smerf.net

Ethereum
Blockchain Development
Week 21

Introduction

The intention of this lab is to look at functions and how variables are stored. All the exercises are completed in Remix IDE or the VM provided.

Code Completion

Writing code in a new language can be a steep learning curve. The approach here is to provide some code with underscores (_) that you are required to complete. These underscores are there to help you. By completing these exercises you will be improving your skills and knowledge of Solidity.

Each exercise starts on a new page. The red numbers in the right-hand margin are estimated minutes you should spend on each exercise.

1 My First React App

5–10

This has to be completed in the VM provided and installed from 4125 VM and follow the instructions below:

1. Open terminal and type: `npx create-react-app my-app2`
2. Type: `cd my-app2`
3. Check: type: `npm start`
4. This should redirect you to a browser.
5. Close this browser down, return to the terminal and push CTRL-Z to cancel the react app.
6. Edit. Type: `code .`

1.1 App

5

Complete the following:

1. Download four files
   (a) App.js
   (b) Header.js
   (c) List.js
   (d) bulma2.css
   (e) Hint: use `wget` for quicker results.
2. Keep these files in download folder
3. Replace the current App.js file, with the recently downloaded file.

1.2 Header

5-10

Add the recently `Header.js` in the `src` folder.

Take a look at this file in Fig. 1.1.

Code Completion

```
import './css/bulma2.css';
function Header(){
    return (
        <>
        <div class="block">
        <section class="hero is-info">
            <div class="hero-body">
                <div class="container">
                    <h1 class="title">
                        Smerf
                    </h1>
                    <h2 class="subtitle">
                        Website with React and Bulma
                    </h2>
                </div>
            </div>
        </section>
        </div>
        <article class="message is-primary">
            <div class="message-header">
                <p> Message Hdr</p>
            </div>
            <div class="message-body">
                Smart miss day which begin. Snug do sold mr smith random.
            </div>
        </article>
        </>
    );
}
export default Header;
```

Figure 1.1: Header element for React.

1.3 List

5-10

Add the recently downloaded `List.js` in the `src` folder.

Take a look at this file in Fig. 1.2

Code Completion

```
function List(){
    return (
        <div class="notification is-primary">
            <ol type="i">
                <li> Carrots </li>
                <li> Broccoli</li>
                <li> Brussels</li>
            </ol>
        </div>

    );
}

export default List;
```

Figure 1.2: List element for React.

1.4 CSS and Deploy

5-10

The final part of this SPA development is the CSS file, follow the instructions below:

1. Create a new sub directory, `css`, in the `src` directory.
2. Add the recently downloaded `bulma2.css` to the `css` directory created above.
3. Return to terminal in the `my-app2` directory and type: `npm start`

2 Properties

This example passes properties from one element to another. 5

Close down any React websites and terminals. Open a new terminal and follow the instructions below:

1. Open terminal and type: `npx create-react-app my-app3`
2. Type: `cd my-app3`
3. Edit. Type: `code .`

2.1 App

Complete the following: 5

1. Download four files
   (a) App.js
   (b) Header.js
   (c) CreateList.js
   (d) bulma2.css
   (e) Hint: use `wget` for quicker results.
2. Keep these files in download folder
3. Replace the current App.js file, with the recently downloaded file.

2.2 Header 5-10

Add the recently `Header.js` in the `src` folder.

Take a look at this file in Fig. 2.1.

Code Completion

```
import './css/bulma2.css';
function Header(){
    return (
        <>
        <div class="block">
        <section class="hero is-info">
            <div class="hero-body">
                <div class="container">
                    <h1 class="title">
                        Smerf
                    </h1>
                    <h2 class="subtitle">
                        Website with React and Bulma
                    </h2>
                </div>
            </div>
        </section>
        </div>
        <article class="message is-primary">
            <div class="message-header">
                <p> Message Hdr</p>
            </div>
            <div class="message-body">
                Smart miss day which begin. Snug do sold mr smith random.
            </div>
        </article>
        </>
    );
}
export default Header;
```

Figure 2.1: Header element for React.

2.3 CreateList 5

Add the recently downloaded `CreateList.js` in the `src` folder. 5-10
Take a look at this file in Fig. 2.2

Code Completion

```
function CreateList(props) {
    let bulmaStyleProp = `notification ${ props.style }`;
    return(
        <div class={bulmaStyleProp}>
            <h2 class="title is-2"> {props.title} </h2>
            <ol type="1">
                {props.items.map( (item) =>
                     <li> {item} </li>
                )}
            </ol>
        </div>
    );
}

export default CreateList;
```

Figure 2.2: List element for React.

2.4 CSS and Deploy

5-10

The final part of this SPA development is the CSS file, follow the instructions below:

1. Create a new sub directory, `css`, in the `src` directory.
2. Add the recently downloaded `bulma2.css` to the `css` directory created above.
3. Return to terminal in the `my-app3` directory and type: `npm start`
4. Add a new array, `meats`, and createList element that displays a list of meats:
   - Lamb
   - Chicken
   - Fish

3 State

5

Complete the following:

1. In the root directory start a new react app by typing: `npx create-app-react my-app4`
2. Change to this directory, `cd my-app4`, and open VSCode by typing `code .`
3. Download five files
   (a) App.js
   (b) Header.js
   (c) CreateList.js
   (d) List.js
   (e) bulma2.css
4. Keep these files in download folder
5. Replace the current App.js file, with the recently downloaded file.

3.1 List

15-20

As in the above exercises, replace or move the files to their appropriate location.

Open a terminal and go to the directory, `cd my-app4`, and start the application, `npm start`.

Check if the changes of state are recorded.

Finally, improve this output by resolving the issue of displaying the previous state in the buttons.

References