Eslint

import js from '@eslint/js'

import globals from 'globals'

import reactHooks from 'eslint-plugin-react-hooks'

import reactRefresh from 'eslint-plugin-react-refresh'

export default [

  { ignores: ['dist'] },

  {

    files: ['\*\*/\*.{js,jsx}'],

    languageOptions: {

      ecmaVersion: 2020,

      globals: globals.browser,

      parserOptions: {

        ecmaVersion: 'latest',

        ecmaFeatures: { jsx: true },

        sourceType: 'module',

      },

    },

    plugins: {

      'react-hooks': reactHooks,

      'react-refresh': reactRefresh,

    },

    rules: {

      ...js.configs.recommended.rules,

      ...reactHooks.configs.recommended.rules,

      'no-unused-vars': ['error', { varsIgnorePattern: '^[A-Z\_]' }],

      'react-refresh/only-export-components': [

        'warn',

        { allowConstantExport: true },

      ],

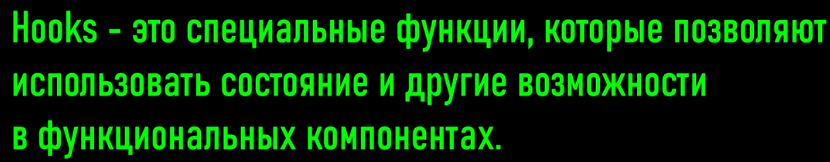
      'react/prop-types': 'off',

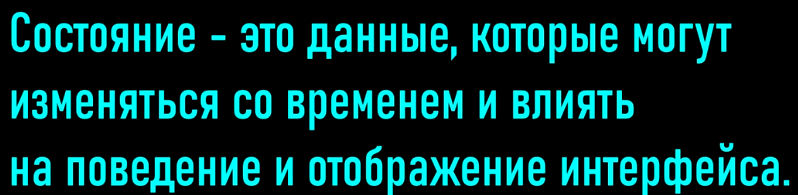
    },

  },

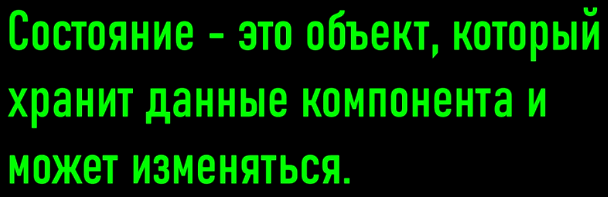
];

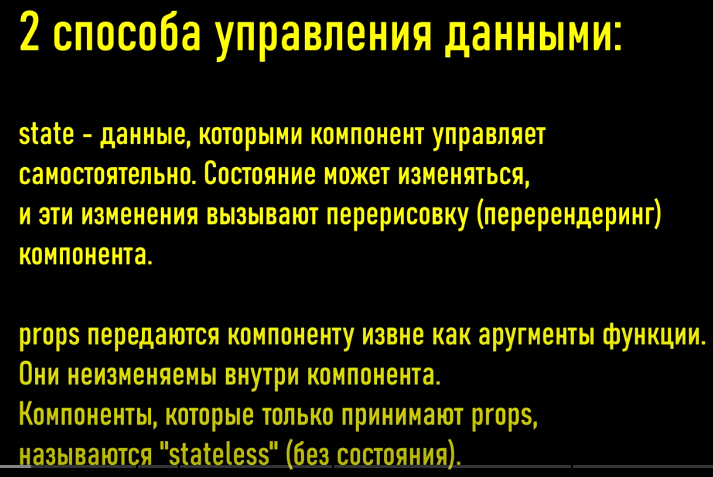
Выключаем поддержку ошибки типов.





В реакт состояние:





Функция обновления состояния работает асинхронно.

Состояние можно передавать только через общего родителя.

Функции обработчики событий должны начинаться через handle

**function HomePage({ clicks, setClicks }) {**

**const handleClick = () => {**

**alert('Click');**

**};**

**return (**

**<div className="home-page">**

**<Button label="Counter" callback={handleClick} />**

**<p>Количество кликов: {clicks}</p>**

**</div>**

**);**

**}**

**Условный рендеринг**

**if**

**function Greeting({ isLoggedIn }) {**

**if (isLoggedIn) {**

**return <h1>Добро пожаловать</h1>;**

**} else {**

**return <h1>Пожалуйста, войдите</h1>;**

**}**

**}**

**export default Greeting;**

**<Greeting isLoggedIn={true} />**

**<Greeting isLoggedIn={false} />**

**Тернарник**

**function Greeting({ isLoggedIn }) {**

**return (**

**<div>**

**{isLoggedIn ? (**

**<h1>Добро пожаловать</h1>**

**) : (**

**<h1>Пожалуйста, войдите</h1>**

**)}**

**</div>**

**);**

**}**

**Оператор двойного аперсанта**

**function Notification({ messages }) {**

**return (**

**<div>**

**{messages.length > 0 && <h1>У вас {messages.length} новых сообщений</h1>}**

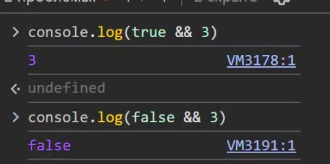
**</div>**

**);**

**}**

**export default Notification;**

**Используется для отображения элемнетов, если условие истинно.**

****

**Рендеринг списков react**

Ключ в элементе позволяет реакту идентифицировать какие элементы были добавлены, изменены, удалены.

Ключи помогают реакту оптимизировать процес рендеринга.

Ключи дожны быть строковыми литералами.

**const numbers = [1, 2, 3, 4, 5];**

**const listItems = numbers.map((number) => (**

**<li key={number.toString()}>{number}</li>**

**));**

**return <ul>{listItems}</ul>;**

**}**

Второй способ

**import { v4 as uuidv4 } from 'uuid';**

**function HomePage() {**

**const numbers = [1, 2, 3, 4, 5];**

**const listItems = numbers.map((number) => (**

// <li key={number.toString()}>{number}</li>

**<li key={uuidv4()}>**

**{number} - {uuidv4()}**

**</li>**

**));**

**return <ul>{listItems}</ul>;**

**}**

**Но лучше использовать id в сущностях.**

**function HomePage() {**

**const users = [**

**{ id: 1, name: 'Ivan' },**

**{ id: 2, name: 'Alice' },**

**{ id: 3, name: 'IvJoban' },**

**];**

**const userList = users.map((user) => <li key={user.id}>{user.name}</li>);**

**return <ul>{userList}</ul>;**

**}**

**export default HomePage;**