

Стилизация в React

№ урока: 6 **Курс:** React Essential

Средства обучения: Браузер Chrome, редактор кода VS Code или любой другой.

Обзор, цель и назначение урока

Узнать, как можно стилизовать компоненты в React. Научиться использовать CSS модули для стилизации.

Изучив материал данного занятия, учащийся сможет

- Стилизовать компоненты при помощи CSS модулей и препроцессора Sass.
- Использовать библиотеку clsx для формирования className компонентов.

Содержание урока

1. Обзор способов стилизации компонентов
2. Inline styles
3. Подключение css
4. Динамические классы
5. CSS модули

Резюме

- **Способы стилизации компонентов:**
 - o Inline CSS
 - o CSS in JS (for example styled-components)
 - o CSS Modules
 - o Sass, SCSS, Stylus, Less
 - o PostCSS
- **Атрибут style** - принимает JavaScript-объект со свойствами в camelCase вместо CSS-строк. Такой подход повышает эффективность и защищает от XSS.
- **Браузерные префиксы** - браузерные префиксы не добавляются к стилям автоматически. Для поддержки старых браузеров необходимо добавить соответствующие свойства стиля. Ключи стилей записываются в camelCase для обеспечения доступа к аналогичным свойствам DOM из JS. Префиксы браузера [кроме ms](#) должны начинаться с заглавной буквы. Вот почему WebkitTransition начинается с заглавной «W».
- **Единицы измерений** - React автоматически добавит суффикс «px» к свойствам стилей с числовым значением. Если вы хотите использовать единицы измерения, отличные от «px», укажите значение в виде строки с желаемой единицей измерения.
- **CSS Модуль** — это CSS файл, в котором все имена классов и анимаций имеют локальную область видимости по умолчанию.
- **Подключение CSS** - каскадные таблицы стилей можно подключать также, как и в простом html файле всеми возможными способами. Create React App проект использует Webpack для обработки всех assets. Чтобы выразить, что файл JavaScript зависит от файла CSS, вам

необходимо импортировать CSS из файла JavaScript. В production сборке все CSS-файлы будут объединены в один уменьшенный файл .css.

- **Sass** (или Syntactically Awesome Stylesheets) — это скриптовый метаязык, разработанный для упрощения написания файлов CSS.

В зависимости от выбранного синтаксиса Sass-файлы могут иметь расширения:

- **.sass** - известный как синтаксис с отступами, через которые реализованы вложенные элементы. Фигурных скобок нет.
- **.scss** - Sassy CSS, где используются фигурные скобки.
- **Динамические классы** - вы можете формировать значения для атрибута «className» на лету. Так как это JSX - вы можете использовать тернарный или логические операторы для формирования значения атрибута className.
- **CLSX** - библиотека для условного построения строк className.
- **CSS in JS** — это подход написания стилей, при котором используется Javascript в качестве языка для описания стилей в декларативном и удобном в поддержке виде. Специальный компилятор превращает JS в CSS, который работает и на стороне сервера, и на стороне клиента в реальном времени.

Закрепление материала

- Что такое CSS Module?
- Какие способы стилизации компонентов вы знаете?
- В чем разница между CSS Modules и SASS?

Дополнительное задание

Задание

Прочитайте документацию create-react-app, а именно - раздел с Styles and Assets.

Самостоятельная деятельность учащегося

Задание 1

Выучите основные понятия, рассмотренные на уроке.

Задание 2

Создайте компонент формы, добавьте поддержку sass в проект - стилизуйте форму с помощью sass.

Задание 3

Добавьте поддержку css модулей в проект, используйте их при написании стилей. Добавьте валидацию на поле формы. Если при отправке поле является не валидным - должен добавиться стиль, при котором border данного поля должен стать красного цвета, а также форма не должна отправиться. Отправкой можно считать вывод строки «submit success» в консоли.

Рекомендуемые ресурсы

[5 способов стилизации компоненты](#)

[Руководство по трём способам стилизации](#)

[Overriding styles with CSS Modules](#)

[clsx](#)

[9 CSS in JS Libraries](#)

[Сравнением библиотек по работе с «CSS-in-JS»](#)

[CSS Custom Properties](#)

<https://ru.reactjs.org/docs/thinking-in-react>