

Разработка программного обеспечения на языке Python

[Обзорная панель](#) ▶ [Мои курсы](#) ▶ [Разработка ПО на языке Python](#) ▶ [Веб-программирование на Python](#) ▶

[Лекция 2. Библиотека Django](#)

Лекция 2. Библиотека Django

Посмотрите видеоуроки и ответьте на контрольные вопросы после лекции

Django



**Don't repeat
yourself**

DRY («Не повторяйся»)

00:00 / 04:33



В данном разделе начнем рассматривать один из основных фреймворков для написания веб-приложений на python. Django - это фреймворк для создания веб-приложений с помощью языка программирования Python. Он используется во многих продуктах.

Как и любой другой фреймворк, «Джанго» – это программная платформа (каркас), к которому разработчик «подключает» дополнительные компоненты приложения. Например, интерфейс пользователя; программные библиотеки; мультимедийный контент (фото, видео, аудио); базы данных и т.д.. Фреймворк отвечает за базовые аспекты работы продукта. А подключаемые модули – за специфичные функции конкретного приложения. Использование фреймворков позволяет: ускорить разработку, упростить поддержку и избежать ошибок в работе приложения.

Django был создан в 2005 году, когда веб-разработчики из онлайн-версии газеты Lawrence Journal-World стали использовать Python в качестве языка для создания веб-сайтов. В 2008 году вышел публичный первый релиз фреймворка. На сегодняшний день он продолжает развиваться.

Так, текущей актуальной версией фреймворка является 4.0, которая вышла в декабре 2021 года. Также регулярно выходят подверсии.

Django довольно популярен. Он используется на многих сайтах, в том числе таких, как Pinterest, YouTube, Google, Mozilla и многих других. Фреймворк является бесплатным. Он развивается как open source, его исходный код открыт.

Сайт на Django строится из одного или нескольких приложений, которые рекомендуется делать отчуждаемыми и подключаемыми. Это одно из существенных архитектурных отличий этого фреймворка. Также, в отличие от других фреймворков, обработчики маршрутов URL в Django конфигурируются явно при помощи регулярных выражений. Для

работы с базой данных Django использует собственный ORM, в котором модель данных описывается классами Python, и по ней генерируется схема базы данных. Данный инструмент позволяет работать с популярными СУБД (например, sqlite и PostgreSQL).

Один из основных принципов фреймворка – DRY (англ. Don't repeat yourself). Для реализации аналогичных или схожих задач используются одни и те же элементы кода, а не пишутся каждый раз новые. Благодаря этому отсутствует дублирование, усложняющее приложения. Например, вам не нужно разрабатывать свою панель администратора, достаточно лишь настроить эту панель у Django. Это сильно отличает его от Flask, где программисту дается большая свобода выбора на начальном этапе.

На Django можно создавать широкий диапазон веб-приложений: от небольших персональных сайтов до высоконагруженных сложных веб-сервисов. Можно выделить следующие преимущества использования данного фреймворка.

- **Масштабируемость.** Архитектура Django строится на независимости составляющих частей. Любой компонент можно заменить или модифицировать, не затрагивая другие.
- **Универсальность.** Django подходит для разработки веб-сайтов и приложений любого типа.
- **Полнота.** Разработчик может создать веб-приложение без использования сторонних компонентов.
- **Сопровождаемость.** Django позволяет повторно использовать код, группировать связанные функции в отдельные модули.
- **Гибкость** - предоставляется несколько инструментов для решения одной задачи.

Отметим еще некоторые особенности фреймворка. Монолитная архитектура. «Джанго» развивается как единый комплекс инструментов разработчика. Это позволяет реализовывать даже крупные проекты, не обращаясь к сторонним приложениям. У Django есть свой менеджер базы данных ORM, которая позволяет работать с различными типами БД. Отсутствие многозадачности (синхронность) – «Джанго» по умолчанию не позволяет отдельным процессам работать с несколькими запросами одновременно. Эта часть фреймворка развивается.

Таким образом, монолитность Django имеет как плюсы, так и минусы, что следует учитывать при выборе данного фреймворка. Django следует использовать для более-менее крупных проектов, где его функционал действительно востребован. Для небольших проектов его функционал может быть избыточен или недостаточно гибок. Например, если вы хотите быстро сделать типовой gest-сервис, можно использовать Django вместе с его библиотекой Django REST Framework. Такое решение будет более оптимальным, чем делать свой сервис на Flask с нуля.

Создание проекта Django

ПРЕДЫДУЩИЙ ЭЛЕМЕНТ КУРСА

◀ [Задание 2. Создание flask приложения](#)

Перейти на...

СЛЕДУЮЩИЙ ЭЛЕМЕНТ КУРСА

[Задание 3. Создание своего первого сайта на Django](#) ►

Контакты +7(391) 206-27-05
info-ms@sfu-kras.ru

[Скачать мобильное приложение](#)

[Инструкции по работе в системе](#)