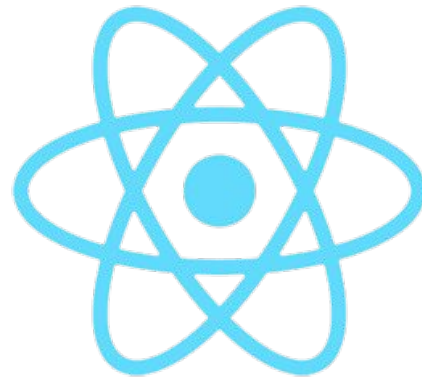


# React JS

ОСНОВЫ ОСНОВ



# Что такое React JS

- React - это библиотека JavaScript, которая используется для создания пользовательского интерфейса.
- Первоначально React предназначался для веба, для создания веб-сайтов, однако позже появилась платформа React Native, которая уже предназначалась для мобильных устройств.
- React представляется идеальным инструментом для создания масштабируемых веб-приложений.

# Виртуальный DOM

- Вся структура веб-страницы может быть представлена с помощью DOM (Document Object Model)- организация элементов html, которыми мы можем манипулировать, изменять, удалять или добавлять новые.
- Для взаимодействия с DOM применяется язык JavaScript.
- Виртуальный DOM представляет легковесную копию обычного DOM. И отличительной особенностью React является то, что данная библиотека работает именно с виртуальным DOM, а не обычным.

# Виртуальный DOM

- Если приложению нужно узнать информацию о состоянии элементов, то происходит обращение к виртуальному DOM.
- 
- Если необходимо изменить элементы веб-страницы, то изменения вначале вносятся в виртуальный DOM. Потом новое состояние виртуального DOM сравнивается с текущим состоянием. И если эти состояния различаются, то React находит минимальное количество манипуляций, которые

# Другие особенности React

- мы можем создать отдельные компоненты и затем их легко переносить из проекта в проект.
- мы используем JSX.
- JSX представляет комбинацию кода JavaScript и XML и предоставляет простой и интуитивно понятный способ для определения кода визуального интерфейса.
- также для компиляции кода нам необходим компилятор Babel, который позволяет работать с кодом по стандарту ES6(ES2015).
- Доступны шаблонизаторы проекта.

# Другие особенности React

- Используется концепция *View* из MVC.
- React позволяет создавать интерфейсы. В известном паттерне Model-View-Controller React ближе всего к пользователю. Он отвечает за представление данных, получение и обработку ввода пользователя. Где и в каком виде вы храните данные, как вы общаетесь с хранилищем, дело ваше. React—это всего лишь *View* вашего приложения.

# Другие особенности React

- «Реактивное программирование»
- React построен на парадигме реактивного программирования. Этот декларативный подход предлагает описывать данные в виде набора утверждений или формул. Изменение одного из параметров ведёт за собой автоматический пересчёт всех зависимостей.

# Самые свежие материалы

- Основной репозиторий:  
[https://github.com/facebook/react.](https://github.com/facebook/react)
- Основной сайт:
- <https://reactjs.org/>



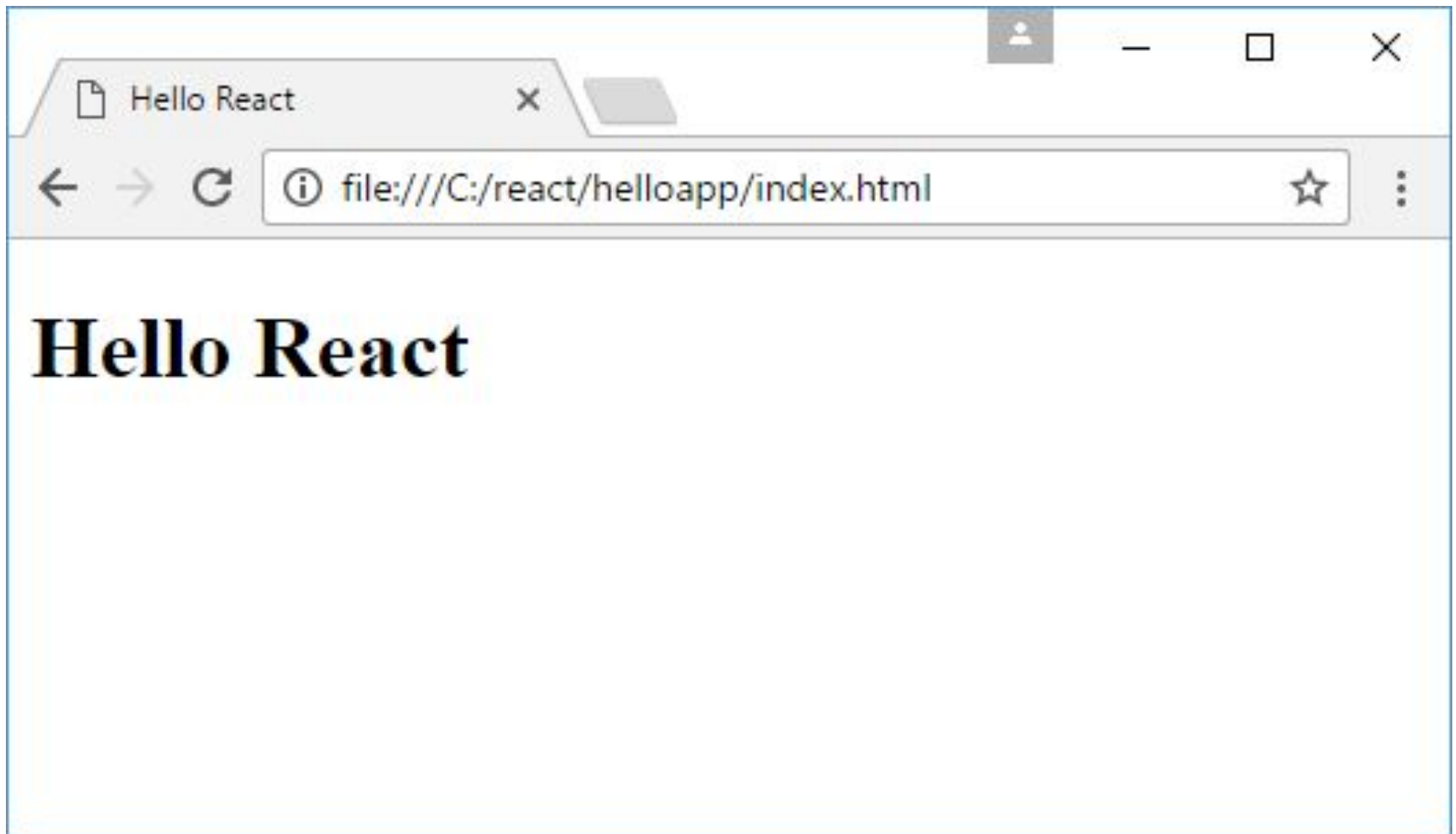
# Начало работы (еще проще)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Hello React</title>
</head>
<body>
  <div id="app"> </div>

  <script crossorigin src="https://unpkg.com/react@16/umd/react.production.min.js"></script>
  <script crossorigin src="https://unpkg.com/react-dom@16/umd/react-dom.production.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/babel-standalone/6.25.0/babel.min.js"></script>

  <script type="text/babel">
// рендеринг элемента
ReactDOM.render(
  <h1>Hello React</h1>, // элемент, который мы хотим создать
  document.getElementById("app") // где мы этот элемент хотим создать
)
</script>
</body>
```

# Начало работы



# Начало работы

- Для рендеринга элемента на веб-странице применяется метод `ReactDOM.render()`, который принимает два параметра.
- Первый параметр представляет компонент, который мы хотим отобразить на веб-странице. В данном случае это обычный заголовок.
- В этот метод передается элемент, который надо добавить на веб-страницу. Вторым параметром - это тот элемент веб-страницы, в котором будет производиться рендеринг компонента из первого параметра. В данном

# Установка

- Существует несколько способов использования React.
- Официально рекомендуемый способ - из npm или Facebook CDN, но дополнительно вы можете клонировать из git и создавать свои собственные сборки.
- Также вы можете использовать стартовый комплект или сэкономить время с помощью генератора каркаса приложения от Yeoman.

# Использование CDN для Facebook

- Для обеспечения быстрого доступа просто подключите библиотеки React и React Dom из CDN fb.me следующим образом:
- `<!-- The core React library -->`
- `<script`  
`src="https://fb.me/react-0.14.0.js"></script>`
- `<!-- The ReactDOM Library -->`
- `<script`  
`src="https://fb.me/react-dom-0.14.0.js"></script`  
`>`

# Установка из NPM

- В руководстве React также рекомендуется использовать `react` и `react-dom` пакеты npm. Чтобы установить их в своей системе, запустите следующую команду в командной строке `bash` в каталоге проекта или сначала создайте новый каталог и `cd` в него.
- 
- `$ npm install --save react react-dom`
- `$ browserify -t babelify main.js -o bundle.js`
- 
- Теперь вы сможете увидеть установку React внутри каталога `node_modules`.

# Что такое JSX?

- JSX - синтаксис XML/HTML, который используется для визуализации HTML из кода JavaScript. React преобразует JSX в собственный JavaScript для браузера, а с помощью предоставленных инструментов вы можете конвертировать HTML-код существующих сайтов в JSX

# Что такое JSX?

- JSX упрощает написание кода, поскольку очень похож на написание привычного HTML, но только изнутри JavaScript. В сочетании с Node это обеспечивает очень последовательный рабочий процесс.



# Что такое JSX?

- JSX не требуется использовать React - вы можете просто использовать простой JS, но это очень мощный инструмент, который упрощает определение древовидных структур и назначает атрибуты, поэтому я настоятельно рекомендую его использование.
-

- Чтобы визуализировать HTML-тег в React, просто используйте имена тегов в нижнем регистре с некоторым JSX следующим образом:
- 
- `//className` is used in JSX for class attribute
- `var fooDiv = <div className="foo" />;`
- `// Render` where `div#example` is our placeholder for insertion
- `ReactDOM.render(fooDiv, document.getElementById('example'));`

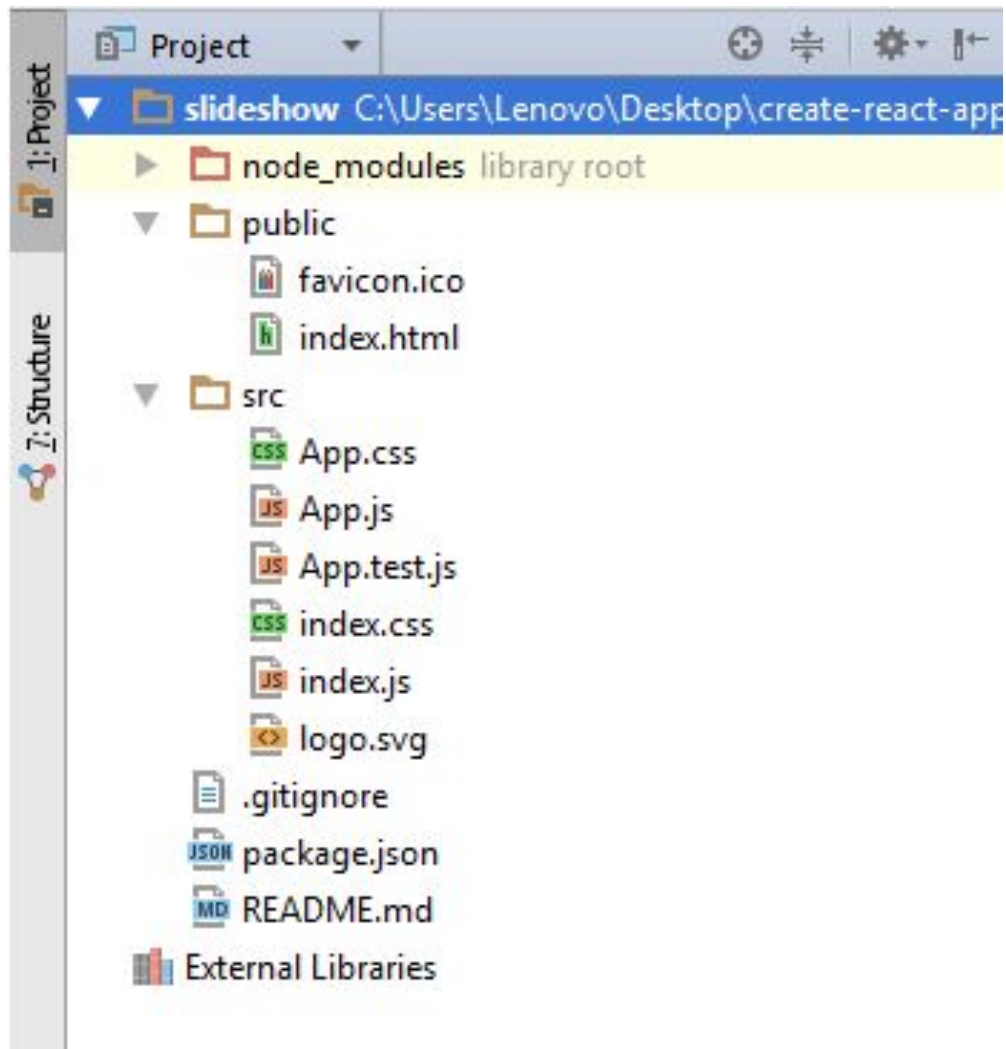
# Создаем проект через create-react-app

- `npm install -g create-react-app`
- `create-react-app my-app`
- 
- `cd my-app`
- `npm start`
-

- Success! Created my-app at `/home/user/workspace/react/my-app`
- Inside that directory, you can run several commands:
  - 
  - `npm start`
    - Starts the development server.
  - 
  - `npm run build`
    - Bundles the app into static files for production.
  -

- drwxrwxr-x 5 user user 4096 май 7 16:27  
my-app
- user@minty ~/workspace/react \$ cd my-app/
- user@minty ~/workspace/react/my-app \$ ls -la
- total 132
- drwxrwxr-x 5 user user 4096 май 7 16:27 .
- drwxrwxr-x 3 user user 4096 май 7 16:23 ..
- -rw-rw-r-- 1 user user 218 май 7 16:27  
.gitignore
- drwxrwxr-x 755 user user 28672 май 7 16:27  
node\_modules
- -rw-rw-r-- 1 user user 367 май 7 16:27

# Что в проекте?



# Структура проекта

- `node_modules/`—в этой папке лежат все зависимости проекта, перечисленные в `package.json` и устанавливающиеся при запуске `npm install`.
- `public/`—содержимое этой папки—это то, что нужно чтобы отрендерить страницу с приложением: `public/index.html`—шаблон приложения и `favicon.ico`—фавиконка.

# Структура проекта

- `src/`—тут лежат все исходники, т.е. те файлы, которые мы будем непосредственно изменять.
- 
- Пожалуй, самое важное, что здесь лежит—это `src/index.js`—входная точка нашего проекта.



# Package.json

- {
- "name": "my-app",
- "version": "0.1.0",
- "private": true,
- "dependencies": {
- "react": "^16.3.2",
- "react-dom": "^16.3.2",
- "react-scripts": "0.9.5"
- },
- "devDependencies": {},
- "scripts": {
- "start": "react-scripts start",

# Индекс

- `import React from 'react';`
- `import ReactDOM from 'react-dom';`
- `import App from './App';`
- `import './index.css';`
- 
- `ReactDOM.render(`
- `<App />,`
- `document.getElementById('root')`
- `);`

- `render()`—метод ReactDOM, который принимает 2 аргумента: компонент, который надо отрендерить, и обертку, в которую надо “положить” результат. В нашем случае в качестве обертки будет элемент с `id="root"`, который задан в `public/index.html`.

# Странный тэг

- `<App />`
- 
- это компонент `App.js`, объявленный при помощи JSX.

# Запуск

- Npm start

Compiled successfully!

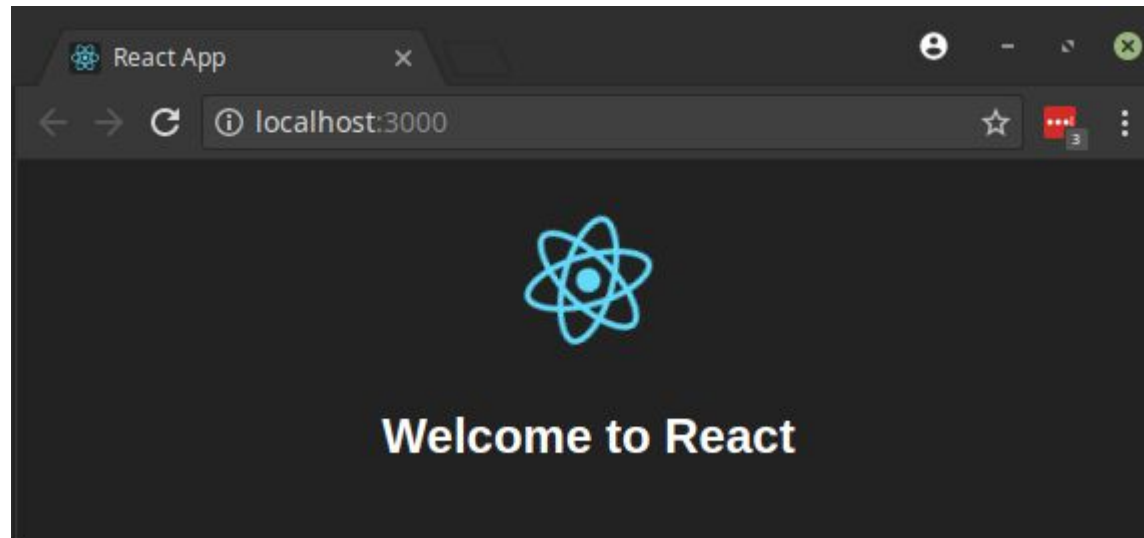
The app is running at:

<http://localhost:3000/>

Note that the development build is not optimized.

To create a production build, use `npm run build`.

# ЧТО МЫ ВИДИМ



To get started, edit `src/App.js` and save to reload.

# Фичи построения

- В дев-среде работает livereload, то есть все изменения автоматически будут отображаться на странице без ее перезагрузки.
- Если нужно зафиксировать (построить) в продакшн
- `npm run build`

# React-компоненты

- class HelloMessage extends React.Component
- {
- render() {
- return (
- <div>
- Hello {this.props.name}
- </div>
- );
- }
- }
- 
- ReactDOM.render(



# ЖИВОЙ КОМПОНЕНТ

- class Timer extends React.Component {
- constructor(props) {
- super(props);
- this.state = { seconds: 0 };
- }
- 
- tick() {
- this.setState(prevState => ({
- seconds: prevState.seconds + 1
- }));
- }
-

# Еще несколько ссылок

- <http://jsraccoon.ru/react-intro>
- <https://code.tutsplus.com/ru/articles/getting-started-with-react--cms-25031>
- <https://reactjs.org/tutorial/tutorial.html>