

# Введение в Javascript

Frontend. Урок 16

# Вопросы на повторение

1. Какими методами можно оптимизировать веб-сайт?
2. Как обеспечить отложенную загрузку элементов веб-сайта?
3. Что такое минификация файлов при оптимизации веб-сайта?
4. Как измерить производительность сайта?

# Постановка проблемы

На данном этапе наши web-страницы уже могут хорошо выглядеть, но сейчас они - только обертка без какой-либо логики - не работают кнопки, поля ввода и т.д.

Для реализации работы элементов и страниц нам необходимо что-то большее, чем стилизация компонентов.

# JavaScript

1. Для решения проблемы описания логики компонентов был придуман язык сценариев (скриптов) **JavaScript**. В отличие от полноценных языков программирования JS не может существовать вне хоста - т.е. браузера или серверного окружения. Т.е. нельзя написать программу, как на Python, например, и запустить её из консоли. Также нельзя написать полноценное приложения не прибегая к использованию браузера (Все известные браузеры написаны на C и C++).

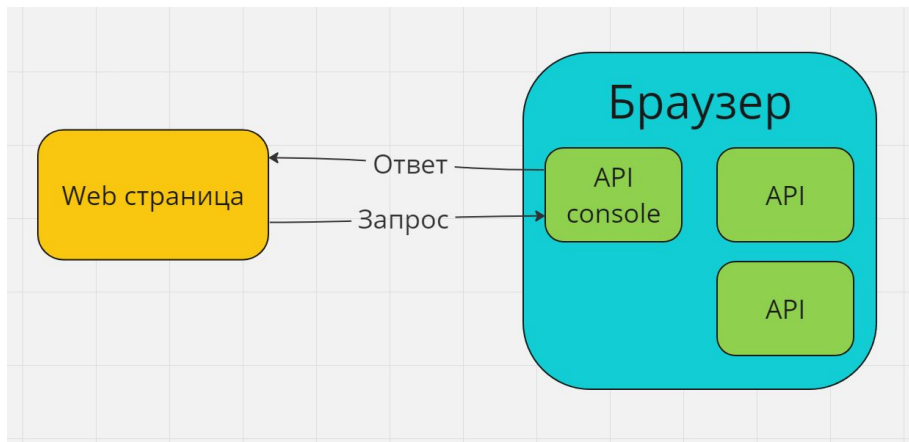
# Консоль в JavaScript

Мы уже выяснили, что JS это сценарный язык и он отличается от языков программирования. Одним из таких отличий является способ взаимодействия с пользователем. Если в языках программирования вы можете напрямую написать команды ввода и вывода данных (print, read и т. п.), то в JS язык не умеет самостоятельно читать и выводить данные, для этого были разработаны специальные API для работы с вводом-выводом.

# API

API (Application Programming Interface) — программный интерфейс, то есть описание способов взаимодействия одной компьютерной программы с другими.

В нашем случае (код выше) console как раз таки и является API для взаимодействия с выводом информации. Это API позволяет браузеру взаимодействовать с кодом JS. Таких API очень много, они упрощают жизнь разработчиков и делают код менее громоздким.



# Консоль в JavaScript

Консоль позволяет на выводить какие-либо данные в нужном нам формате - например, `console.log()` выводит лог(можно сравнить с записью из журнала). Также у этого API еще много разных функций, с которыми познакомимся позже.

Чтение данных.

Также все современные браузеры предоставляют нам функции взаимодействия такие как:

- `alert`
- `prompt`
- `confirm`

# Консоль в JavaScript

## **alert**

```
alert("Привет! Хорошая погода сегодня!")
```

alert просто выводим сообщение в всплывающем окошке (модальном окне).

## **prompt**

```
alert("Привет, "+prompt("Введите ваше имя..."))
```

prompt позволяет вводить данные в поле ввода и возвращает их, как результат выполнения.

## **confirm**

```
confirm("Вы точно уверены?")
```

confirm спрашивает нашего разрешения, пока что непонятно где его можно применить, но в дальнейшем, мы этому научимся!



# Практика А

Начало работы над мессенджером.

Верстаем html-содержимое сайта (для мобильных устройств), CSS делать не надо

Ссылка:

<https://www.figma.com/file/jzXEpB10oA8j9dVMVnELsc/%D0%9C%D0%B5%D1%81%D1%81%D0%B5%D0%BD%D0%B4%D0%B6%D0%B5%D1%80-1.0?type=design&node-id=0-1&mode=design&t=grPxleoDXFKOMHGJ-0>

# Практика В

Используя функции взаимодействия выведите в консоль заполненную форму регистрации ФИО + почта.