# Добавление динамики к сайтам

1. Добавление динамики в сайтам возможно через интеграцию скриптов в HTML код, которые реагируют на действия пользователя (клики, клавиатура, положение мыши) и изменяют html код, основываясь на этих действиях.
2. 1
3. Подавляющее большинство современного фронтенд- написано на JavaScript (и его надмножествах, как TypeScript).

основные причины рассмотрены в JavaScript

# JavaScript

Подавляющее большинство современного фронтенд-разработки написано на JavaScript (и его надмножествах, как TypeScript).

основные причины рассмотрены ниже

## Монополия браузеров

JavaScript — единственный язык программирования, который понимают все основные браузеры (Chrome, Firefox, Safari, Edge) нативно, без дополнительных преобразований.

Браузеры предоставляют мощные API для работы со страницей (так называемый DOM — Document Object Model), и доступ к этим API возможен только из JavaScript.

Это исторически сложившийся стандарт. JavaScript был создан специально для оживления веб-страниц в Netscape Navigator в 1995 году и с тех пор стал неотъемлемой частью веб-стандартов (спецификаций W3C и ECMA).

## Прямой доступ к DOM

JavaScript позволяет напрямую манипулировать содержимым, стилями и структурой веб-страницы в реальном времени:

Изменять текст, картинки.

Добавлять или удалять элементы.

Менять CSS-классы, создавая анимации и динамические эффекты.

Реагировать на действия пользователя (клики, нажатия клавиш, движения мыши).

Никакой другой язык не может сделать это напрямую в браузере.

## Эволюция языка и экосистемы

JavaScript не остался в 1995 году. Он прошел огромный путь развития:

ES6 (2015) и более поздние стандарты (ES2020+): Появление классов, модулей, стрелочных функций, async/await сделало язык современным, мощным и удобным для разработки сложных приложений.

TypeScript: Язык от Microsoft, который строго типизирован и компилируется в JavaScript. Он решает главную "болевую точку" JS — слабую типизацию — и стал промышленным стандартом для больших проектов, но на выходе всё равно дает JavaScript.

Огромная экосистема (npm): У JavaScript самый большой в мире репозиторий бесплатных библиотек и пакетов (npm). Любую задачу, от слайдера до сложной карты, можно решить, установив готовый пакет.

## Появление мощных фреймворков и библиотек

JavaScript-сообщество создало невероятно мощные инструменты для построения сложных одностраничных приложений (SPA):

React, Vue.js, Angular, Svelte — эти фреймворки и библиотеки позволяют структурировать код, разбивать интерфейс на компоненты и эффективно управлять состоянием приложения. Они все написаны на JavaScript (или TypeScript) и компилируются в него.

# Другие языки скриптов

Да, но они либо провалились, либо остались нишевыми решениями:

Flash (ActionScript): Был невероятно популярен для интерактивного контента и игр, но был закрыт из-за проблем с безопасностью, производительностью и отсутствием мобильной поддержки. Его смерть окончательно утвердила власть JS.

WebAssembly (Wasm): Это не язык, а бинарный формат для браузера. Он позволяет запускать в браузере код, написанный на C++, Rust, Go и других языках, с производительностью, близкой к нативной.

Но ключевой момент: WebAssembly не заменяет JavaScript, а дополняет его. Он идеален для сложных вычислений (игры, симуляции, редакторы), но для работы с DOM и браузерными API ему всё ещё нужен JavaScript. Они работают в тандеме.

Dart: Язык от Google, который компилируется в JavaScript. Нашел свою основную нишу как язык для фреймворка Flutter, но не смог составить конкуренцию JS в вебе.

CoffeeScript, Elm и др.: Были попытки создать языки, которые компилируются в "более красивый" JS. Но с развитием самого JavaScript (особенно после ES6) их необходимость практически исчезла.

# Разработка JS скриптов

1. Разработка JS скриптов расмотрено в Прогр. инженерия\Web приложения\Front end\ JS Frontend разработка