

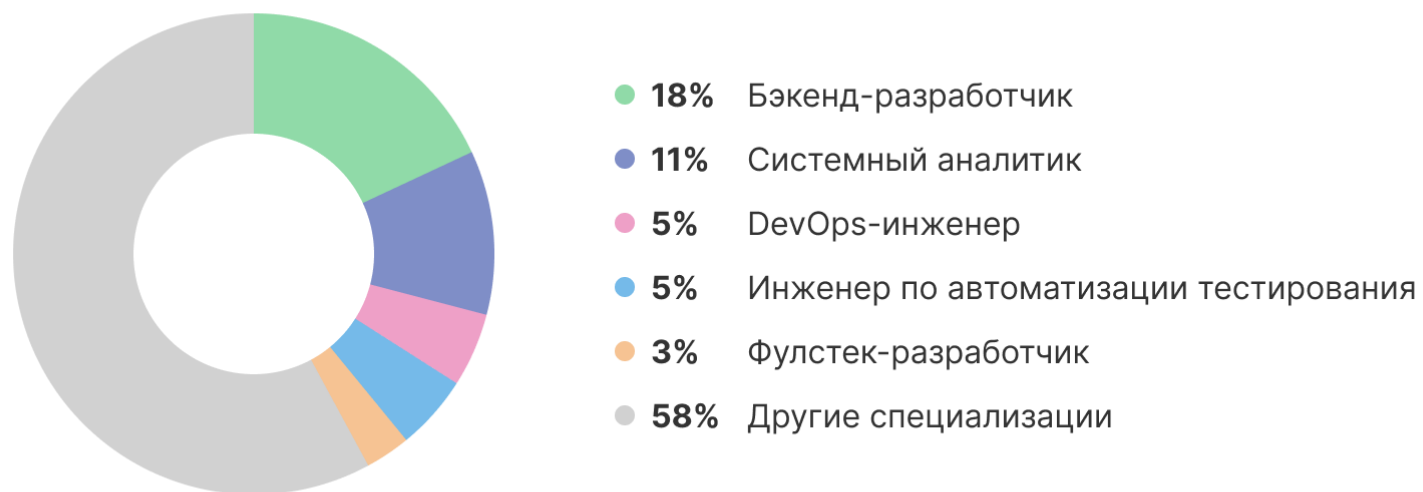
# Веб-программирование

Вводное занятие

О современной веб-разработке

# Веб-разработка в 2025

Топ 5 специализаций, которые искали в 4кв 2024













Активность найма на IT-рынке в 4 квартале 2024

Хабр Карьера

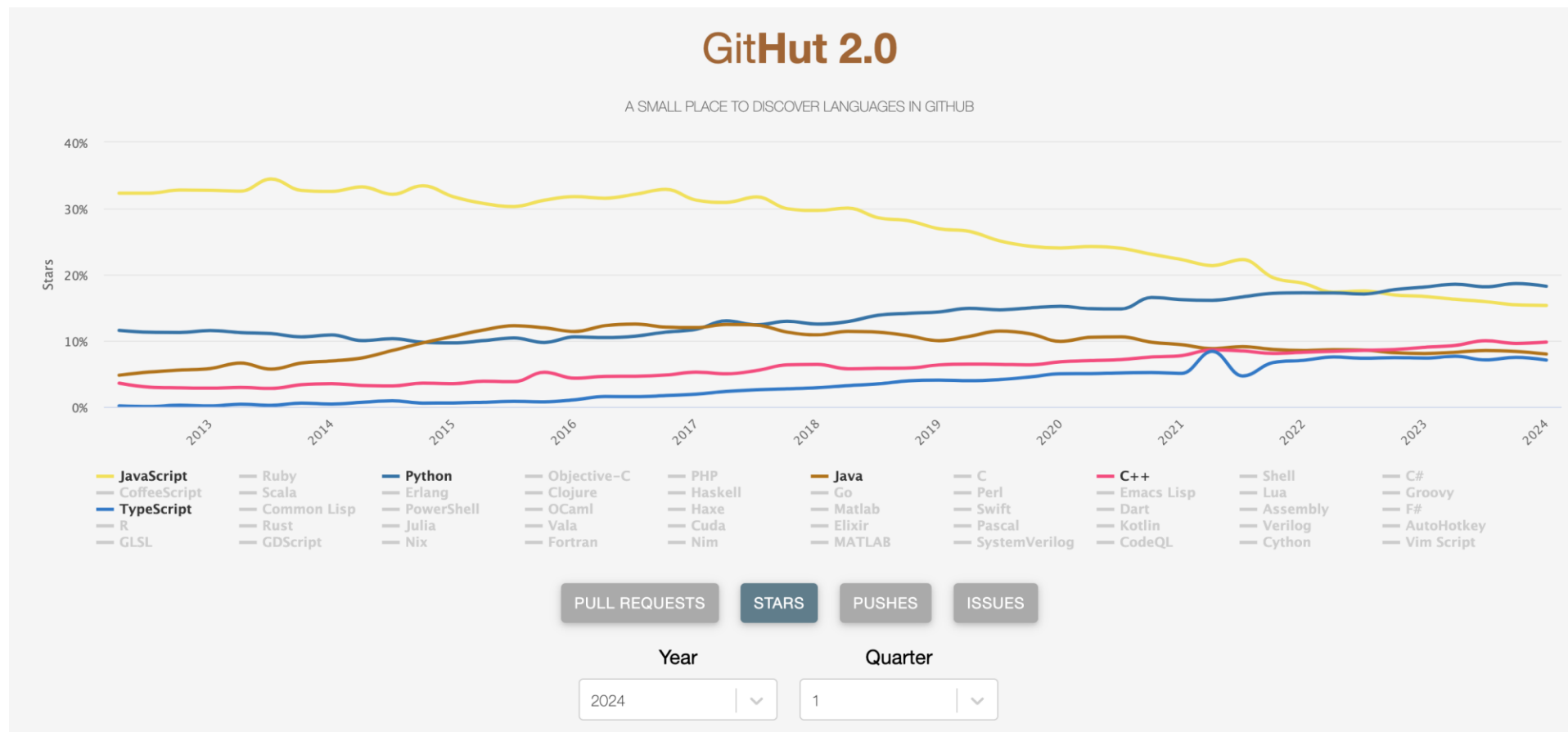
[https://habr.com/ru/companies/habr\\_career/articles/875024/](https://habr.com/ru/companies/habr_career/articles/875024/)

# Веб-разработка в 2025

Feb 2025	Feb 2024	Change	Programming Language		Ratings	Change
1	1			Python	23.88%	+8.72%
2	3	▲		C++	11.37%	+0.84%
3	4	▲		Java	10.66%	+1.79%
4	2	▼		C	9.84%	-1.14%
5	5			C#	4.12%	-3.41%
6	6			JavaScript	3.78%	+0.61%
7	7			SQL	2.87%	+1.04%
8	8			Go	2.26%	+0.53%
9	12	▲		Delphi/Object Pascal	2.18%	+0.78%
10	9	▼		Visual Basic	2.04%	+0.52%

<https://www.tiobe.com/tiobe-index/>

# Веб-разработка в 2025



<https://madnight.github.io/github/#!/stars/2024/1>

О курсе

# Структура курса

## Занятие 1

- «Современная веб-разработка»
- ДЗ: найти макеты и познакомиться с HTML, CSS

## Занятие 2

- «Design to code» — верстаем по макетам
- ДЗ: сверстать своё приложение

# Структура курса

## **Занятие 3**

- «Базовый JavaScript»
- ДЗ: сделать задачи на JavaScript

## **Занятие 4**

- «Введение в React»
- ДЗ: перенести вёрстку на React



# Структура курса

## **Занятие 5**

- «Клиент, сервер, API»
- ДЗ: подключить API к приложению

## **Занятие 6**

- «Облака и хостинги»
- ДЗ: развернуть приложение в облаке

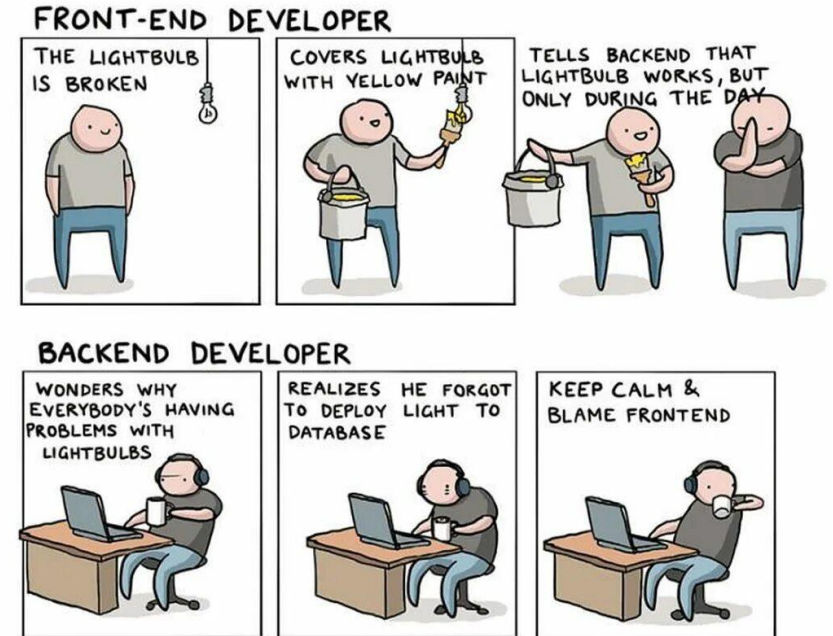
# Структура курса

## Остальные занятия

- «Веб и машинное обучение»
- «TypeScript»
- «Доступность интерфейсов»
- «Безопасность веб-приложений»

# Что даст курс?

- Познакомитесь с тем, как создаются современные веб-приложения
- Узнаете, чем занимаются веб-разработчики
- Разработаете своё собственное веб-приложение



# Что даст курс?



<https://neurorama.press/>

# Как сдать курс?

- «План А» — делаем домашние задания, набираем баллы
- «План Б» — сразу сдаём полноценный проект

# FAQ

**A.** Где найти домашние задания?

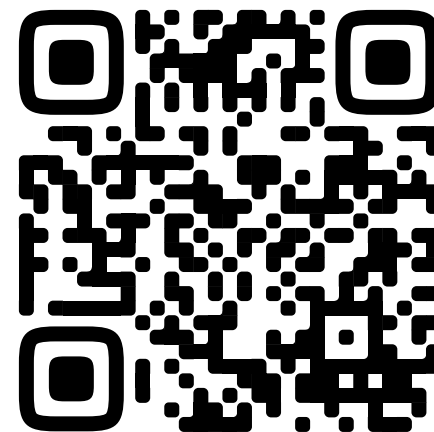
**Q.** В репозитории курса на GitHub

**A.** В каком формате сдавать задания?

**Q.** В виде ссылки на GitHub.

**A.** Есть ли дедлайны ДЗ? Есть ли штрафы?

**Q.** Нет и нет



<https://github.com/lyaplyap/frontend-development-course>

Что такое фронтенд?

# Ретроспектива

How it started

```
RETRO
├── howitsgoing
├── howitstarted
│   ├── index.html
│   ├── script.js
│   └── styles.css
└── ...

howitstarted > <> index.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <title>My web app</title>
6     <link rel="stylesheet" href="styles.css">
7     <script src="script.js">
8   </head>
9   <body>
10    <div>Hello, world!</div>
11  </body>
12 </html>
```

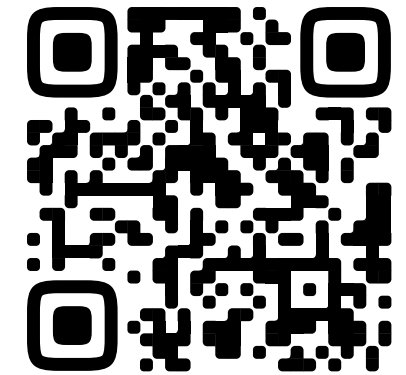
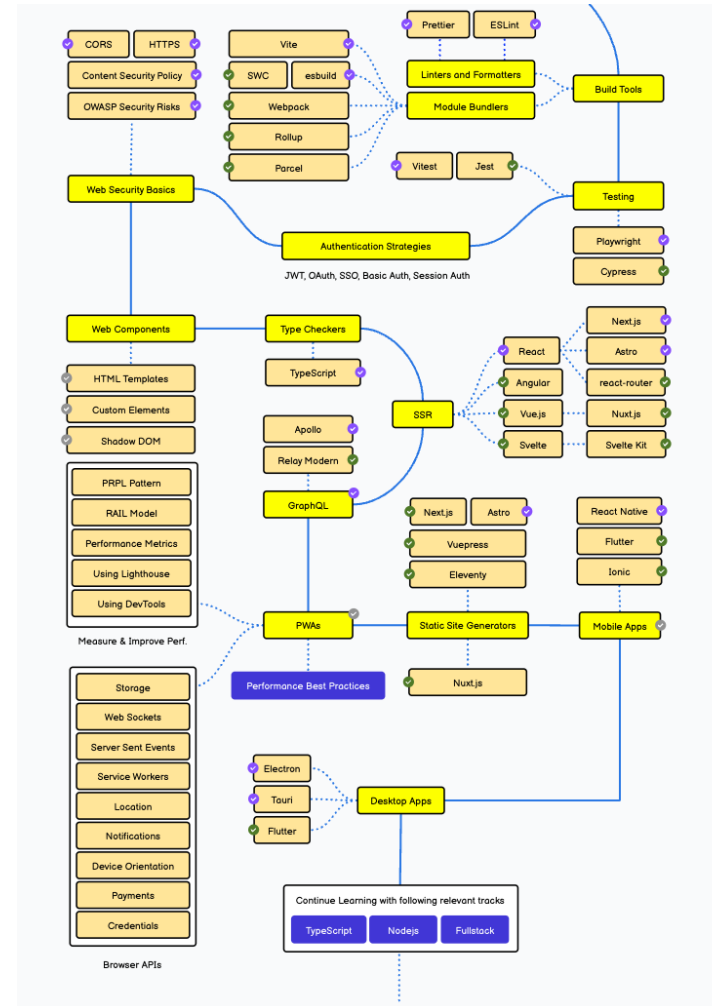
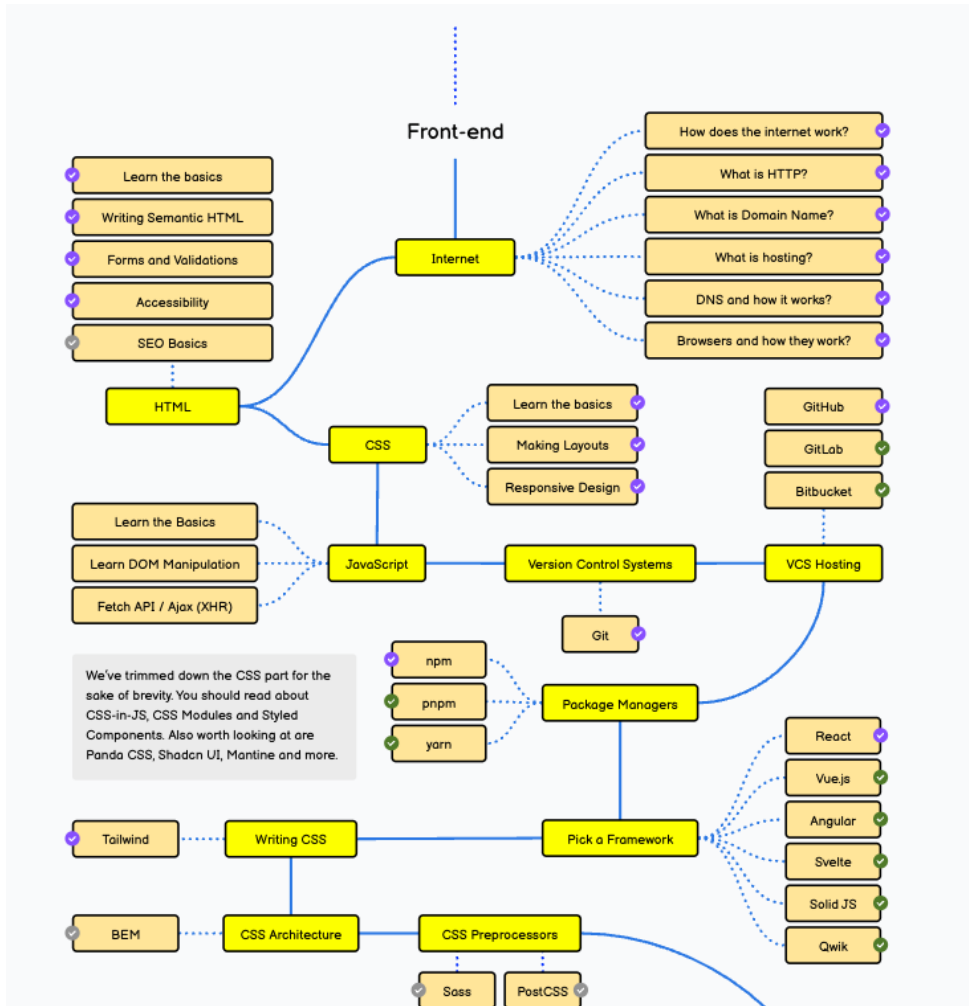
How it's going

```
RETRO
├── howitsgoing
│   ├── node_modules
│   ├── public
│   │   └── vite.svg
│   └── src
│       ├── assets
│       ├── App.css
│       ├── App.tsx
│       ├── index.css
│       ├── main.tsx
│       ├── vite-env.d.ts
│       ├── .gitignore
│       ├── eslint.config.js
│       ├── index.html
│       ├── package-lock.json
│       ├── package.json
│       ├── README.md
│       ├── tsconfig.app.json
│       ├── tsconfig.json
│       ├── tsconfig.node.json
│       └── vite.config.ts
└── ...

howitsgoing > src > App.tsx
1 import { useState } from 'react'
2 import { useReducer } from 'react'
3 import { viteLogo } from './assets'
4 import './App.css'
5
6 function App() {
7   const [count, setCount] = useState(0)
8
9   return (
10     <div>
11       <a href="#home">Home</a>
12       <a href="#about">About</a>
13       <a href="#contact">Contact</a>
14       <img alt="Vite logo" data-bbox="100 100 150 150"/>
15       <a href="#docs">Docs</a>
16       <img alt="Vite logo" data-bbox="100 100 150 150"/>
17     </div>
18     <h1>Vite + React</h1>
19     <div class="counter">
20       <button>Count</button>
21       <button>Reset</button>
22     </div>
23     <p>Edit src/App.tsx</p>
24   )
25 }
26 export default App
```

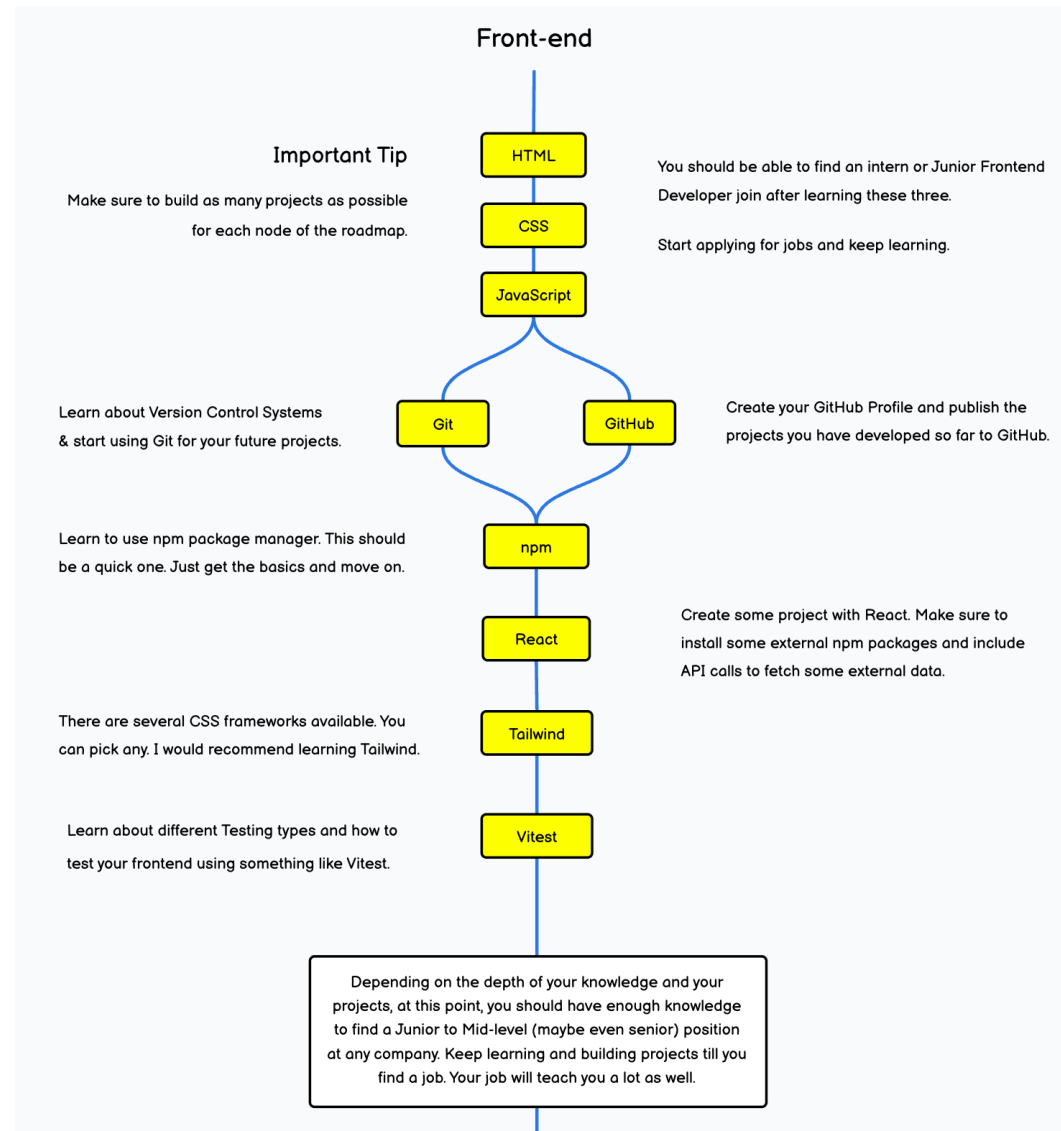


# Дорожная карта фронтенд-разработчика



<https://roadmap.sh/frontend>

# Дорожная карта фронтенд-разработчика



# Как работают веб-приложения?

## Фронтенд (клиентская часть / клиент)

- Обменивается данными с бэкендом
- Отрисовывает данные в нужном виде

## Бэкенд (серверная часть / сервер)

- Обменивается данными с фронтендом
- Обрабатывает данные
- Общается с базой данных



<https://doka.guide/tools/web-app-works/>

# Ключевые технологии

# HTML, CSS и JS

- **HTML** — язык разметки для создания структуры веб-страницы и представления контента
- **CSS** — язык каскадных стилей, который задаёт визуальное оформление для HTML
- **JavaScript** — основной язык программирования в вебе



**HTML**



**HTML + CSS**



**HTML + CSS  
+ JAVASCRIPT**

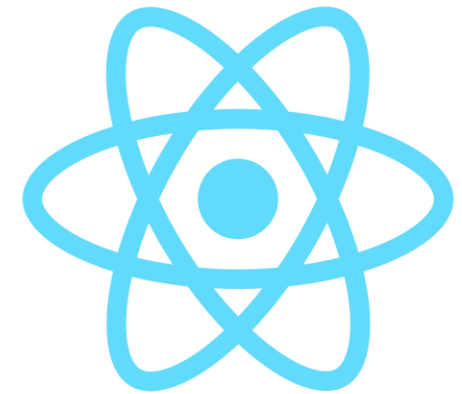
# UI-библиотеки и фреймворки

В мире фронтенда много различных фреймворков для разработки веб-приложений.

Большинство из них различаются подходами к разработке и инструментами, которыми они обладают.

Но в основе всех этих фреймворков часто лежат одни и те же концепции:

- Реактивность
- Virtual DOM

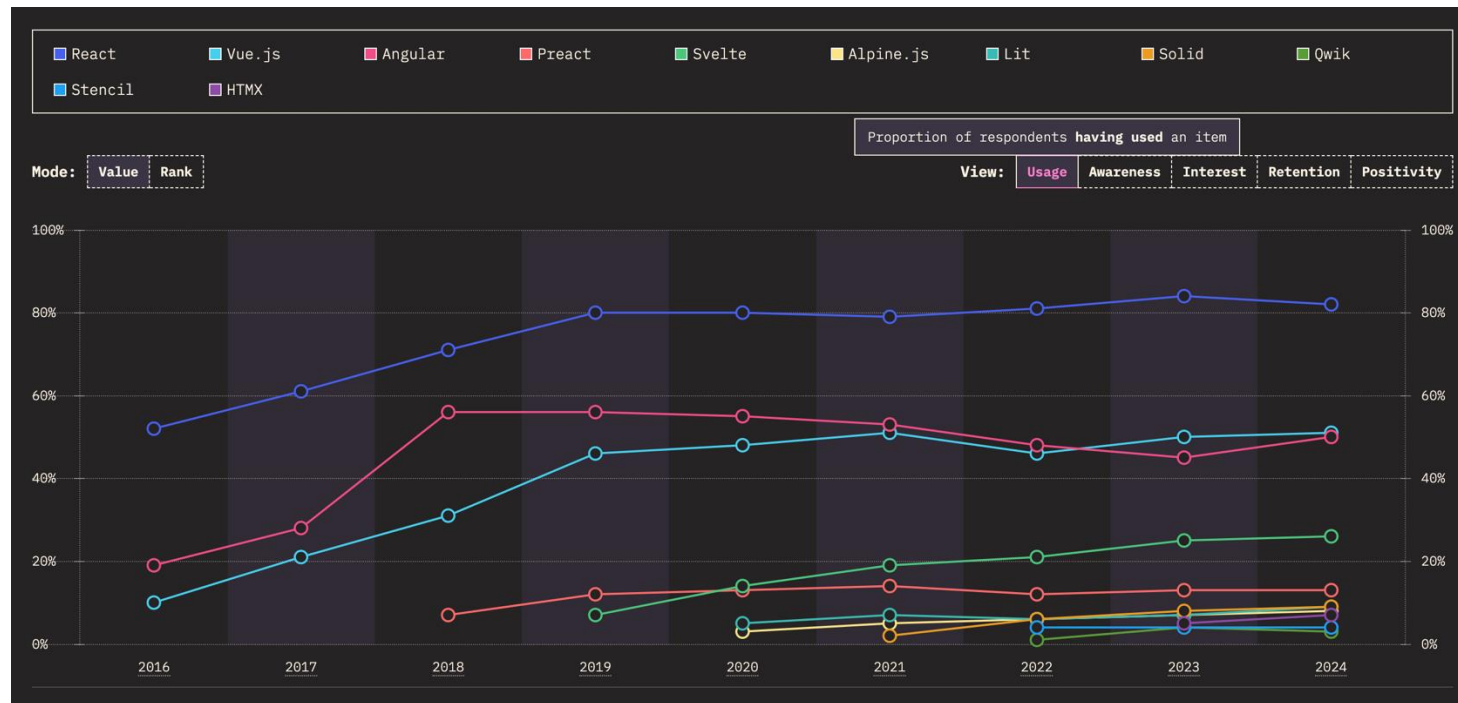


Логотип React

# UI-библиотеки и фреймворки

## Популярные инструменты:

- React
- Svelte
- Vue
- Angular



<https://2024.stateofjs.com/en-US/libraries/front-end-frameworks/>

# Виды веб-приложений

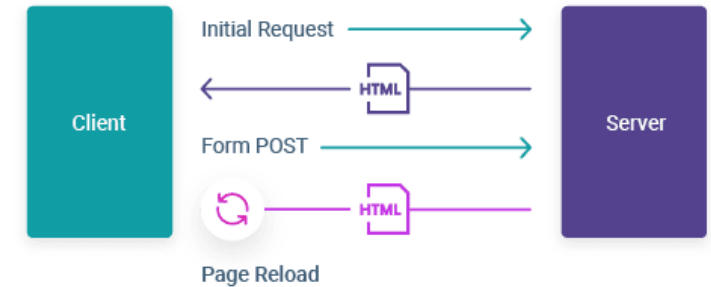
## Статические многостраничные приложения (MPA)

- Набор готовых сверстанных страниц
- Динамическая генерация HTML на сервере

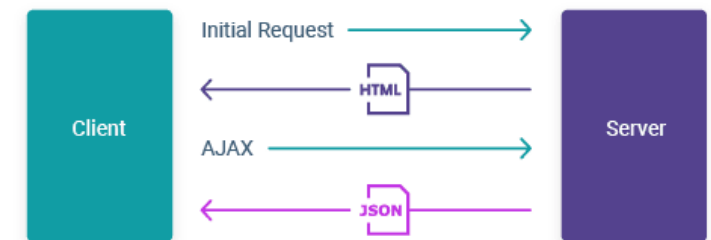
## Одностраничные приложения (SPA)

- Client-Side Rendering (CSR)
- Server-Side Rendering (SSR)

Multi-Page Lifecycle



SPA Lifecycle





# TypeScript

В JavaScript слабая динамическая типизация:

- В операциях с переменными разных типов они будут автоматически приведены к одному
- Любая переменная может произвольно менять свой тип во время выполнения программы

**TypeScript** = JavaScript + строгая типизация



Логотип TypeScript

Инфраструктура

# Серверный JavaScript

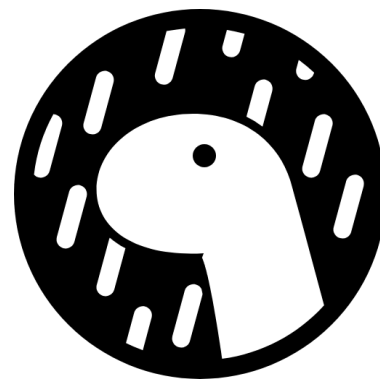
**Node.js** — среда выполнения кода JavaScript вне браузера, позволяющая писать серверный код для веб-страниц и веб-приложений, а также для программ командной строки.

## Альтернативы:

- Deno
- Bun



Node.js



Deno



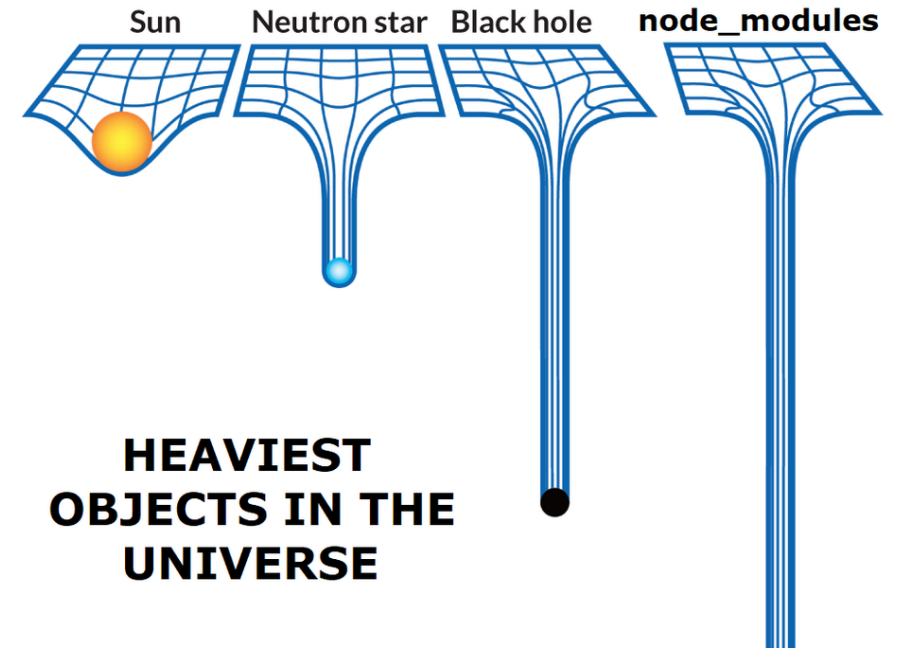
Bun

# Пакетные менеджеры

Для удобной работы с ними существуют пакетные менеджеры (менеджеры зависимостей). Они помогают выбрать версию библиотеки и гарантируют безопасность.

## Популярные менеджеры:

- npm – стандартный (идёт сразу с Node.js)
- yarn
- pnpm

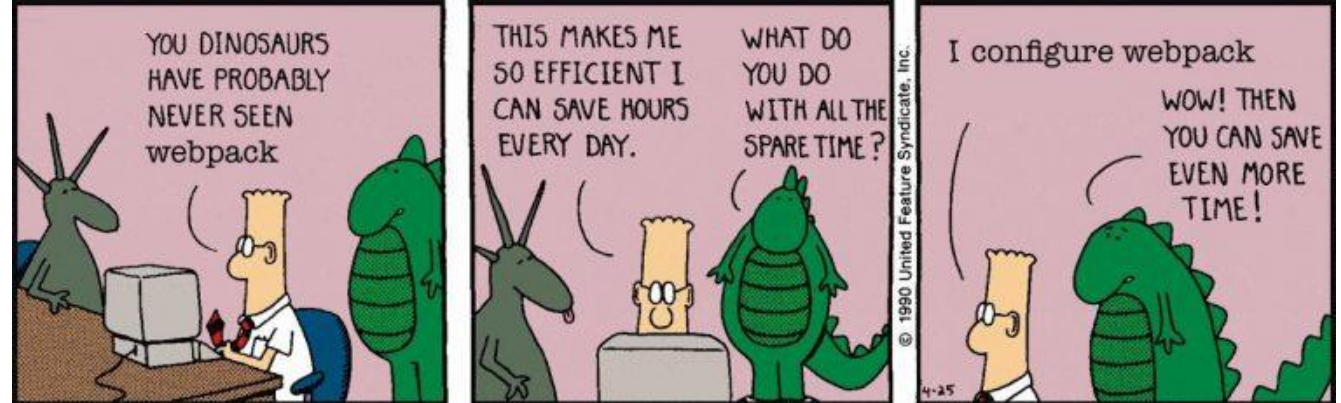


# Сборка

**Бандлер (сборщик)** — программа, которая упаковывает сложный проект со многими файлами и внешними зависимостями в один или несколько файлов, которые будут отправлены браузеру.

## Популярные сборщики:

- vite
- webpack
- esbuild
- parcel



# ЛИНТИНГ

Статический анализ (линтинг) — процесс исследования исходного кода без его непосредственного запуска.

## Плюсы линтинга:

- Уменьшается количество ошибок
- Быстрее код-ревью
- Соблюдение кодстайла

**ESLint** — один из популярных линтеров для JavaScript



# Тестирование

Тесты делают код более прочным и живучим. Одновременно с этим тесты — это отличная документация, которая не врёт и не устаревает.

## Виды тестов:

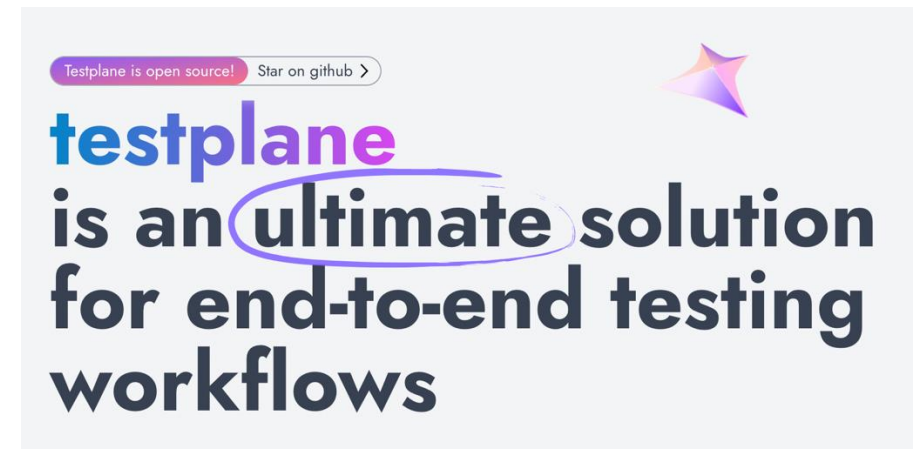
- Юнит-тесты
- Интеграционные тесты
- e2e-тесты
- Скриншотные тесты



# Тестирование

## Популярные инструменты:

- Jest (юнит, интеграционные и снапшоты)
- Testplane (скриншотные и e2e)
- Playwright (скриншотные и e2e)





# No-code решения

**No-code**-инструменты позволяют обычным пользователям создавать веб-приложения без необходимости написания программного кода (или с минимальной необходимостью — **Low-Code**).

## Примеры:

- WordPress/Tilda/Wix/etc
- Notion
- Airtable
- Typeform

## No Code

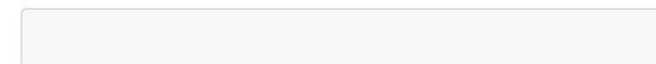
---

No code is the best way to write secure and reliable applications. Write nothing; deploy nowhere.

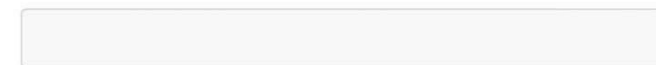
## Getting Started

---

Start by not writing any code.



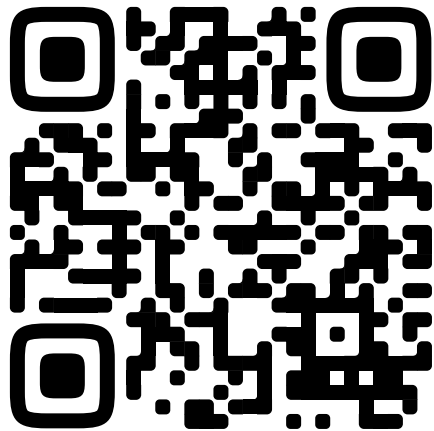
This is just an example application, but imagine it doing anything you want. Adding new features is easy too:



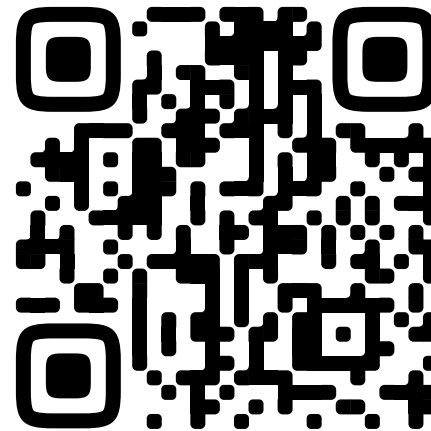
The possibilities are endless.

# Домашнее задание

- Ищем макеты приложения, которое будем делать в дальнейшем.
- Знакомимся с HTML и CSS.



<https://www.figma.com/community>



<https://doka.guide/>

# Полезные материалы

- <https://github.com/lyaplyap/frontend-development-course> — данный курс на GitHub
- <https://doka.guide> — про веб-разработку понятным языком от профессиональных разработчиков
- <https://www.youtube.com/watch?v=aoK2axqgR7k&list=PLXtiZNKlobF6U1V0yV248MT8ubBMg8iH2> — лекции от школы разработки интерфейсов Яндекса (ШПИ 2024)
- <https://htmlbook.ru> – один из самых толковых ресурсов HTML и CSS
- <https://learn.javascript.ru> – самый подробный учебник по JavaScript с примерами и задачами
- <https://developer.mozilla.org/ru> – документация HTML, CSS, JS и WebAPI от Mozilla