

# Разработка программного обеспечения на языке Python

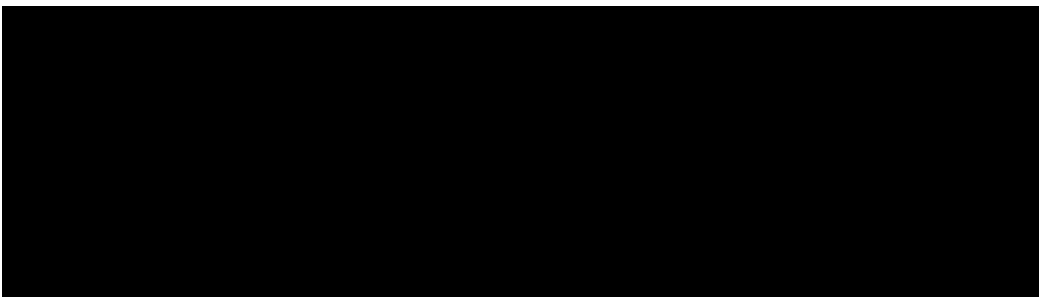
[Обзорная панель](#) ▶ [Мои курсы](#) ▶ [Разработка ПО на языке Python](#) ▶ [Веб-программирование на Python](#) ▶

[Лекция 1. Концепции web-приложения](#)

## Лекция 1. Концепции web-приложения

Посмотрите видеоуроки и ответьте на контрольные вопросы после лекции

### Web. Концепции



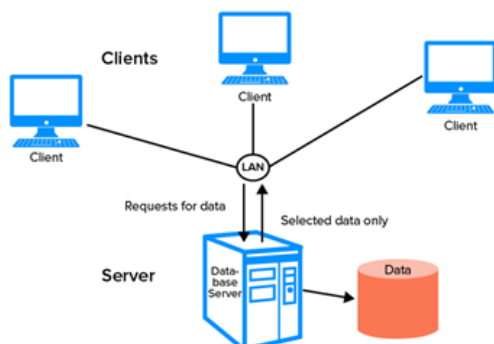
00:00 / 05:07



Веб-разработка в широком смысле подразумевает создание и обслуживание веб-сайта или веб-приложения. Веб-приложение — клиент-серверное приложение, в котором клиент взаимодействует с веб-сервером при помощи браузера.

Основными этапами процесса являются веб-дизайн, вёрстка страниц, программирование на стороне клиента и сервера, а также конфигурирование веб-сервера.

Логика веб-приложения распределена между серверной и клиентской частью, тем самым реализуется технология «клиент-сервер».



Клиентская часть реализует пользовательский интерфейс, формирует запросы к серверу и обрабатывает ответы от него. Серверная часть получает запрос от клиента, выполняет вычисления, после этого формирует веб-страницу и отправляет её клиенту по сети с использованием протокола HTTP.

Языки и технологии программирования, которые в предназначены для работы с веб-приложениями, можно условно разделить на две пересекающиеся группы: клиентские (относящиеся к front-end) и серверные (к back-end).

Под **Frontend** понимается разработка пользовательского интерфейса и функций, которые работают на клиентской стороне веб-сайта или приложения. Это всё, что видит пользователь, открывая веб-страницу, и с чем он взаимодействует.

Компоненты frontend-разработки включают такие технологии, как CSS, HTML и JavaScript – то есть всё то, что отвечает за отображение контента на странице вашего браузера. CSS и HTML отвечают за верстку – стили и расположение элементов. Язык программирования JavaScript позволяет управлять такими элементами на странице.

**Backend**-разработка — это набор аппаратно-программных средств, при помощи которых реализована логика работы сайта, то есть реализация его серверной части.

Это подключения к базам данных, предоставление их клиенту, управление пользователями и т.д.

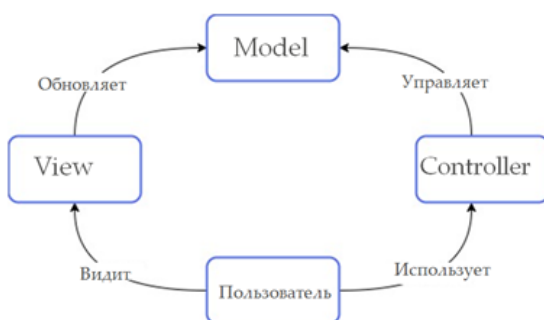
Когда пользователь дает запрос на какую-либо страницу (переходит на неё по ссылке или вводит адрес в адресной строке своего браузера), то вызванная страница сначала обрабатывается на сервере, то есть выполняются все программы, обрабатывающие введенный запрос, и только потом возвращается клиенту. Для разработки сервера могут быть использованы разные языки программирования, и один из них как раз Python.

**Фреймворк** веб-приложений — программная платформа, определяющая структуру программной системы. Это каркас программной системы, в данном случае, веб-приложения. Он является инструментом, облегчающим разработку и объединение разных компонентов большого программного проекта. Многие фреймворки упрощают доступ к базам данных, разработку интерфейса, и также уменьшают дублирование кода. Примеры веб-фреймворков, написанных на различных языках программирования приведены на слайде.

В Python Web-framework'ом является библиотека, которая позволяет разрабатывать и запускать веб-приложение, содержащее в качестве back-end код на Python.

В зависимости от масштаба приложения, следует выбирать разный веб-фреймворк. Так называемые микрофреймворки подходят для небольших проектов. Они легкие, имеют простую структуру и в случае чего легко масштабируются. К таким фреймворкам относятся, например, Flask и Bottle. Примером полноценного фреймворка является популярный фреймворк Django. Он более сложен в настройке, но имеет и более богатый функционал.

Поговорим про устройство фреймворков. Не только в Python, но и в других языках, для написания веб-приложений используется такой паттерн как MVC.



MVC расшифровывается как Model-View-Controller. Приложение делится на три слоя:

- первый слой – Model – отвечает за доступ к данным;
- второй слой – View – отвечает за отображение данных (к примеру, это может быть HTML-страница);
- третий слой – Controller – осуществляет связь между моделью и представлением.

Паттерн MVC может быть реализован по-разному. Например, стандартным является подход, когда контроллер принимает пользовательский ввод, отправляет изменения модели, а представление уже реагирует на эти изменения и отображает новую информацию.

Также существуют разные другие вариации архитектурных паттернов, описывающих принцип организации кода в приложениях: MVP, MVVM, MTV. Последний используется в Django, и мы поговорим про него подробнее на следующих лекциях.

В нашем курсе будем изучать принципы разработки веб-приложений с применением фреймворков на Python.

[Основы html](#)

#### ПРЕДЫДУЩИЙ ЭЛЕМЕНТ КУРСА

◀ [Запись синхронных занятий по курсу "Веб-программирование на Python"](#)

Перейти на...

#### СЛЕДУЮЩИЙ ЭЛЕМЕНТ КУРСА

[Задание 1. Дизайн страницы приложения](#) ▶

© 2010-2023 Центр обучающих систем  
Сибирского федерального университета, sfu-kras.ru

Разработано на платформе moodle  
Beta-version (3.9.1.5.w3)

[Политика конфиденциальности](#)

[Соглашение о Персональных данных](#)

[Политика допустимого использования](#)

**Контакты** +7(391) 206-27-05  
[info-ms@sfu-kras.ru](mailto:info-ms@sfu-kras.ru)

[Скачать мобильное приложение](#)

[Инструкции по работе в системе](#)