1. **What Is package.json?(with full description)**

(<https://heynode.com/tutorial/what-packagejson/#:~:text=Your%20package.,dependencies%20required%20by%20the%20application>. )

1. **React Bundler**

(<https://www.dhiwise.com/post/embark-an-enlightening-journey-with-react-bundler> )

1. **Parcel-Bundler**

(<https://github.com/parcel-bundler/parcel> )

1. **Dependencies**

(<https://medium.com/@reemshakes/devdependencies-vs-dependencies-in-reactjs-db7261e13012#:~:text=When%20your%20project%20needs%20code,list%20of%20your%20project's%20dependencies>. )

1. **What's the difference between tilde(~) and caret(^) in package.json?**

(<https://stackoverflow.com/questions/22343224/whats-the-difference-between-tilde-and-caret-in-package-json> )

1. **[Package.json vs Package-lock.json]**
2. ***What is the purpose of package.json?***
3. ***Tilde (~) and carat (^), and their difference***
4. ***What is package-lock.json?***
5. ***What is the purpose of package-lock.json?***
6. ***Comparing package.json and package-lock.json***
7. ***What is the role of npm-shrinkwrap.json in versioning?***

(<https://www.atatus.com/blog/package-json-vs-package-lock-json/> )

1. **Babel in React**

( <https://www.scaler.com/topics/react/what-is-babel-in-react/> )

1. **Difference Between HTML and JSX**

( <https://www.freecodecamp.org/news/html-vs-jsx-whats-the-difference/> ) ( <https://codersera.com/blog/react-functional-components/> )

1. **Functional Componet**

( <https://codersera.com/blog/react-functional-components/> )

**Config-driven-UI**

[**https://www.freecodecamp.org/news/javascript-optional-chaining/**](https://www.freecodecamp.org/news/javascript-optional-chaining/)

**map array**

<body>

    <div *id*="root"></div>

    <script>

*// Creating HTML tag using Javascript*

*const* heading = document.createElement("h1");

      heading.innerHTML = "Hello World From Javascript!";

*const* root = document.getElementById("root");

      root.appendChild(heading);

    </script>

  </body>

Q:-How does browser understand what is document,createElement,getElementById all these things how browser can Understand??

* Browser has Javascript Engine that exectute this JavaScript
* But Browser Don’t Understand React so first we need to get react into our project

# There is 2 way of adding react to our project

* Via CDN links
* npm install react / npm i react

CDN Links

<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>

1 and 2 links are different for a reason check BookPage-11

Q:- What is CORS (Cross-Origin Resource Sharing)

* <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>

# Creating HTML Tag Element in React

<body>

// Before root and after root all element work fine only affected those inside the root by react (but what is rendering matter, I’m just giving example of root)

    <div *id*="root">

*//Whatever inside the id root will be replaced after root.render*

</div>

    <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>

    <script>

*// Its the Job of 1st CDN Link react*

*const* heading = React.createElement("h1", {}, "Hello World From React!");

*// Its the Job of 2nd CDN Link react-dom*

*const* root = ReactDOM.createRoot(document.getElementById("root"));

      root.render(heading);

    </script>

</body>

* *const* heading = React.createElement("h1", {}, "Hello World From React!");
* this heading Element at the end of the is Javascript Object
* root.render(heading);
* this render method is basically responsible to take this “heading” object and put it up, Convert it into the “HTML heading tag” and put it up(root i.e mention in code) on the DOM

# NPM is not Node package Manager, in npm’s official website there is no place where it has written that npm is Node Package Manager.

NPM Does not have Full Form 😅, so basically npm manages packages but it does not stand for Node Package Manager..

# **Initializing a New Project**

To create a new project, navigate to your desired project folder and run the following command:

$ npm init

This command will prompt you to enter some basic information about your project, such as the name, version, description, and entry point. Once you've provided the required information, ***npm will generate a package.json file***, which will contain all your project's metadata and dependencies.

# **Understanding package.json**

The package.json file is the heart of your project, as it stores all the necessary information about your project, such as its name, version, description, dependencies, and more. Here's a simple example of a package.json file:

{

"name": "my-project",

"version": "1.0.0",

"description": "A simple example project",

"main": "index.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

"dependencies": {

"express": "^4.17.1"

}

}

In this example, the project is named "my-project" and has a single dependency: the Express.js framework. The dependencies object lists all the packages required for your project to run correctly.

# **what-is-npm-and-how-does-it-work**

( <https://reintech.io/blog/what-is-npm-and-how-does-it-work> )

# **Package.json** is Configuration for npm

(<https://heynode.com/tutorial/what-packagejson/#:~:text=Your%20package.,dependencies%20required%20by%20the%20application>) ( <https://reintech.io/blog/what-is-npm-and-how-does-it-work> )

# **React Bundler**

(<https://www.dhiwise.com/post/embark-an-enlightening-journey-with-react-bundler> )

* Parcel Bundler Ignite our App
* npm install -D parcel
* here -D is devDependency(search on net for more info)
* These are your development dependencies. Dependencies that you need at some point in the development workflow but not while running your code (e.g. Babel or Flow).