



# Working with React



Mohamed Mukthar Ahmed

1



## Working with React



### Outline

First React App

Setting Up React

Easy Way

CDN Links

Quickly Try JSX

Our React App

The index.html File

The index.js File

Mohamed Mukthar Ahmed

2

## Our First React



- **React** is not a scary library to work with.
- Let's see how we can setup react.
  - Easy way
  - Correct way
- Easy way of setting up React
  - Both **React** and **ReactDOM** are available over a CDN
  - The quickest way to try **JSX** is to add the **Babel**. The URL for that is also available
    - Now we can use JSX in any script tag by adding **type=text/babel**

Mohamed Mukhtar Ahmed

3

## Setup



### CDN Links

Both React and ReactDOM are available over a CDN.

```
<script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
<script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
```

The versions above are only meant for development, and are not suitable for production.  
Minified and optimized production versions of React are available at:

```
<script crossorigin src="https://unpkg.com/react@18/umd/react.production.min.js"></script>
<script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.production.min.js"></script>
```

Mohamed Mukhtar Ahmed

4

## Setup



React

Docs

Tutorial

Blog

Community

Search

### Quickly Try JSX

The quickest way to try JSX in your project is to add this `<script>` tag to your page:

```
<script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>
```

Now you can use JSX in any `<script>` tag by adding `type="text/babel"` attribute to it. Here is an [example HTML file with JSX](#) that you can download and play with.

This approach is fine for learning and creating simple demos. However, it makes your website slow and **isn't suitable for production**. When you're ready to move forward, remove this new `<script>` tag and the `type="text/babel"` attributes you've added. Instead, in the next section you will set up a JSX preprocessor to convert all your `<script>` tags automatically.

### Add JSX to a Project

Adding JSX to a project doesn't require complicated tools like a bundler or a development server. Essentially, adding JSX **is a lot like adding a CSS preprocessor**. The only requirement is

Mohamed Mukhtar Ahmed

5

## The index.html File



```
File Edit Selection View Go Run Terminal Help index.html - MyFirstProject - Visual Studio Code

index.html X
index.html > ...
1 <html>
2   <head>
3     <link rel="stylesheet" href="styles.css">
4     <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
5     <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
6     <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>
7   </head>
8   <body>
9     <div id="root"></div>
10    <script src="index.js" type="text/babel"></script>
11  </body>
12 </html>
13
```

Mohamed Mukhtar Ahmed

6



## The index.js File

- Since we have included the required libraries in **index.html** we have access to the global object **ReactDOM**
- We can write **declarative** code similar to HTML inside the **ReactDOM.render()** method
  - The **ReactDOM.render()** takes TWO arguments
    - What do you want to render?
    - Where do you want to render?

```

File Edit Selection View Go Run Terminal Help index.js - MyFirstProject - Visual Studio Code
index.html JS index.js x
JS index.js
1 ReactDOM.render(<h1>Hello, everyone!</h1>, document.getElementById("root"))
2
  
```

Mohamed Mukthar Ahmed

7



## The index.js File

```

File Edit Selection View Go Run Terminal Help index.js - MyFirstProject - Visual Studio Code
index.html JS index.js x
JS index.js
1 ReactDOM.render(<h1>Hello, everyone!</h1>, document.getElementById("root"))
2
  
```

Mohamed Mukthar Ahmed

8



## What you'll learn

JSX State HTTP  
 Event Handling **Components**  
 Props Styling Forms  
 More! Lifecycle Methods

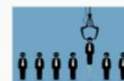
Mohamed Mukhtar Ahmed

9



## Why React?

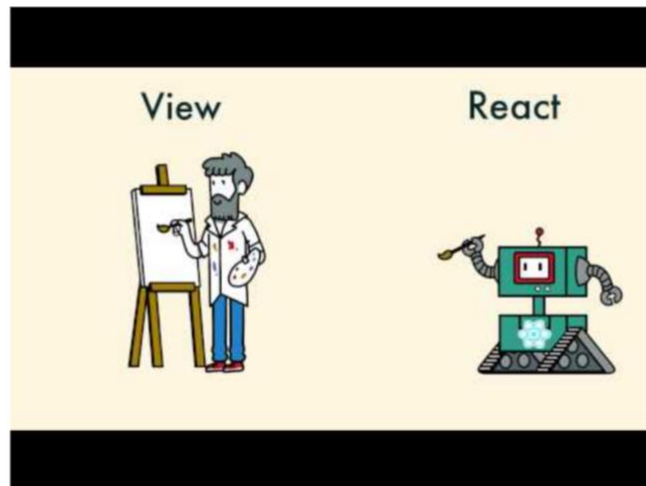
- Virtual DOM
- Reusable (and clearer) Web Components
- Maintained by Facebook
- Hireable



Mohamed Mukhtar Ahmed

10

## Virtual DOM



Mohamed Mukhtar Ahmed

11



## Why React?

It's composable!



Mohamed Mukhtar Ahmed

12



## Why React?

It's composable!



Mohamed Mukthar Ahmed

13



```

<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Navbar</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent"
    aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>

  <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item active">
        <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Link</a>
      </li>
      <li class="nav-item dropdown">
        <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button"
          data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
            Dropdown
          </a>
        <div class="dropdown-menu" aria-labelledby="navbarDropdown">
          <a class="dropdown-item" href="#">Action</a>
          <a class="dropdown-item" href="#">Another action</a>
          <div class="dropdown-divider"></div>
          <a class="dropdown-item" href="#">Something else here</a>
        </div>
      </li>
      <li class="nav-item">
        <a class="nav-link disabled" href="#">Disabled</a>
      </li>
    </ul>
    <form class="form-inline my-2 my-lg-0">
      <input class="form-control mr-sm-2" type="search" placeholder="Search" aria-label="Search">
      <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>
    </form>
  </div>
</nav>

```

Web Components

Mohamed Mukthar Ahmed

14



```
<body>  
  <MySweetNavbar />  
  <MainContent />  
  <MySweetFooter />  
</body>
```

Web Components

Mohamed Mukhtar Ahmed

15



## Why React?

It's composable!

It's declarative!

### Declarative

What should be done?

"Just tell me what to do, and I'll worry about how I get it done"

### Imperative

How should it be done?

"Describe to me every step on how to do something, and I'll do it"

Mohamed Mukhtar Ahmed

16