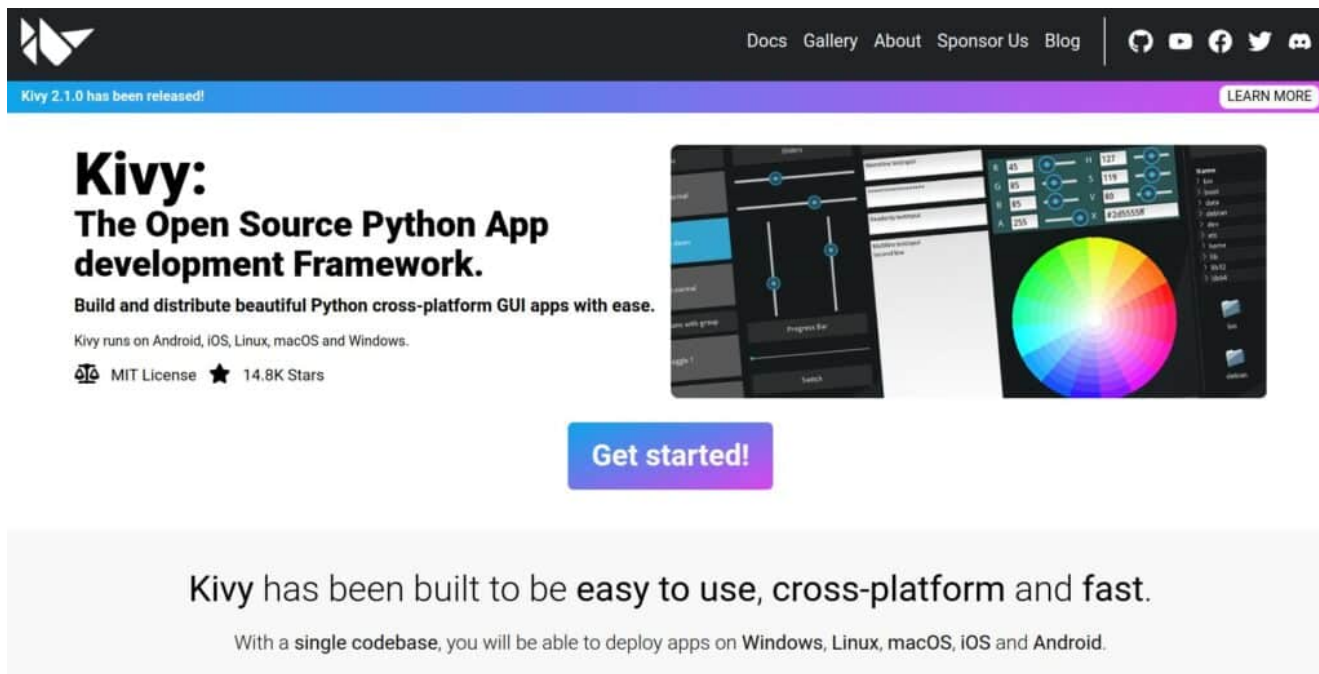
Особенности:

- Поддержка распространенных виджетов, используемых в приложениях с графическим интерфейсом.
- Событийно-ориентированное программирование для обработки взаимодействия с пользователем
- Встроенная поддержка изображений и цветов
- Кросс-платформенная совместимость с Windows, Linux и

macOS

- Часть стандартной библиотеки
- Легкий вес

Kivy



Kivy:
The Open Source Python App development Framework.

Build and distribute beautiful Python cross-platform GUI apps with ease.

Kivy runs on Android, iOS, Linux, macOS and Windows.

MIT License ★ 14.8K Stars

[Get started!](#)

Kivy has been built to be **easy to use, cross-platform and fast.**

With a single codebase, you will be able to deploy apps on Windows, Linux, macOS, iOS and Android.

Kivy – это фреймворк с открытым исходным кодом для создания кроссплатформенных мобильных и настольных приложений на языке Python. Он хорошо подходит для создания интерактивных и визуально привлекательных приложений, таких как игры и мультисенсорные приложения, а его главная фишка – единая кодовая база для всех основных платформ (Windows, Linux, macOS, iOS и Android). В отличие от PyQt, Kivy является бесплатным с открытым исходным кодом, не является оберткой вокруг внешней библиотеки пользовательского интерфейса и лицензируется по лицензии MIT, что означает, что вы можете поставлять и монетизировать свои приложения Kivy.

Особенности:

- Инструментарий с открытым исходным кодом для создания приложений с поддержкой мультитач.
- Поддержка различных платформ, включая мобильные и настольные

- Встроенная поддержка графики, анимации и элементов пользовательского интерфейса
- Возможность интеграции с другими библиотеками и фреймворками

PySide



PySide – это набор инструментов с открытым исходным кодом для создания многоплатформенных приложений. Он основан на библиотеке Qt (аналогично PyQt) и предоставляет доступ к широкому спектру функциональных возможностей. PySide также поддерживает множественные методы ввода и событийно-ориентированное программирование и включает встроенную поддержку графики и мультимедиа. Поскольку он лицензирован под LGPL, вы можете легче создавать проприетарное программное обеспечение. Он поддерживается компанией Qt и поддерживает Linux, macOS и Windows.

Особенности:

- Лицензия LGPL
- Поддерживается компанией Qt
- Поддержка отображения изображений, видео и других типов

мультимедиа в вашем приложении

- Три версии для использования с Qt4, Qt5 и Qt6 соответственно

PySimpleGUI



PySimpleGUI набирает популярность в сообществе Python благодаря своему простому и легкому в использовании API. Это отличный вариант для создания простых и удобных графических пользовательских интерфейсов в Python, позволяющий легко добавить графический интерфейс к уже работающим скриптам. PySimpleGUI объединяет в себе возможности 4 различных библиотек графического интерфейса пользователя, PySide, Tkinter, wxPython и Remi.

Особенности:

- Отличная документация
- Простой в освоении и использовании
- Полный набор готовых к использованию виджетов
- Поддержка Python 3.4+

Фреймворки машинного обучения Python

Ниже перечислены программные библиотеки Python, которые предоставляют готовые функциональные возможности и инструменты, помогающие разработчикам создавать и развертывать модели машинного обучения, поддерживая такие общие задачи, как предварительная обработка данных, обучение и оценка моделей, а также инструменты для развертывания моделей на производстве.

scikit-learn



scikit-learn – самая популярная библиотека машинного обучения, используемая как в технологической индустрии, так и в академических кругах. Она предоставляет эффективные инструменты для решения общих задач ML и построена на базе других научных библиотек, таких как NumPy и SciPy.

Особенности:

- Бесплатность и открытый исходный код
- Эффективные инструменты для добычи и анализа данных
- Предоставляет широкий спектр алгоритмов для классификации, регрессии, кластеризации и уменьшения размерности

- Активное сообщество и развитие, с хорошо документированным API и учебниками

TensorFlow



TensorFlow

TensorFlow – это платформа с открытым исходным кодом для построения и развертывания моделей машинного обучения. Разработанная командой Google Brain в 2015 году, она представляет собой мощную библиотеку для глубокого обучения с широким набором инструментов для построения и обучения нейронных сетей. TensorFlow можно использовать для решения различных задач, таких как обработка изображений и языков, распознавание речи и предиктивная аналитика.

Особенности:

- Инструменты для построения нейронных сетей
- Поддержка мобильного и веб-развертывания с помощью TensorFlow.js и TensorFlow Lite
- Инструменты для визуализации и отладки, такие как TensorBoard

- Возможность работы на различных платформах, включая CPU, GPU и TPU.
- Доступность на языках Python, C и C++

PyTorch



Наряду с TensorFlow, PyTorch (разработанный исследовательской группой ИИ Facebook) является одним из наиболее используемых инструментов для построения моделей глубокого обучения. Его можно использовать для решения различных задач, таких как компьютерное зрение, обработка естественного языка и генеративные модели.

Особенности:

- Обширная документация и большое сообщество разработчиков
- Простая интеграция с поддержкой облачных вычислений
- Легкость в освоении, удобные инструменты.
- Более питоническое восприятие, чем у других фреймворков
- Бесплатный и с открытым исходным кодом

Keras



Keras

Simple. Flexible. Powerful.

Keras – это высокоуровневый фреймворк глубокого обучения, способный работать поверх TensorFlow, Theano и CNTK. Он был разработан Франсуа Шолле в 2015 году и призван обеспечить простой и удобный интерфейс для построения и обучения моделей глубокого обучения. Keras предоставляет строительные блоки для быстрого построения моделей машинного обучения. Именно поэтому он широко используется в соревнованиях по науке о данных, таких как Kaggle.

Особенности:

- Простой и удобный интерфейс для построения и обучения моделей глубокого обучения
- Поддержка построения и обучения нейронных сетей с использованием широкого спектра архитектур и оптимизаторов
- Включает мощную экосистему библиотек и инструментов для глубокого обучения, таких как Keras Tuner и KerasRL
- Возможность работы на различных бэкендах, таких как TensorFlow, CNTK или Theano
- Поддержка распределенных вычислений
- Встроенная поддержка визуализации и отладки моделей

Механизмы научных вычислений Python

Эти фреймворки помогают студентам, ученым и исследователям выполнять свои задачи, не зацкливаясь на низкоуровневых деталях вычислений. Фреймворки для научных вычислений часто включают функциональность для линейной алгебры, оптимизации, интерполяции, интеграции и других распространенных задач в научных вычислениях.

NumPy



NumPy – это библиотека для научных вычислений, предназначенная для работы с многомерными массивами и матрицами числовых данных. Она также предоставляет широкий спектр математических функций для работы с этими массивами. NumPy является основой для многих других библиотек и фреймворков Python, используемых в науке о данных, включая SciPy, Pandas, scikit-learn и Tensorflow.

Особенности:

- Легко справляется с многомерными массивами.
- Поддержка математических инструментов, таких как процедуры линейной алгебры, преобразования Фурье и

- генерация случайных чисел
- Обширная библиотека математических функций
- Быстрее по сравнению с операциями на ванильном Python (некоторые реализации сделаны на C)
- Поддержка большого разнообразия аппаратных средств

SciPy



SciPy предоставляет набор алгоритмов и функций, построенных поверх NumPy. Он помогает выполнять общие научные и инженерные задачи, такие как оптимизация, обработка сигналов, интегрирование, линейная алгебра и многое другое.

Особенности:

- Бесплатный и с открытым исходным кодом
- Определяет алгоритмы и функции для научных и инженерных задач
- Активное сообщество и отличная документация
- Доступен каждому программисту, независимо от уровня его опыта

Pandas



Pandas – это мощная и гибкая библиотека с открытым исходным кодом, используемая для анализа данных на языке Python. Она предоставляет высокопроизводительные структуры данных (например, знаменитый DataFrame) и инструменты анализа данных, которые облегчают работу со структурированными данными.

Особенности:

- Высокопроизводительные структуры данных, такие как DataFrame и Series.
- Поддержка чтения и записи данных в и из нескольких форматов, включая CSV, Excel и SQL.
- Облегчает анализ реальных данных
- Базируется на других пакетах для работы с данными, таких как scikit-learn

Matplotlib

matplotlib

Matplotlib – это широко используемый инструмент для визуализации данных на языке Python. Он предоставляет объектно-ориентированный API для встраивания графиков в приложения. Он разработан как хорошо настраиваемый, и предоставляет широкий спектр возможностей для создания графиков, диаграмм и визуализаций. По этим причинам Matplotlib часто используется в проектах по изучению данных, машинному обучению и научным вычислениям.

Особенности:

- Поддержка 2D и 3D черчения
- Может создавать интерактивные рисунки
- Обычно встраивается в блокноты Jupyter и GUI-приложения
- Обширная документация и активное сообщество

Фреймворки для тестирования Python

Эти фреймворки призваны помочь вам в утомительном процессе тестирования вашего кода. Если вы используете разработку, ориентированную на тестирование (TDD), то следующие фреймворки просто необходимы в вашем рабочем процессе.

Pytest



Pytest – это широко распространенная среда тестирования, разработанная для простоты использования и расширения. Он помогает писать элегантные тесты как в небольших, так и в сложных кодовых базах Python.

Особенности:

- Простота в освоении и использовании
- Доступно более 800 плагинов
- Задаёт рабочий процесс для создания модульных тестов
- Подробная информация об ошибках

Unittest



Unittest – это встроенная среда тестирования Python, основанная на соглашениях тестирования xUnit. Он обеспечивает питонический и объектно-ориентированный способ создания автоматизированных тестовых случаев и костюмов тестов.

Особенности:

- Простой и удобный интерфейс для написания и запуска тестов
- Тестирование параметризации, фикстур и маркеров
- Может работать в сочетании с pytest или nose
- Часть стандартной библиотеки Python

nose2

nose

is nicer testing for python

nose2 является преемником фреймворка тестирования nose, основное отличие в том, что nose2 поддерживает современные версии Python. nose2 расширяет встроенную библиотеку unittest и предоставляет более мощный и гибкий способ написания и запуска тестов. Это расширяемый инструмент, поэтому вы можете использовать множество встроенных и сторонних плагинов в своих интересах.

Особенности:

- Расширяемость с помощью плагинов
- Основан на встроенном unittest
- Поддержка Python 3

Асинхронные фреймворки

Эти типы фреймворков Python предоставляют набор инструментов и библиотек, которые позволяют разработчикам писать асинхронный код более управляемым и эффективным способом.

Информация

Написание асинхронного кода позволяет выполнять определенные задачи параллельно, не блокируя выполнение других задач.

asyncio



asyncio лежит в основе многих других асинхронных фреймворков. Он позволяет разработчикам писать параллельный код, используя синтаксис `async/await`, и рассчитан на обработку тысяч параллельных соединений. asyncio предоставляет единый API для нескольких транспортных протоколов, таких как TCP, UDP, SSL/TLS, и взаимодействия подпроцессов.

Особенности:

- Встроен в Python
- База для нескольких других фреймворков Python
- Включает API для управления подпроцессами и выполнения сетевого ввода-вывода
- Отличная документация
- Часть стандартной библиотеки Python

Aiohttp



Aiohttp

Aiohttp – это асинхронный клиент-серверный фреймворк HTTP, построенный на основе пакета `asyncio`. Он предоставляет простой и интуитивно понятный интерфейс для обработки HTTP-запросов и ответов, а также поддержку промежуточного ПО, сессий и других инструментов веб-разработки.

Особенности:

- Протокол HTTP на стороне клиента и на стороне сервера
- Обработывает большое количество одновременных соединений
- Позволяет создавать асинхронные веб-приложения

Заключение

Фреймворки Python помогают улучшить организацию кода, повысить производительность и сделать процесс разработки программного обеспечения более простым и эффективным. Ваша задача, как разработчика Python, заключается не в том, чтобы стать экспертом по всем 25 фреймворкам Python, которые мы представили выше, а в том, чтобы тщательно отобрать те, которые вас больше всего интересуют, создать с их помощью проекты, которые вас действительно увлекают, и полностью освоить один или два фреймворка.