NPM usage

```
npm init # create a package.json
npm install package-name # install package
# & add --save to also add in package.json
npm install --save package-name
npm install # Install all req's in package.json
node file.js # Run a given file with node
```

Node Terminology

Node.js Run-time spin-off from Chrome that allows us to use node on the "server side" (e.g. in the terminal)

NPM Node Package Manager - used for downloading JavaScript packages, and other JS-dev related tasks.

dependencies Packages that are required to be pre-installed for a given package to be able to run.

`node_modules` Contains your downloaded dependencies.

`package.json` Contains meta-data, dependencies, and NPM configuration about a project.

React Terms

JSX JavaScript variant allowing HTML in JS

Webpack Tool that combines and processes JS, CSS and other static assets

Babel Tool that compiles or translates from one JS variant to another

render React renders templated HTML to the page.

state Keep the variables that change in state, and modify them with setState to rerender your page

JSX Examples

```
// Single HTML element
const paragraph = (
    <p>Lorem ipsum</p>
);

// JSX Templating
const color = "green";
const pWithTemplating = (
    <p>Favorite color: {color}</p>
);

// Loops within JSX
const data = ["alice", "bob", "candice"];
const divOfParagraphs = (
    <div>
        {
            data.map(item => (
                <p>Name: {item} </p>
            ))
        }
    </div>
);
```

useState hook

```
// Create a state value that is initially set to 0. The
// value is stored in `count` and updated with `setCount`
const [count, setCount] = useState(0)

// Adds one to `count` and then re-renders the component
setCount(count + 1)
```

Full React Example

```
import { useState, useEffect } from "react";

// CSS & images can be included "magically"
import "./App.css";
import sendIcon from "./images/envelope.png";

function App() {
    // Define starting state and set functions
    const [message, setMessage] = useState("");
    const [chatLog, setChatLog] = useState([]);

    // Define functions. Must have for forms.
    function onMessageChange(ev) {
        const value = ev.target.value;
        setMessage(value); // Modify state
    }

    // Called when the page loads to fetch data
    // See useEffect cheatsheet for details
    useEffect(() => {
        fetch("http://some.com/api/")
            .then(response => response.json())
            .then(data => {
                console.log("Data received:", data);
                setChatLog(data.messages);
            });
    }, [])

    // Temporary variables and debugging go here
    let messageCount = chatLog.length;
    console.log("rendering!", messageCount);

    // Return the JSX of what you want visible
    return (


            <div className="App">
                <h1>{messageCount} new messages</h1>
                {
                    chatLog.map(text => (
                        <p>Message: {text}</p>
                    ))
                }
                <input onChange={onMessageChange}
                    value={message} />
                <button onClick={() => alert("Hi")}>
                    <img src={sendIcon} />
                    Send message
                </button>
            </div>


    );
}
```