# JavaScript

JavaScript используется для создания интерактивных элементов в html.

Фреймворки для JS помогают упростить создание сложных интерактивных элементов.

## React

### Информация перед стартом

1. Для создания react компонентов в любом проекте (полноценный фронтенд или просто один отдельный визуальный react компонент) требуется создать отдельную папку с проектом реакта.

По сути создание отдельного компонента похоже на создание большого реакт проекта, но различается наполнением структуры в папке проекта.

1. Таким образом для разных целей создание структуры react проекта будет отличаться.

Команды для создания проектов:

npm init - базовая команда для создания JS проекта почти с нуля ( всё в ручную, из пустого проекта в ручную можно собрать любой из тех что ниже)

npx create-react-app my-app – классический реакт проект

npm create vite@latest my-app --template react – реакт проект со сборщиком vite, который гораздо быстрее и современее.

npx create-next-app@latest my-website – шаблон реакт проекта на фреймворке next для больших сервисов.

### Необходимое ПО

1. Для корректной работы любого фронтэнда или отдельных реакт компонентов нужно убедится что базовые вещи работали

JavaScript

Node js

npm

Для этого требуется установить Node.js

1) Проверка установки npm + nodejs

node -v

npm –v

Обновление npm до последней версии

npm install -g npm@latest

2) npm install -g npm@latest не работает

> PS D:\#work\zadaniya\5\_administrate\_BD> npm -v

10.9.2

> PS D:\#work\zadaniya\5\_administrate\_BD> npm install -g npm@latest

npm error code SELF\_SIGNED\_CERT\_IN\_CHAIN

npm error errno SELF\_SIGNED\_CERT\_IN\_CHAIN

npm error request to https://registry.npmjs.org/npm failed, reason: self-signed certificate in certificate chain

npm error A complete log of this run can be found in: C:\Users\Rud\_MS\AppData\Local\npm-cache\\_logs\2025-03-07T07\_45\_27\_507Z-debug-0.log

Вот что помогло:

> PS D:\#work\zadaniya\5\_administrate\_BD> npm config set strict-ssl false

### Создание проектов

#### Создание чистого JS проекта

1. Создание папки с проектом

npm init / npm init -y

1) после этого автоматически создаётся package.json

{

"name": "react\_dbsm\_modules",

"version": "1.0.2",

"description": "",

"main": "./dist/index.js",

"module": "./dist/index.mjs",

"types": "./dist/index.d.ts",

"repository": {"url": "https://github.com/Xhale-1/react\_dbsm\_modules"},

"scripts": {

"build": "tsup"

},

"keywords": [],

"author": "",

"license": "ISC",

"devDependencies": {

"tsup": "^8.5.0",

"typescript": "^5.8.3"

}

}

Плюс установить зависимости разработчика

npm install -D typescript tsup

2) Если программа на ts добавить tsconfig.json

{

"compilerOptions": {

"strict": true,

//"noImplicitAny": true,

"esModuleInterop": true,

//"strictNullChecks": true,

"target": "ES2022",

"moduleResolution": "Node10",

"module": "CommonJS",

"declaration": true,

"isolatedModules": true,

"noEmit": true,

"outDir": "dist"

},

"include": ["src"],

"exclude": ["node\_modules"]

}

3) если мы решили поставить компилятор tsup нужно добавить tsup.config.ts

import { defineConfig } from 'tsup';

export default defineConfig({

format: ['cjs', 'esm'],

entry: ['./src/index.ts'],

dts: true,

shims: true,

skipNodeModulesBundle: true,

clean: true,

});

4) Нужно создать папки dist и src + в src добавить любой js файл

5) проверить все ли работает с помощью npm run build

6) если да, то добавить свои файлы в src

7) регистрация

npm adduser

npm whoami

#### Создание универсального проекта на vite

1. Создать структуру и базовые build run скрипты

Можно так:

npm create vite@latest

Можно указать имя и фреймворк (react или vue)

npm create vite@latest my-vue-app -- --template vue

npm create vite@latest my-app -- --template react

cd react\_comps

npm install

npm run dev

### Библиотеки

#### Тестирование js npm библиотеки

1) Создаём папку

2) npm init –y

3) npm install react\_dbsm\_modules

4) компилияция файла: npx tsc file.ts. После чего появится file.js

5) Запуск скомпилированного файла: node file.js

#### Тестирование без скачивания js npm библиотеки

1) npm run build бибилиотеку

2) npm link

3) в другом проекте устанавливаем эту бибилиотеку npm link react\_dbsm\_modules