Уроки по React (Junior)

**02.Зачем NodeJS frontend-разработчику?**

Frontend-разработка — разработка под браузер.

Браузер (Chrome, Opera и др.) — прикладное ПО для просмотра страниц и др.

NodeJS — платформа с открытым исходным кодом для работы с JS, построенная на движке Chrome V8. Она позволяет писать серверный код для веб-приложений и динамических веб-страниц, а также программ командной строки.

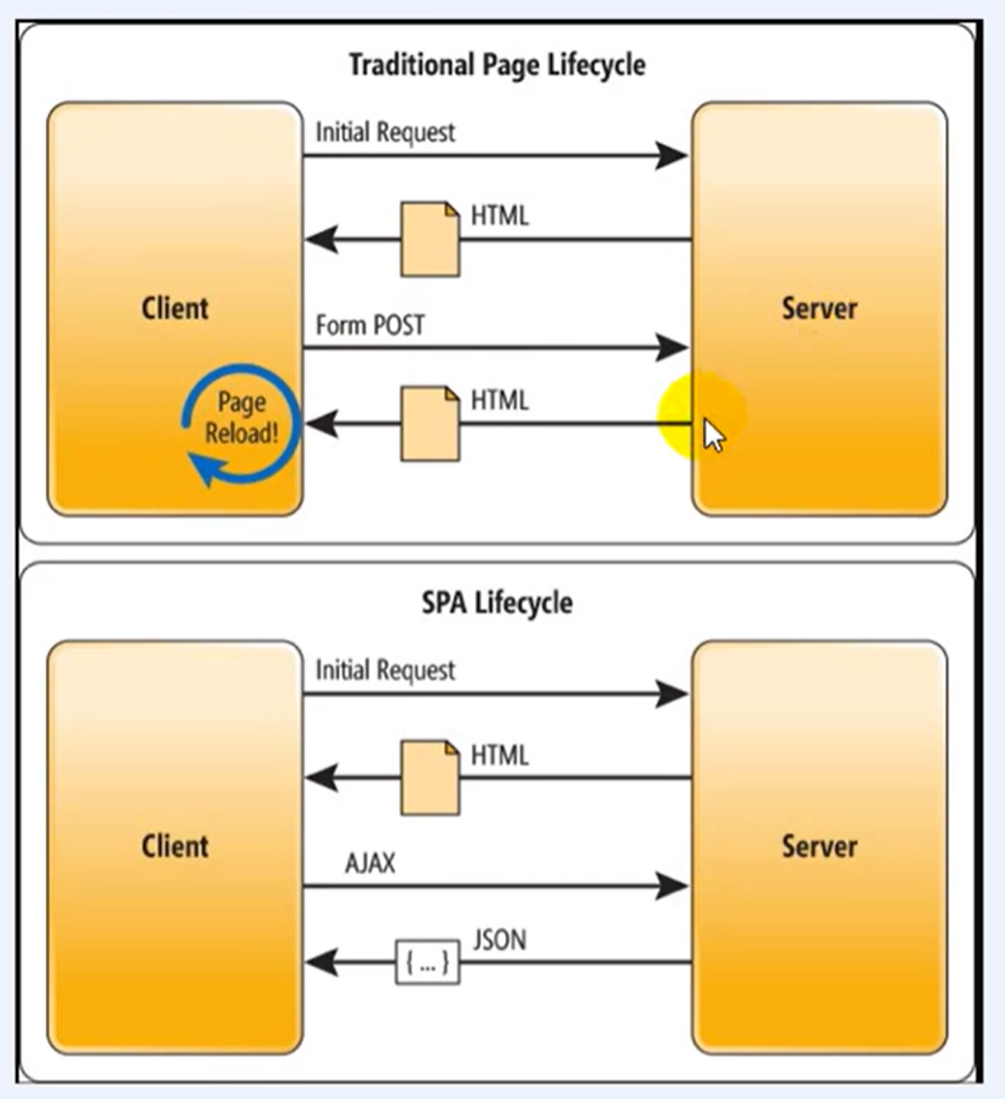
Браузер и NodeJS используют движок v8, поэтому оба воспринимают JS.

Npm, Yarn — пакетные менеджеры (устанавливают и следят за множеством пакетов в NodeJS).

Npx — инструмент для выполнения/запуска пакетов NodeJS, поставляется в комплекте с npm.

React работает с NodeJS, т.к. здесь есть все необходимые пакеты для компиляции и вывода данных.

**05.SPA: single page application**



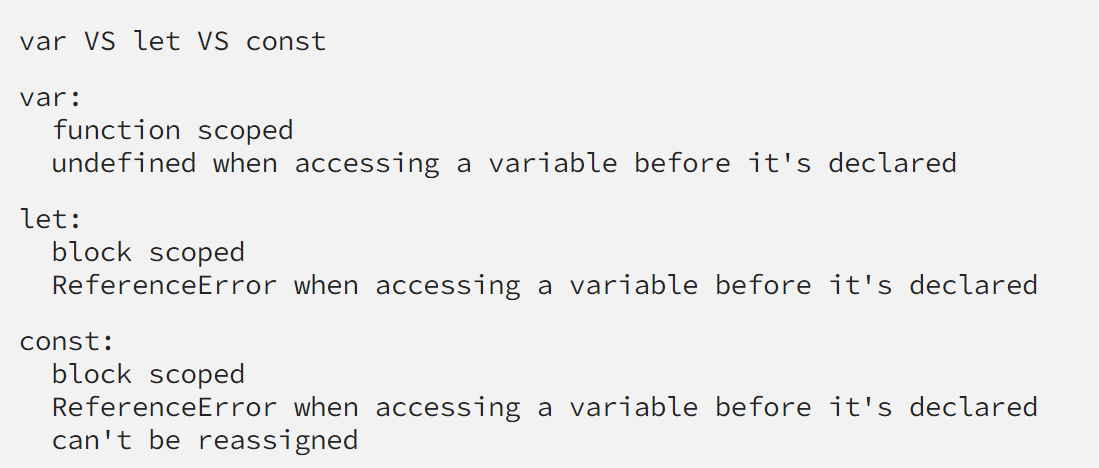
SPA это [веб-приложение](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5) или [веб-сайт](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B0%D0%B9%D1%82), использующий единственный [HTML-документ](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0) как оболочку для всех веб-страниц и организующий взаимодействие с пользователем через динамически подгружаемые [HTML](https://ru.wikipedia.org/wiki/HTML), [CSS](https://ru.wikipedia.org/wiki/CSS), [JavaScript](https://ru.wikipedia.org/wiki/JavaScript), обычно посредством [AJAX](https://ru.wikipedia.org/wiki/AJAX).

**06.Как устроен проект на React?**

JSX — расширение синтаксиса JS, похожее на HTML.

Вместо атрибута class необходимо указывать classname.

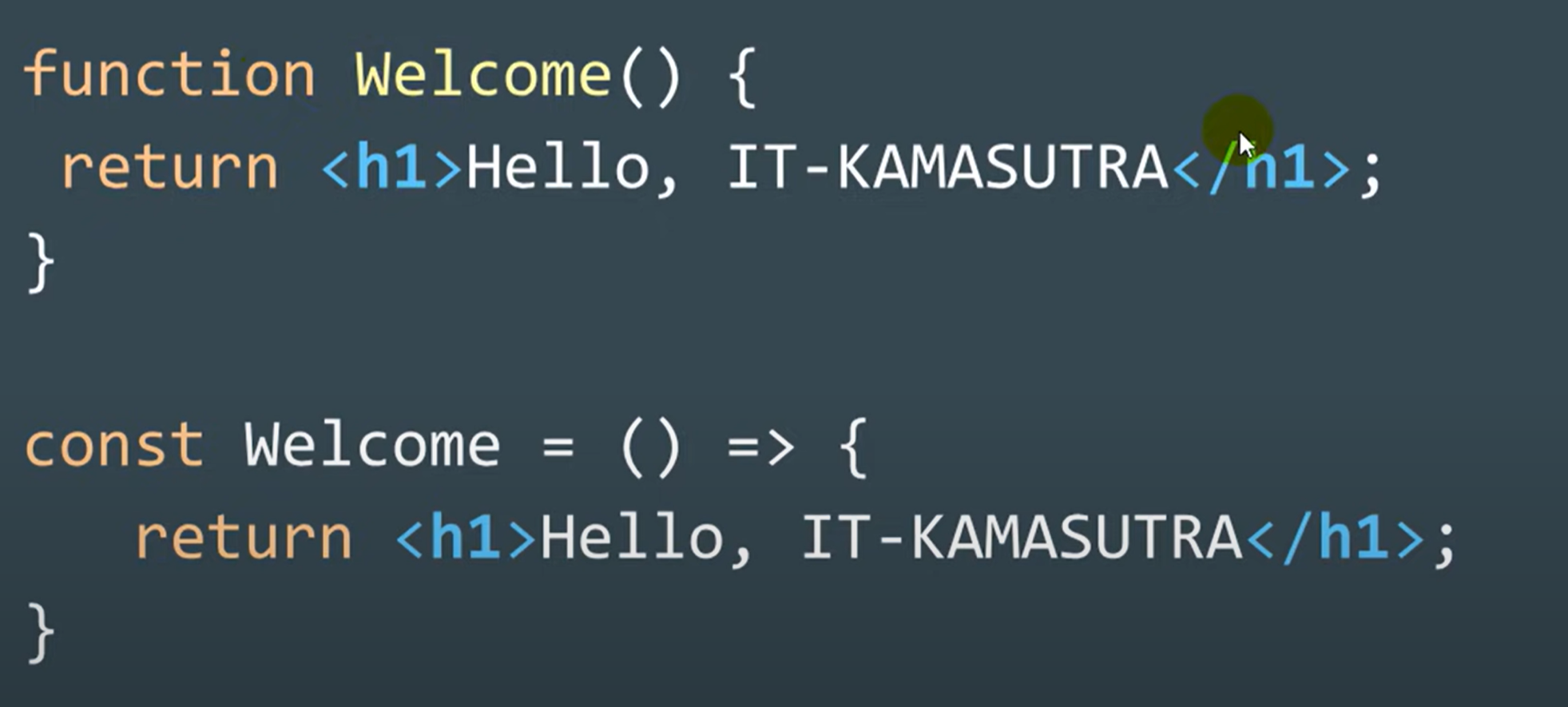
Компонента возвращает один корневой элемент.

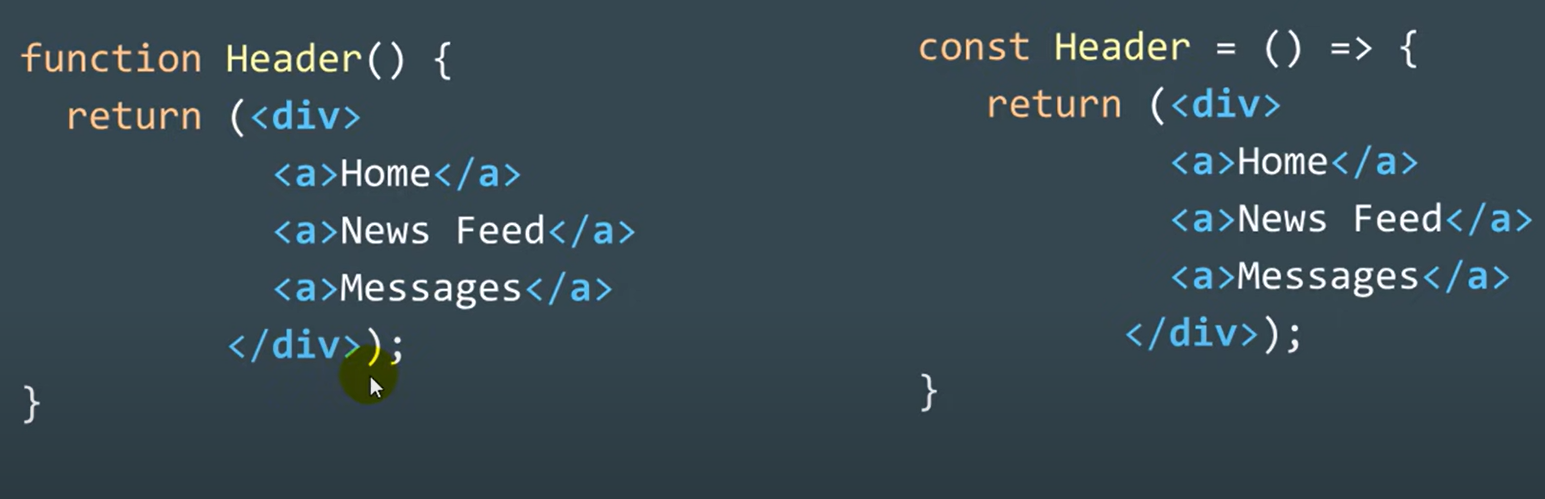


[Var vs let vs const](https://medium.com/nuances-of-programming/%D0%B2-%D1%87%D1%91%D0%BC-%D1%80%D0%B0%D0%B7%D0%BD%D0%B8%D1%86%D0%B0-%D0%BC%D0%B5%D0%B6%D0%B4%D1%83-var-let-%D0%B8-const-%D0%B2-javascript-3084bfe9f7a3)

**07.Что такое компонента?**

Компонента — функция, возвращающая разметку JSX. Компонента = тег.





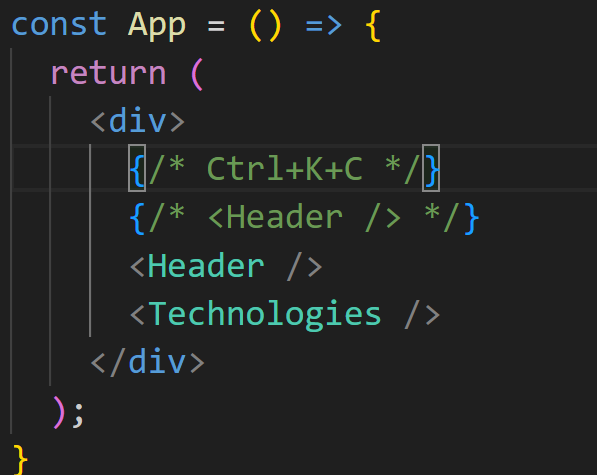
Использование компоненты внутри разметки JSX: <имя />, например, <Header />. Вариант Header(); неправильный!

**08.Теория по import/export**

[Модульный подход к разработке web-приложений с использованием JavaScript: AMD и RequireJS](https://habr.com/ru/articles/152833/)

**10.Опасность default + auto import**

Из-за default при экспорте и случайных подмен можно импортировать другой элемент



**11.Network. Верстка. Grid**

Для того, чтобы можно было отобразить на странице картинки, в компонентах необходимо указывать ссылки на картинки, расположенные в папке public.

[БЭМ методология](https://ru.bem.info/methodology/)

Grid: [MDN](https://developer.mozilla.org/ru/docs/Web/CSS/CSS_grid_layout/Basic_concepts_of_grid_layout), [doka](https://doka.guide/css/grid-guide/), [CSS tricks](https://css-tricks.com/snippets/css/complete-guide-grid/).

Flexbox: [MDN](https://developer.mozilla.org/ru/docs/Learn/CSS/CSS_layout/Flexbox), [doka](https://doka.guide/css/flexbox-guide/), [CSS tricks](https://css-tricks.com/snippets/css/a-guide-to-flexbox/).

**12.Разбивка на компоненты**

Ctrl+K+F=отформатировать код

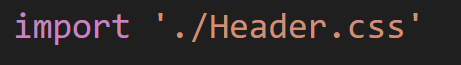
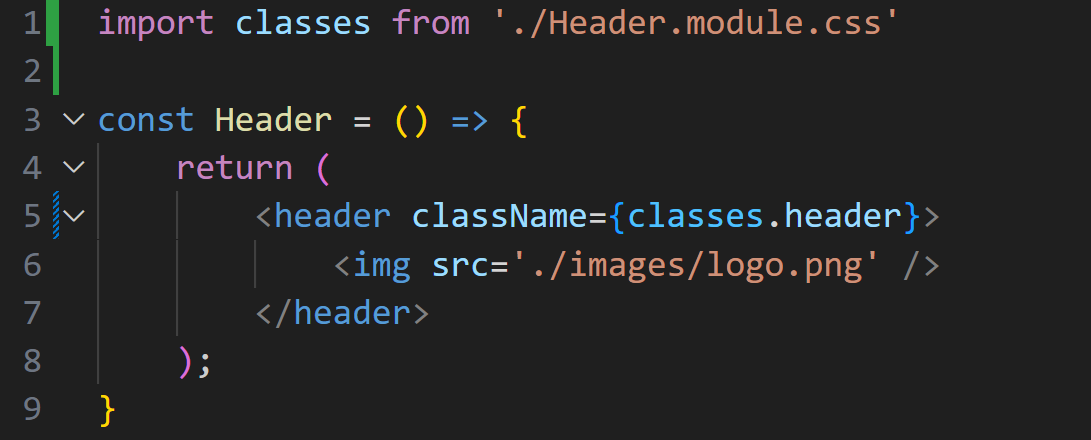
**14.CSS-modules**

[Эволюция CSS](https://habr.com/ru/companies/vk/articles/319956/)

Создание CSS модуля:



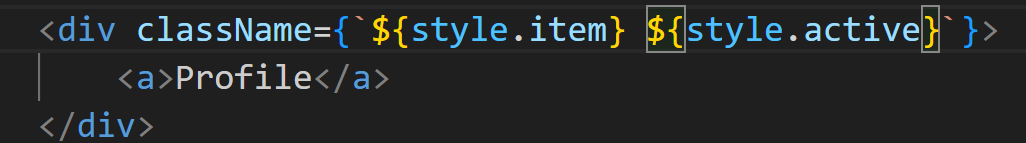
Отличие в импорте и использовании .css файла и модуля CSS

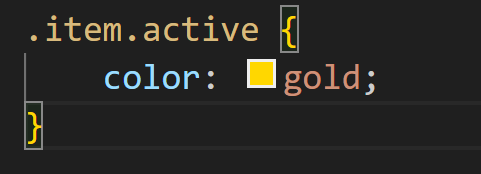
 => 

Как выглядит результат разделения стилей: «.компонента\_класс\_id»



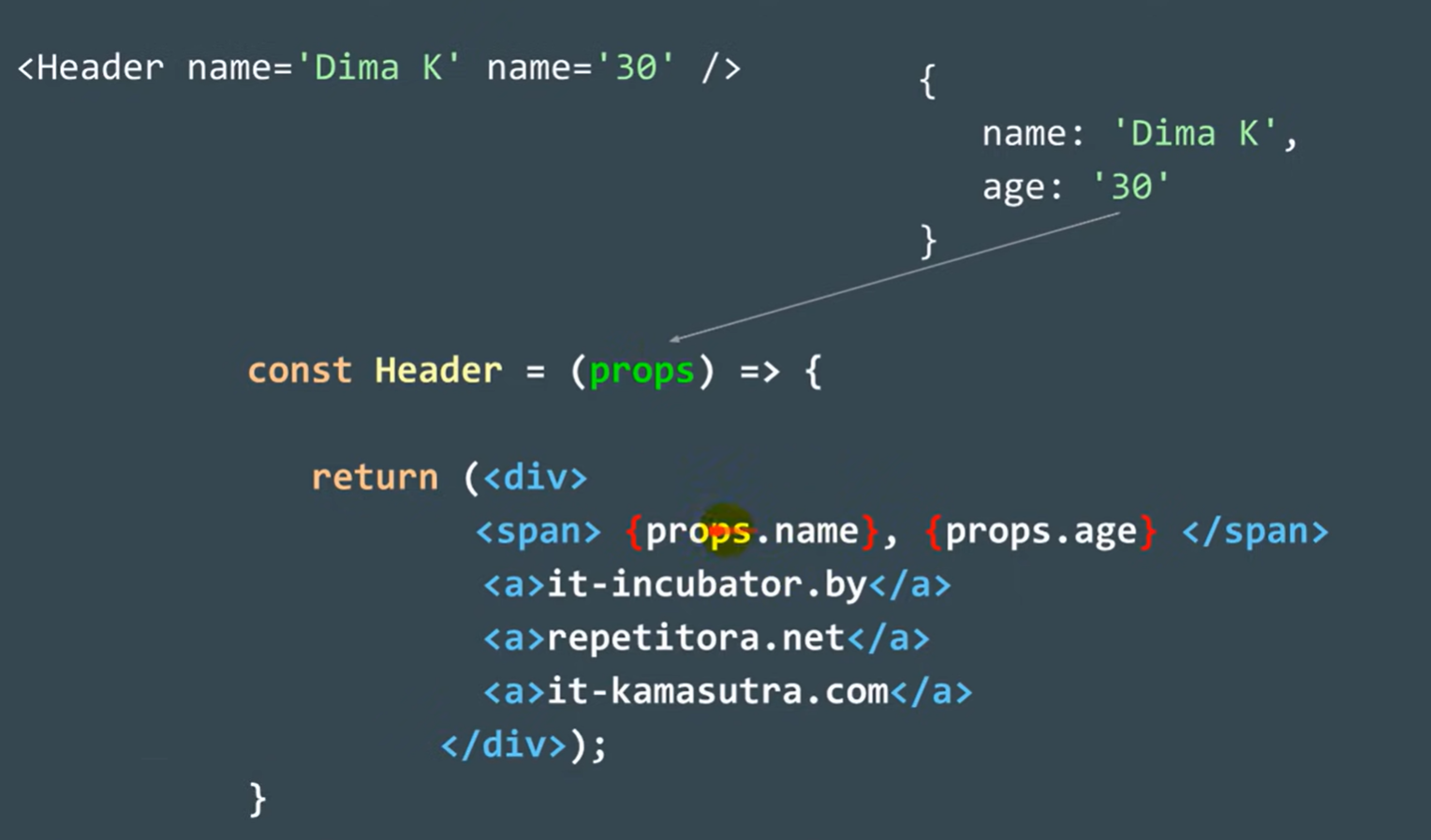
Если необходимо прикрепить 2 класса, то:

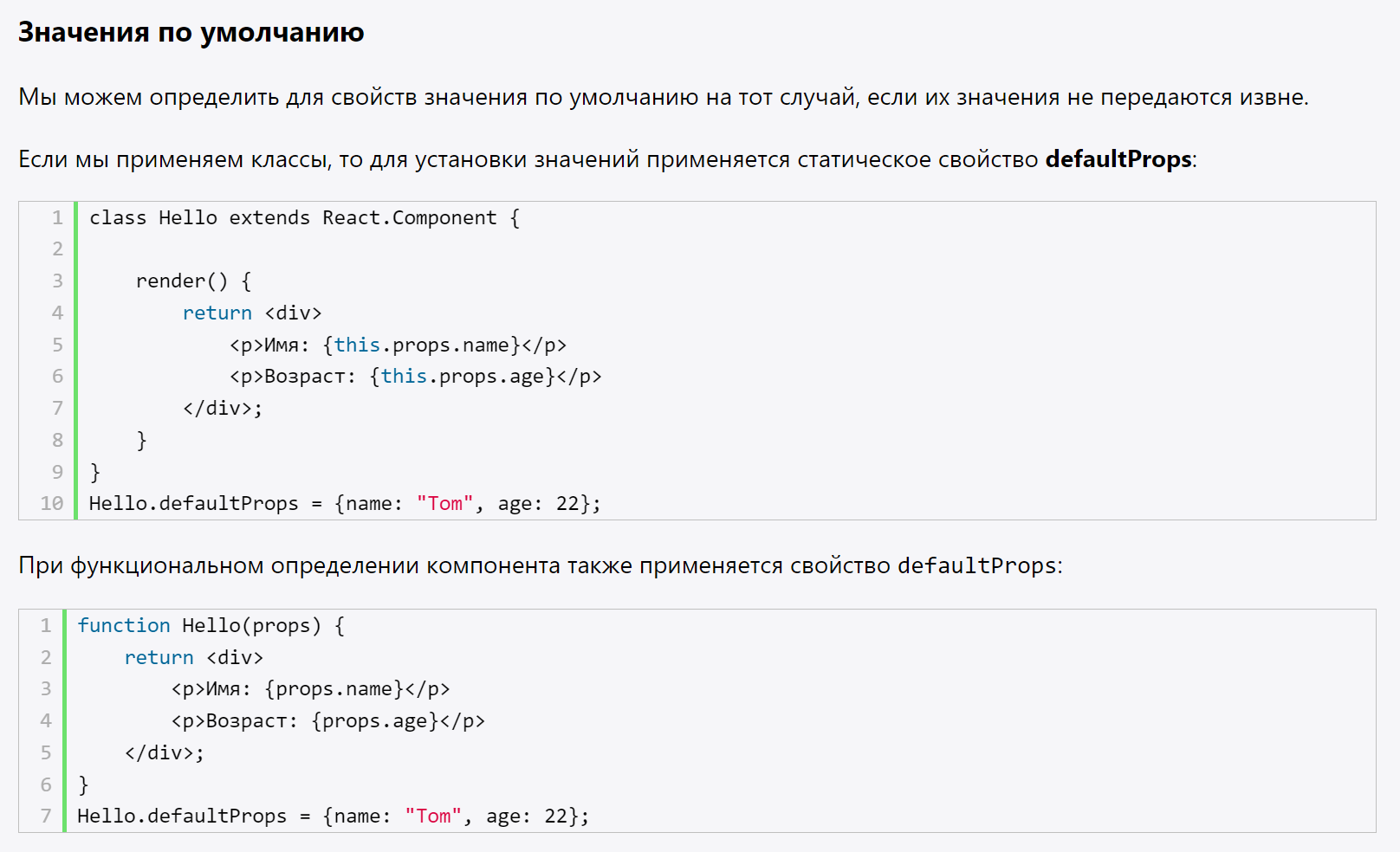




**16.Props**

Props — это входные данные React-компонентов, передаваемые от родительского компонента дочернему компоненту. Props приходят в компоненту всегда, но могут быть как пустым объектом, так и объектом со свойствами. Props предназначены только для чтения.





**20.NavLink**

[React Router](https://reactdev.ru/libs/react-router/)предоставляет компонент <Link> для создания ссылок в приложении. «За сценой» <Link> рендерит <a> с настоящим href, так что люди, использующие клавиатуру для навигации или экранные считывающие устройства (screen readers), смогут без проблем пользоваться приложением. <NavLink> — это специальный тип <Link>, позволяющий определять стили для активного состояния ссылки.

Необходимо, чтобы NavLink был внутри Router, пусть даже через вложенные компоненты.

[History API](https://doka.guide/js/window-history/)

[PreventDefault](https://doka.guide/js/event-prevent-default/)

**22.Route exact**

В React v6 не нужно использовать exact, все пути по умолчанию должны полностью совпадать. В нашем случае к path в App.js нужно добавить \* <Route path="/dialogs/\*" element={<Dialogs />} />

