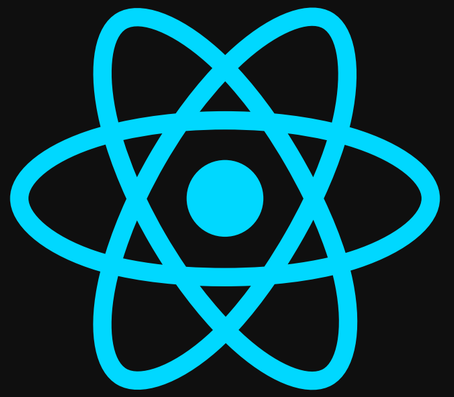
Презентация для nFactorial 2025

Тема: React для новичков

React — это библиотека для создания пользовательских интерфейсов.



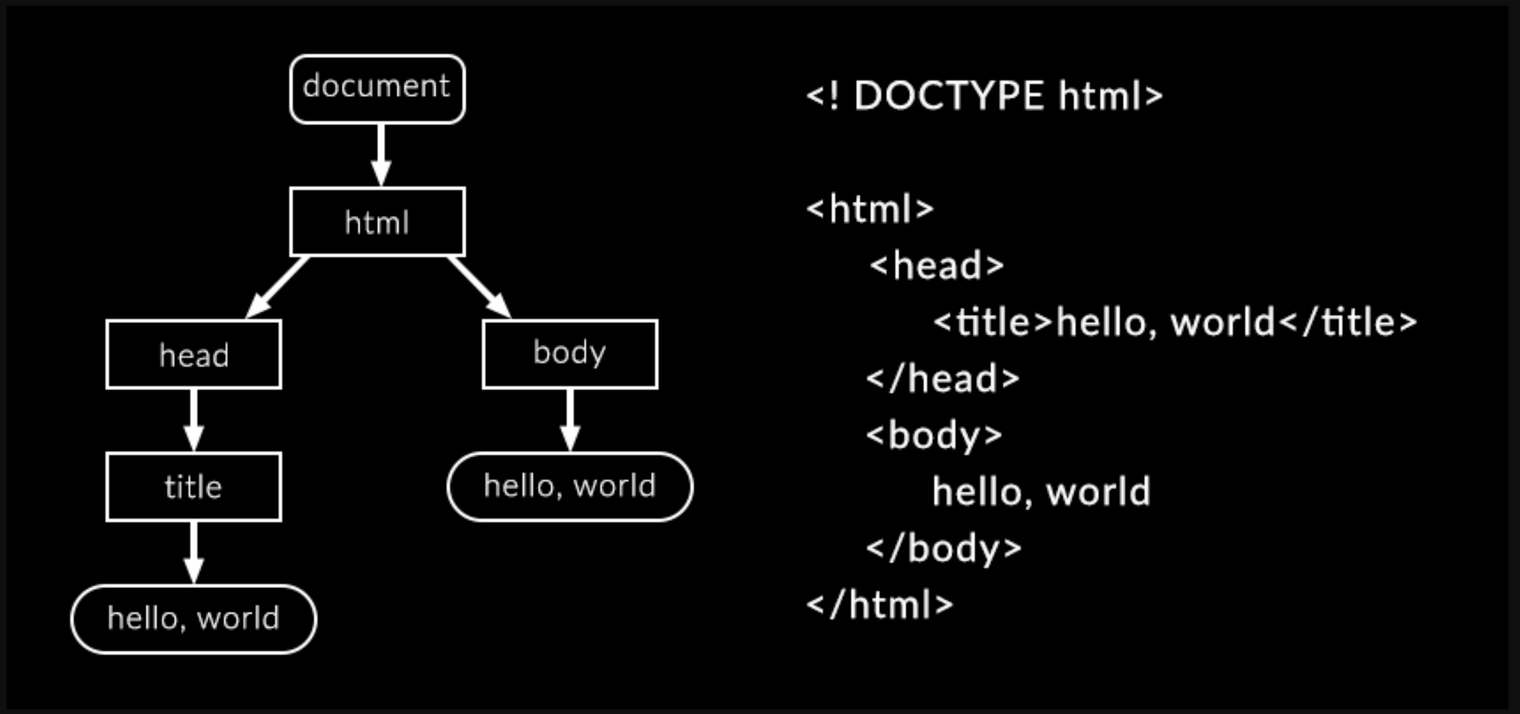
Зачем нужен **React**?

1) Обновлять только нужные части сайта, не перезагружая всю страницу

2) Делить интерфейс на компоненты – независимые блоки, как детали конструктора LEGO

DOM - как выглядит и что это?

Document Object Model (DOM) — это программный интерфейс для веб-документов. Он представляет структуру документа как дерево узлов, где каждый узел является частью документа, такой как элемент, текст, комментарий и так далее.



Обычная вёрстка + JavaScript

Это становится сложно и медленно,
если на странице много элементов

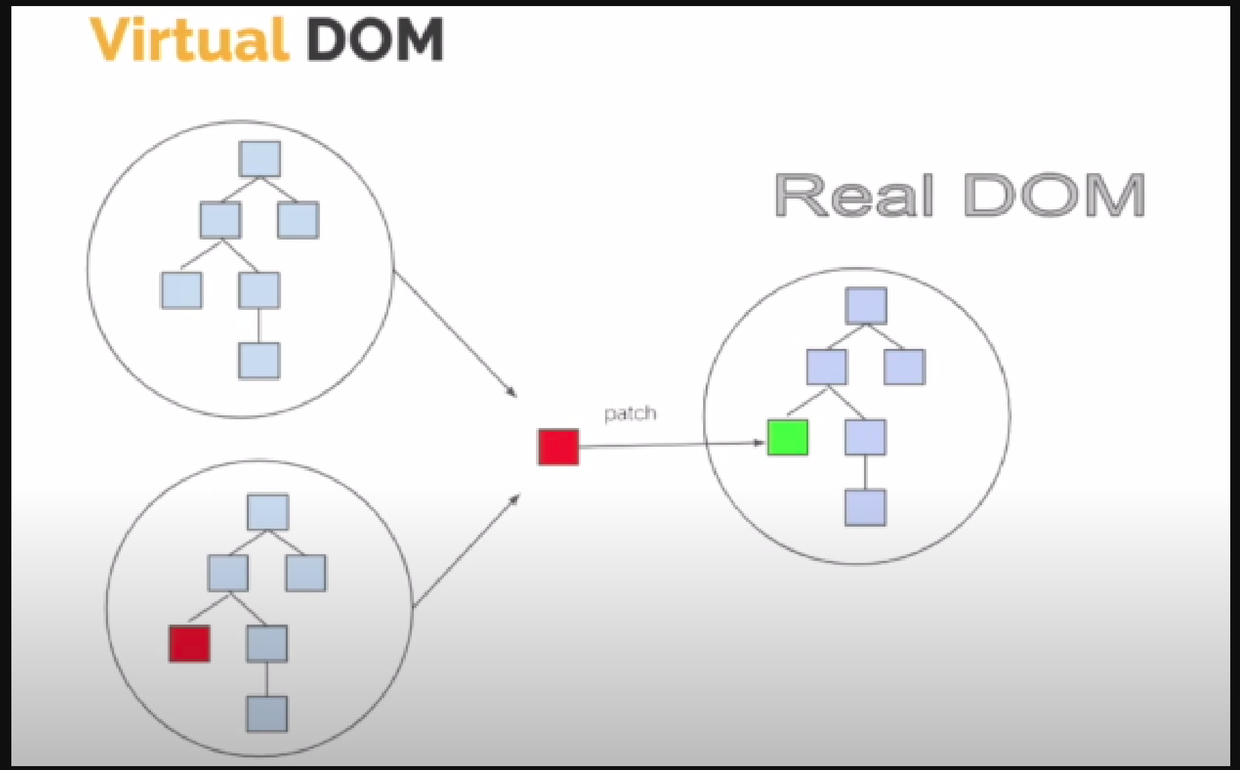
Мы вручную выбираем элементы DOM (через document.querySelector)

```
const button = document.querySelector('button.add');
const input = document.querySelector('input.new');
const itemsList = document.querySelector('ul.items');
const totalText =
document.querySelector('span.total');
```

Показать в excailDraw

React

В React мы не обновляем DOM сами, React обновляет DOM сам



Разница между стандартной верстkой и React

Что происходит при изменении данных?	Обычный JS	React
Обновляется весь DOM вручную?	Да, если сам пишешь	Нет
Страница перезагружается?	Иногда (если сам заставляешь)	Нет
Обновляется только нужная часть?	Нужно писать вручную	React делает это сам
Нужно думать, *как* обновить?	Да	Нет, ты просто описываешь *что должно быть*

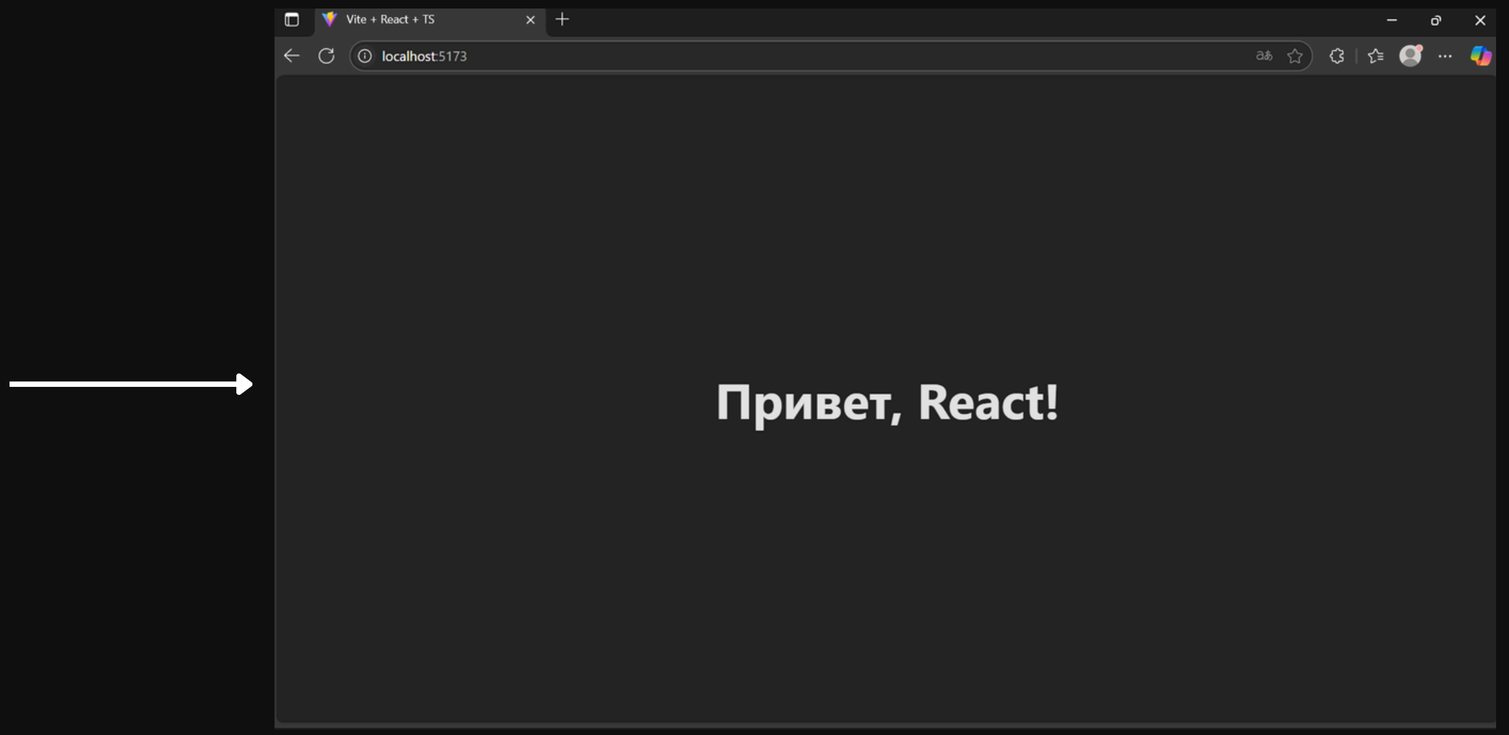
JSX: HTML внутри JavaScript

JSX - синтаксис, позволяющий писать HTML-подобный код внутри JavaScript.

```
const element = <h1>Привет, React!</h1>;
```

переменная

```tsx
import './App.css'

function App() {

  const element = <h1>Привет, React!</h1>;


  return (
    <>
      <div>
      {element}
      </div>
    </>
  )
}

export default App
```



React-router-dom

react-router-dom - это библиотека, которая позволяет делать переходы между страницами в приложении на React, без перезагрузки страницы.

```
npm i react-router-dom
```

Установка зависимости

```
import './App.css'
import { Routes, Route } from 'react-router-dom';
import DrawPad from './pages/DrawPad/DrawPad';

function App() {

  return (
    <>
      <div className='App'>
        <Routes>
          <Route path="/" element={<DrawPad />} />
        </Routes>
      </div>
    </>
  )
}

export default App
```

импорт библиотеки

Это обёртка, внутри которой указываешь все возможные маршруты (страницы) твоего приложения.

<Route> - это отдельный маршрут

```
export default function Drawing() {

  return (
    <div >
        Draw
    </div>
    )
}
```

Пропсы

Props - это входные данные, которые ты передаёшь в компонент извне.

Подающий

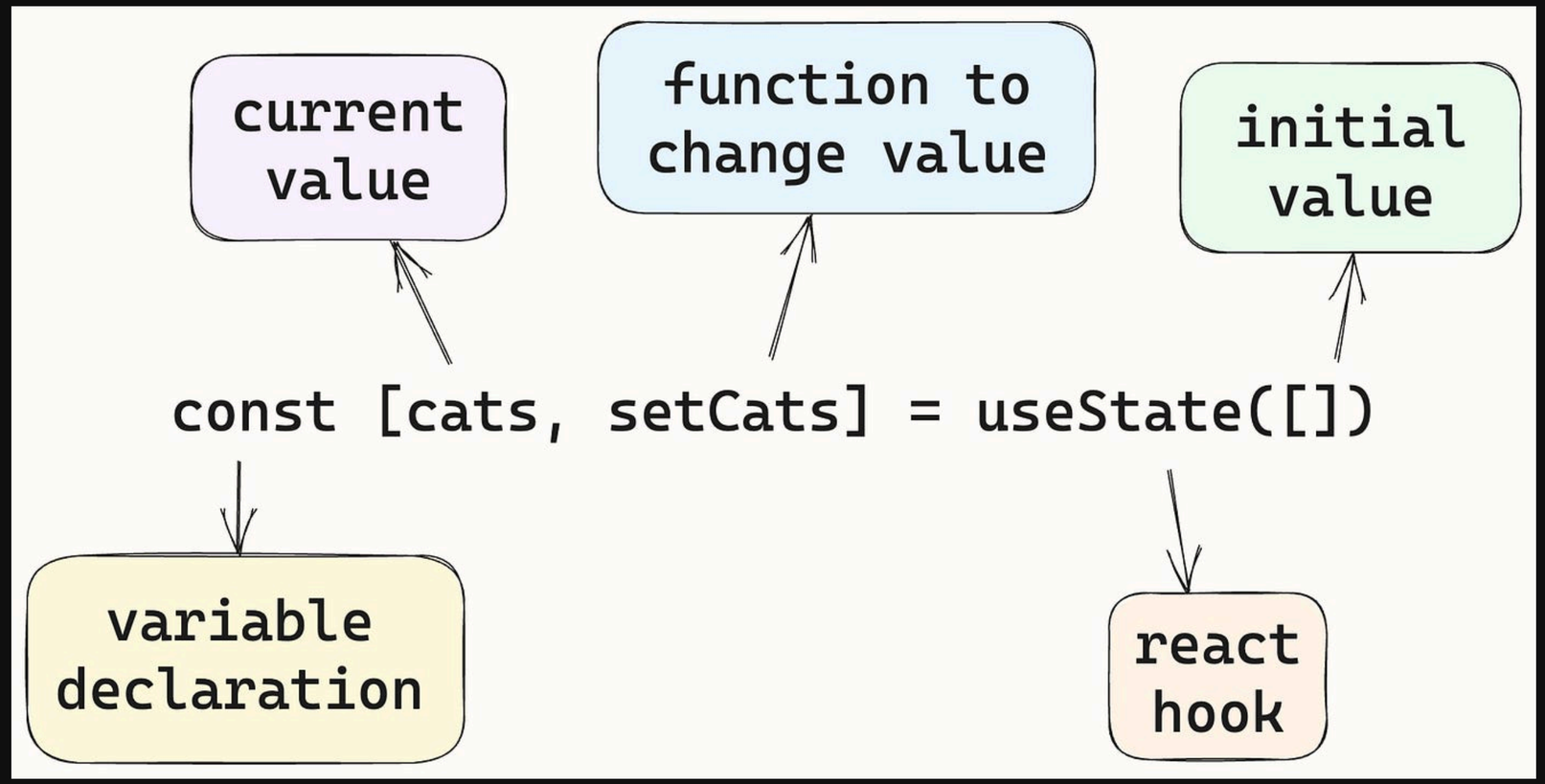
```jsx
import React from 'react';
import UserCard from './components/UserCard';

function Users() {
  const users = [
    { id: 1, name: 'Бахр', age: 22 },
    { id: 2, name: 'Баха', age: 27 },
    { id: 3, name: 'Газиз', age: 19 },
  ];

  return (
    <div>
      <h1>Список пользователей</h1>
      {users.map((user) => (
        <UserCard key={user.id} name={user.name} age={user.age} />
      ))}
    </div>
  );
}

export default Users;
```

Принимающий

```jsx
import React from 'react';

function UserCard(props) {
    console.log(props);

  return (
    <div style={{ border: '1px solid #ccc', padding: '10px', marginBottom: '10px' }}>
      <h2>{props.name}</h2>
      <p>Возраст: {props.age}</p>

    </div>
  );
}

export default UserCard;
```

Хуки — это такие волшебные функции в React, которые дают компоненту "память" и "мозги".

useState - это способ хранить данные внутри компонента.
Когда ты хочешь, чтобы компонент запоминал какое-то значение и обновлялся, ты используешь useState.



useEffect- это способ сказать React: “Сделай что-то после показа или обновления компонента.”

```
useEffect(() => {
  fetch('https://jsonplaceholder.typicode.com/users')
    .then((res) => res.json())
    .then((data) => {
      setUsers(data);
      setLoading(false);
    });
}, []);
```

```
const [users, setUsers] = useState([]);


useEffect(() => {
  fetch('https://jsonplaceholder.typicode.com/users')
    .then((res) => res.json())
    .then((data) => {
      setUsers(data);
      setLoading(false);
    });
}, [users]);
```

отслеживаем изменение

Домашнее задание