



ReactJS: Hello World App Setup

HARI BABU MUTCHAKALA

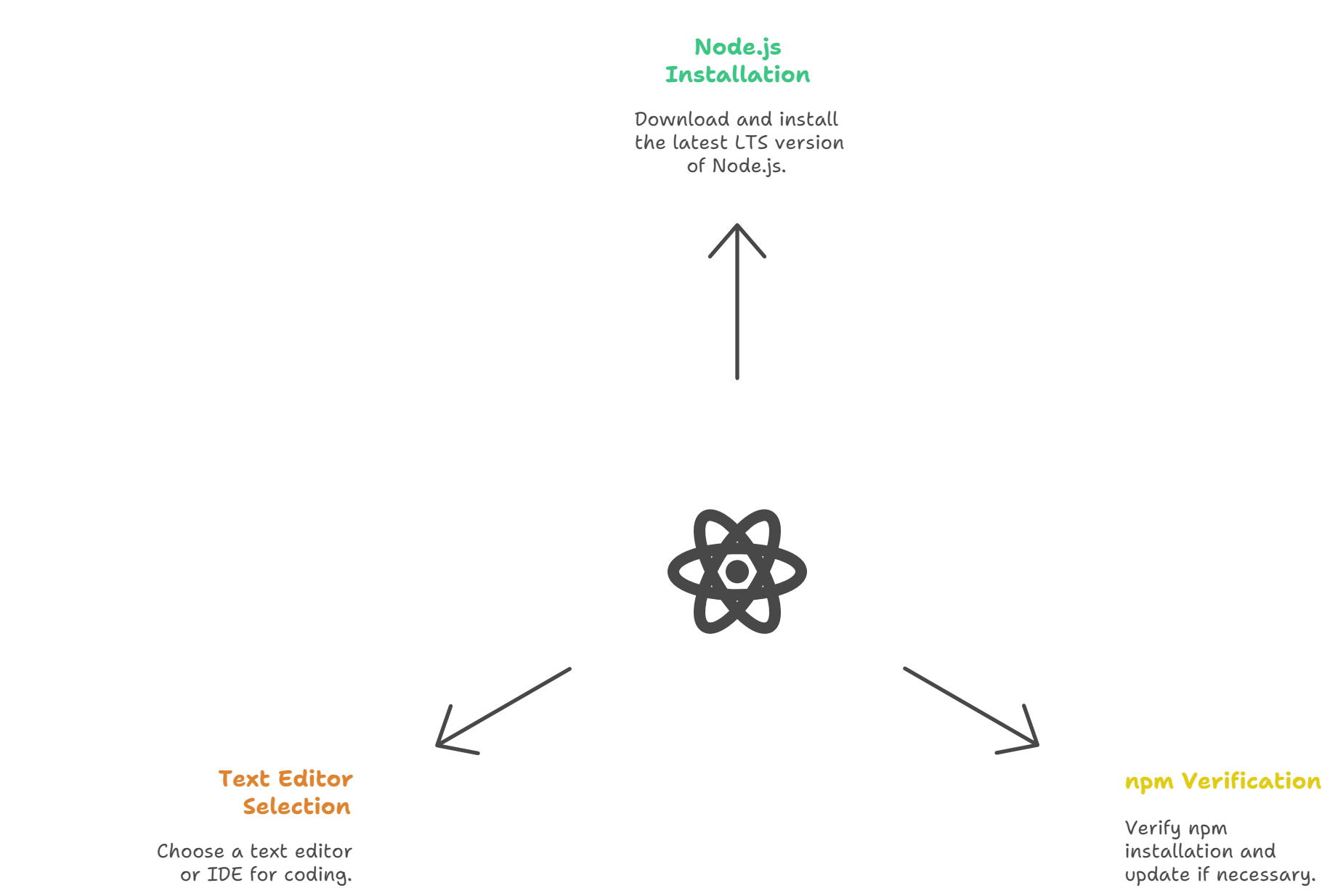
This document provides a step-by-step guide to setting up a basic "Hello, World!" application using ReactJS. It covers the necessary prerequisites, project initialization using Create React App, modification of the default app to display "Hello, World!", and running the application. This guide is intended for beginners who are new to ReactJS and want to quickly get started with a simple project.

Prerequisites

Before you begin, ensure you have the following installed on your system:

- **Node.js:** React requires Node.js to run. Download and install the latest LTS version from the official Node.js website: [https://nodejs.org/][https://nodejs.org/]
- **npm (Node Package Manager):** npm comes bundled with Node.js. You can verify its installation by running `npm -v` in your terminal. If you need to update npm, run `npm install -g npm@latest`.
- **Text Editor:** Choose a text editor or IDE for writing your code. Popular options include Visual Studio Code, Sublime Text, Atom, or WebStorm. Visual Studio Code is highly recommended due to its excellent React support and extensions.

React Development Setup

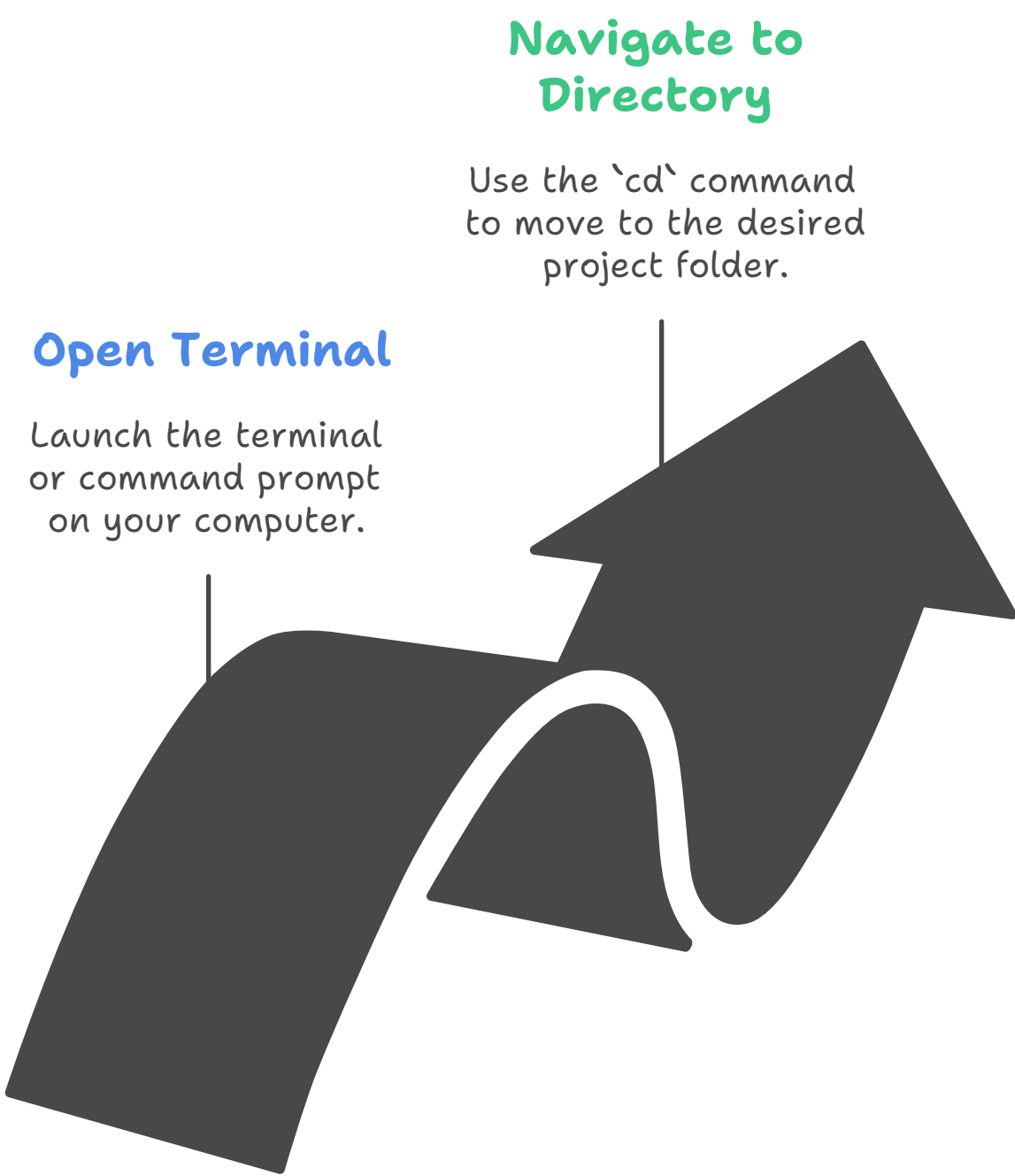


Creating a New React App

The easiest way to create a new React application is by using Create React App [CRA]. CRA is a command-line tool that sets up a new React project with a sensible default configuration.

- 1. **Open your terminal or command prompt.**
- 2. **Navigate to the directory where you want to create your project.** For example, you might have a projects folder in your home directory. Use the cd command to navigate there:

Setting Up React Project



Made with  Napkin

```
```bash
```

```
cd ~/projects
```

```
```
```

3. Run the following command to create a new React app named "hello-world":

```
```bash
```

```
npx create-react-app hello-world
```

```
```
```

`npx` is a tool that comes with npm and allows you to run packages without installing them globally. Create React App will scaffold a new project with all the necessary dependencies and configurations. This process may take a few minutes.

4. Navigate into the newly created project directory:

```
```bash
```

```
cd hello-world
```

```
```
```

Modifying the App to Display "Hello, World!"

Now that you have a basic React app set up, you can modify it to display the "Hello, World!" message.

1. **Open the src/App.js file in your text editor.** This file contains the main component of your application.

2. **Replace the existing code in App.js with the following:**

```
```javascript
```

```
import React from 'react';
```

```
import './App.css';
```

```
function App() {
```

```
 return (
```

```
<div className="App">
```

```
<header className="App-header">
```

```
<h1>Hello, World!</h1>
```

```
</header>
```

```
</div>
```

```
);
```

```
}
```

```
export default App;
```

```
...
```

**\*\*Explanation:\*\***

\* ``import React from 'react';``: Imports the React library, which is essential for creating React components.

\* ``import './App.css';``: Imports the CSS file associated with this component. You can modify this file to style your application.

\* ``function App() { ... }``: Defines a functional component named ``App``. This is the main component that will be rendered.

\* ``return ( ... )``: Returns the JSX (JavaScript XML) that defines the structure of the component.

\* ``<div className="App"> ... </div>``: A container div with the class name "App".

\* ``<header className="App-header"> ... </header>``: A header element with the class name "App-header".

```
* `<h1>Hello, World!</h1>`: An `<h1>` heading element that displays the
 "Hello, World!" message.
```

```
* `export default App;`: Exports the `App` component so that it can be used
 in other parts of the application.
```

3. **[Optional] Modify the CSS:** You can also modify the `src/App.css` file to change the styling of your application. For example, you can change the background color or font size. However, this is not necessary for the basic "Hello, World!" example.

## Running the Application

Now that you have modified the `App.js` file, you can run the application to see the "Hello, World!" message in your browser.

1. **In your terminal, make sure you are still in the hello-world directory.**
2. **Run the following command to start the development server:**

```
```bash
```

```
npm start
```

```
```
```

This command will start a development server and automatically open your default web browser to the address `http://localhost:3000`. If your browser doesn't open automatically, you can manually navigate to this address.

3. **You should see the "Hello, World!" message displayed in your browser.**

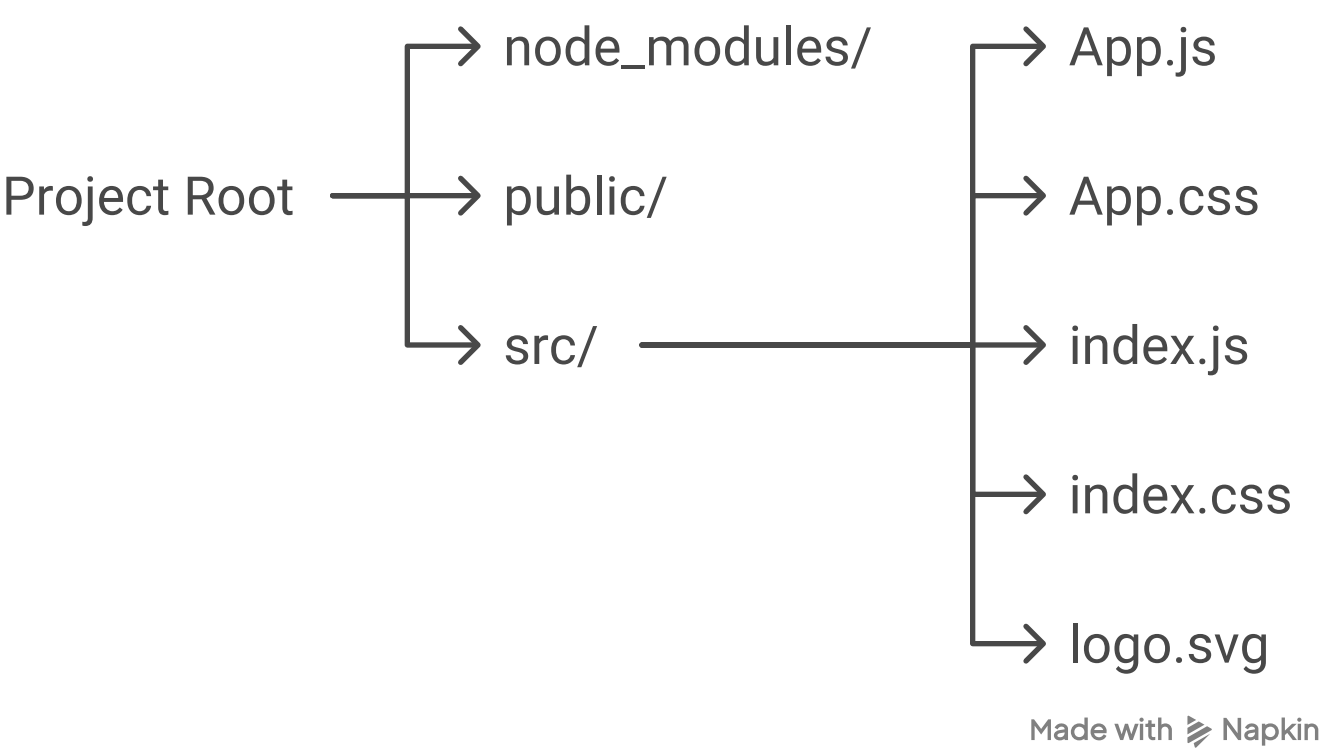
## Understanding the Project Structure

Create React App sets up a well-organized project structure. Here's a brief overview of the key files and directories:

- `node_modules/`: This directory contains all the dependencies installed by npm. You typically don't need to modify anything in this directory.
- `public/`: This directory contains static assets such as the `index.html` file, which is the entry point of your application.
- `src/`: This directory contains the source code for your React components.
  - `App.js`: The main component of your application.
  - `App.css`: The CSS file associated with the `App` component.
  - `index.js`: The entry point for rendering the React application into the `index.html` file.
  - `index.css`: Global CSS styles for the application.

- logo.svg: The React logo.

## React Project Structure



## Conclusion

Congratulations! You have successfully set up a basic "Hello, World!" application using ReactJS. This guide provides a foundation for building more complex React applications. You can now explore the React documentation and experiment with different components and features to expand your knowledge. Remember to consult the official React documentation at [\[https://reactjs.org/\]](https://reactjs.org/)[\[https://reactjs.org/\]](https://reactjs.org/) for more in-depth information and examples.

# From Beginner to React Developer

