December 16, 2021

React Interactive

1. npx create react app << node package runner in terminal

Text

Description automatically generated

2. Take a moment to read this and what each npm command means:

Text

Description automatically generated

3. Cd into folder and this is what it looks like

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

4. npm install

5. Generates automatic .gitignore

Graphical user interface, text

Description automatically generated

6. How we get in the browser:

Text, website

Description automatically generated

Text

Description automatically generated

7. src folder: index.js, index.css is lowercase, but App.js is upper case.

8. Index.js: import all the tools you need, resources. Top to bottom. We’re importing react. And importing CSS. Import App.js, what you’re importing is an element, it’s an JSX function.

Text

Description automatically generated

9. .render similar to a return statement. Have 2 arguments. First one is application, whole application. If our whole application exists, load it. 2nd is selectors, select id div root. Get the element and render the component.

10. Can’t access something you don’t import.

11. npm start, to see what they’re even styling.

Graphical user interface, website

Description automatically generated

12. <React.StrictMode> “don’t do this”, doesn’t allow you to run risky operations. The App is the child that is running on the StrictMode.

Text

Description automatically generated

13. Rendering elements,

A screenshot of a computer

Description automatically generated

14. React render class:

A screenshot of a computer

Description automatically generated with medium confidence

15. App.js is a file but the App component, element, is being exported. It is a functional component. JSX element. Similar to JavaScript syntax, except uppercase and import it correctly. Inside of the function, have curly braces. When writing function, we did not write return wrapped in parentheses, this is a React thing, multi line require parentheses. It’s returning an HTML element. Then at the bottom we have the export. We define our component. Exporting and importing comes down to default setup. You can only **“export default” and “import default” component ONCE!** You can also write it at the top, but best practice is to write it at the bottom.

Text

Description automatically generated

16. Building something. If you make a change, your server will notify you.

Text

Description automatically generated

17. Now you can inspect the components:

A screenshot of a computer

Description automatically generated

18. React we use class name. Because class is a reserved word. So if you want to style then use className!!!

Text

Description automatically generated

19. Make a class component then put a button, export it, and render it from App.js. Inside src folder, create a new folder called “components”.

20. Inside components folder, create new file called Form.js

21. rcc on the first line if you have snippets in Form.js, should give you below:

A screenshot of a computer

Description automatically generated with medium confidence

22. Make a form line. Our class is form. Thus we can’t use class. Extends Component: render is the only required method.

Text

Description automatically generated

Text

Description automatically generated

23. Add a title to the form and a button

Text

Description automatically generated

24. Import it and render it:

On the top of App.js: import Form from \*./components/Form.js\*<< the parent

Text

Description automatically generated

25. On the bottom put </Form> to render it. << the child

A screenshot of a computer

Description automatically generated with medium confidence

26. Using the tree model, we can modify App.js but the way it is currently set up, the first child component is the first div, then the second is the form, both are child of index.js. The current problem is that we have the div and the form component rendered side by side. Erasing the div for now should solve the problem. React has an imaginary div.

Text

Description automatically generated

Or the use the React Fragment before and after div tag

Text

Description automatically generated

This is the same as above: **But every React library must be imported!!!! So an import line must be typed out! But if you don’t type out React. Fragment, you don’t need to import React.**

A screenshot of a computer

Description automatically generated with medium confidence

27. Props and state: what got rendered

Graphical user interface, text, application

Description automatically generated

28. Cleared div out in App.js and it looks like this on browser:

Text

Description automatically generated

Graphical user interface, application

Description automatically generated

29. When you render the page this is what happens

A picture containing graphical user interface

Description automatically generated

30. State and Props:

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

A screenshot of a computer

Description automatically generated

If things have changes, it re-renders the change. If not, it doesn’t. React takes js code and using .render method to render it.