

Followup к первому занятию

Что такое бэкенд и фронтенд

Приложений, которые существуют только на сервере или только на клиенте (мобильный телефон, браузер), очень мало. Большинство приложений имеют две части - на сервере есть какая-то бизнес-логика приложения, там хранятся данные и происходят какие-то расчёты, а на клиенте это все отображается удобным для пользователя способом.

Для примера возьмем компанию Airbnb. У них есть бэкенд, который работает на их серверах и три вида фронтенда: веб-интерфейс для браузера, приложение для Android и приложение для iOS. Все эти три приложения на самом деле взаимодействуют с одним и тем же бэкендом. Каждый из этих интерфейсов получает данные со списками стран, городов и доступных фильтров для того, чтобы показать пользователю наполненный вариантами выбора интерфейс, а когда пользователь вводит параметры поиска, клиент передает эти параметры на все тот же сервер, который и производит поиск квартир, а потом передает список этих квартир назад на клиент, который отображает их пользователю.

Какие языки для чего используются

Еще раз подчеркну, что это очень условный список и единственная его задача - нарисовать в твоей голове очень общую картину происходящего на рынке.

- **C/C++** - разработка всего, что требует очень экономного и эффективного использования ресурсов компьютера - операционных систем, игр с крутой графикой вроде какого-нибудь Cyberpunk 2077, программ для embedded систем (от железа торговых автоматов по продаже жвачки до датчиков сигнализаций).
- **C# (.NET)** - большой корпоративный бэкенд, а также игры, написанные с использованием движка Unity.
- **Java** - большой корпоративный бэкенд (часто связанный с финансами) и разработка приложений для Android.
- **Kotlin** - современная замена Java для разработки приложений Android (но знание Java от Android-разработчиков все равно продолжают ждать).
- **Swift/Objective C** - языки для разработки приложений для macOS и iOS.
- **PHP** - разработка сайтов всех уровней сложности. Сейчас это современный и совершенно нормальный язык, который до сих пор имеет не очень хорошую репутацию из-за грехов своих старых версий. Надо сказать, что большая часть работы на рынке PHP разработки - это до сих пор "сделай говносайт на коленке - за 50 долларов".
- **Python** - в первую очередь это язык, который используется во всех направлениях **data science** и **machine learning**. Но он также часто используется для разработки бэкенда разной степени сложности. Язык, который нужно знать практически всем, если не как основной, то хотя бы как второй из-за его удобства для автоматизации ежедневных задач.

Чем мы будем заниматься в ближайшее время

Мы будем учиться писать довольно сложный бэкенд на языке Python. Для этого нам понадобится узнать о том, как работает довольно большой набор сопутствующих технологий. До нового года мы будем проходить: - Основы работы с командной оболочкой - Основы программирования, общие для всех современных императивных языков - Основы Python, его модель данных и идиоматику - Базовые понятия алгоритмов и структур данных

Домашнее задание

Установить Windows Subsystem for Linux

1. Посмотри часть [видео](#) и следуй инструкциям, чтобы установить WSL2. Тебя интересует часть видео с начала и до пятой минуты. Потом начинается разбор того, как установить другие дистрибутивы, а это нам пока не нужно.
2. Открой окно WSL и введи команду `sudo apt-get update`, дождись обновления (пока снова не появится приглашение к вводу).
3. Попробуй ввести команду `python3` и убедись, что она открывает REPL питона. Выйди из него командой `exit()`
4. Теперь тебе доступна полноценная командная оболочка и все команды в ней работают в точности так, как они будут работать в настоящем Linux окружении.
5. Ну и напоследок немного терминологии. В Linux мы называем то, что в Windows называлось *командной*

строкой, командной оболочкой (*shell*).

Самостоятельное чтение

1. [Концептуальные различия между файловыми системами Windows и Linux.](#)
2. [34 Команды Linux, Которые Должен Знать Каждый Пользователь.](#) С ними можно поиграть внутри командной оболочки WSL.