

# REACT INTRODUCTION

APRIL 2022

YONATAN BENEZRA

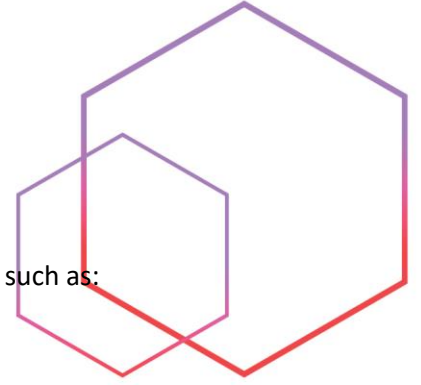
- DO NOT DISTRIBUTE –

- DO NOT COPY – ALL RIGHTS RESERVED TO THE  
AUTHOR –

## About React

- JavaScript library which was written by Facebook in 2013.

- It speeds up and simplifies the frontend development process.
- It has many key features that aid in the development of user interfaces, such as:
  - JSX
  - Components
  - The virtual DOM (which brings us to what we have learned)



## React Components

With React, our webpages will be broken into chunks called components, which will allow us to write their display and logic code all in one.

This will increase the modularity of our user interfaces and enable us to compose components in infinite combinations and reuse them over and over again.

On many web pages, you will have similar user interface components over and over again. Think, for example, of Facebook's like button. It appears on posts, videos, and pictures.

If you were using standard HTML, each one of those buttons would need to be:

- written separately
- given a class in order to add styling to them
- have JavaScript code that would need to be written to implement their behaviors.
- If you needed to update these buttons, the code would likely need to be changed in many different places.

React's component-based architecture saves us the trouble and allows us to reuse the button over and over again, and to update the code in one place only, when needed.

## Installation & Setup

First, you will need Node.js.

Enter your code file and go to the file you would like to create your first React app with. Open up your command line and run:  
`npx create-react-app color-switcher.`

Heads up, this command may take a few minutes to run.

- `npx` is a tool that comes with Node which allows you to run commands without installing them first.
- We are using it to run `create-react-app` which does what the name implies! It scaffolds a React application for us and sets up Babel and Webpack, two super important tools we'll come back to later in the tutorial.
- Finally, `color-switcher` is the name of our application. You will need to re-run the `npx create-react-app your-app-name` command for each app you wish to build.

20 centuries later... Once your app is created, `cd` into the directory that `create-react-app` created for us.  
`cd color-switcher`

## Create React App Generated Files

Open the directory in your Text Editor (Visual studio code) and inspect the files that were created. There are a bunch! But don't be intimidated, let's do a quick run-through of the files create-react-app generated. The highlighted files are the ones that will be important to us:

- node\_modules/ - These are the Node modules we will use for our project. This is essentially other people's code that is going to make our life easier.
- public/
  - favicon.ico - the icon that shows up when you "favorite" a site.
  - index.html - the file our JavaScript attaches to
  - logo.png files - different icons for our app
  - manifest.json - allows us to turn our app into a progressive web app. We will not be doing this today.
  - robots.txt - declares the rules for bots on our site
- src/ :
  - App.css - Where our App component's styling will go
  - App.js - Where our App component will be written
  - App.test.js - A file where you can write automated tests for your App component
  - index.css - A file for global styles for your application
  - index.js - Configuration for your React application
  - logo.SVG - a React logo
  - serviceWorker.js - code to configure
  - setupTests.js - configuration for automated testing
- .gitignore - files you want to keep local and hide from git
- package.json - holds project metadata
- package-lock.json - auto-generated file to track dependencies.

Although many files have been created, we only need to worry about a very few of them.

## The index.js File

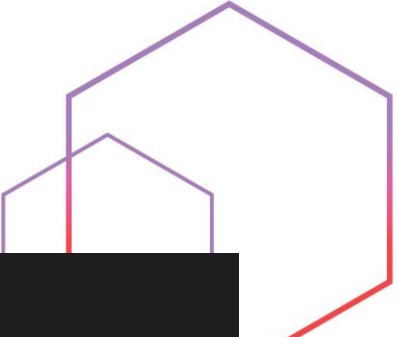
First, let's break down what is in the index.js:

- import React from 'react' - import the React library, We will need this in every file where we use JSX syntax.
- import ReactDOM from 'react-dom' - ReactDOM allows us to use React in the browser.
- import './index.css' - apply the global styles.
- import App from './App' - import the App component

Now for the important code! We are going to take our <App> component and attach it to an element with the id #root. You can find that element in the public/index.html file. This allows our React code to render on the page.

React utilizes the virtual DOM, which is a virtual representation of the DOM that you would normally interact with within Vanilla JavaScript. This ReactDOM.render renders this virtual DOM to the actual DOM.

Behind the scenes, React does a lot of work to efficiently edit and re-render the DOM when something on the interface needs to change.



```
ReactDOM.render(  
  <React.StrictMode>  
    <App />  
  </React.StrictMode>,  
  document.getElementById('root')  
)
```

The `<React.StrictMode>` component is a tool that will give you extra warnings about issues that may exist in your code.

Again, we'll just ignore the service worker code, it goes above and beyond what we need to do. You don't need to change anything in the `index.js` file. I just wanted to run through it since it is so key to making our application work.

