Introduction to React

CS5610: Web Development

Joydeep Mitra

Pre-class Activity

- Follow the instructions in the README to setup the repo

  https://github.com/CSE-316-Software-Development/learn-react/tree/main

What is React?

- A JavaScript library for rendering user interfaces (UIs).
- Key Idea =>
  - User interface is built from reusable small units (e.g., button) and combined to form larger units.
  - These units are called components!
- React components are designed to encapsulate interaction and presentation (**interaction-based design**).

Defining Components

- A collection of JavaScript functions sprinkled with HTML-like tags to render content in the browser.
- Consider the example in *src/pages/components/profile.js* and note the following:
  - What is returned from the function?
  - If we remove the parenthesis from *return* will it make a difference?
  - What is special about the name of the function?

Defining Components

- Once defined, a component can be used in different contexts by nesting it in other components.
- An example is shown in *src/components/gallery.js*
  - Why do we see three images now?

For You to Do

1. The code snippet in *src/pages/qcomps/profile_mistake.js* has mistakes. Identify and fix them.
   - Add necessary imports in *src/pages/index.js* to load the component.
   - Run the server and check in the browser.
2. Define a React component in “src/pages/qcomps/firstcomp.js” to render the following HTML content in the browser:

```
<div>
      <h1> My first component </h1>
      <ol>
        <li> Components: UI Building Block</li>
        <li> Defining a Component </li>
        <li> Using a component </li>
      </ol>
</div>
```

JSX Syntax

- JSX is a syntax extension for JavaScript that allows developers to embed HTML elements inside React components.
- Key Syntax rules:
  1. Returns a single Root element from a component.
  2. Tags must be closed explicitly (e.g., `<li> … </li>`). Self-closing tags must be closed as in `<img />`
  3. Attributes must be in camelCase (e.g., `textWidth`, `className`)
  4. Attribute values can be single or double-quoted strings or JavaScript values
     - JavaScript values should be enclosed within {}

For You to Do

- Based on the JSX rules, answer the following questions:
  - Identify and fix all mistakes in the JSX code in “src/pages/qcomps/bios.js”:
  - Modify “src/pages/qcomps/todos.js”, such that the person’s name and their image is displayed. Also, the alt attribute must be set with the person’s name.

Component Props

- Components can be defined with input parameters called **props**.
- Parent components use props to pass information to their child components.

Component Props

- Explore code in “src/pages/components/profile_props.js” for an example of how props are used, and then consider the following:
  - What is the meaning of the input parameter of the function Avatar?
  - What is the meaning of the attributes in <Avatar /> in relation to the function Avatar?
  - Modify index.js and load the components in a browser to see what gets displayed.

Component Props

- Review the code in “src/pages/qcomps/gallery_props.js”. Run it and review how it gets rendered. However, the Gallery component contains some very similar markup for two profiles. Modify the code to extract a Profile component out of it to reduce the duplication. You’ll need to choose what props to pass to it.

Container Components

- A container component is defined to hold several kinds of JSX elements.
- When we nest content within a JSX tag, the parent component receives that content in a prop called **children**.
- Let’s explore the example in “src/pages/components/square.js”.
  - Note how the parent component is used to render its *children*.

Conditional Rendering

- Components often need to display things based on conditions.
- We can conditionally render JSX using JavaScript syntax like if-else statements and ternary operators.
- Let’s explore the example in “src/pages/components/prop_items.js”.

For You to Do

- Modify the component in “src/pages/qcomps/prop_items.js” such that the packed items (packed=true) are striked. You can use <del> text </del> to strike text. Can you use ternary operators to do the same?

Rendering lists

- A common way to render lists is to iterate over elements in an Array and convert each element into a JSX element.
- Let’s explore the code snippet in “src/pages/components/list_plain.js” and consider the following questions:
  - When you run the code and render it in the browser do you see any warnings? What do they mean?
  - Now, compare this with the code in “src/pages/components/list_keys.js”. What difference do you see? Why do the warnings disappear?
  - However, is our choice of key in “src/pages/components/list_keys.js” ideal?

Keys in React Lists

- React uses keys to map an item in the list to the correct component.
- Keys are akin to file names in a file system. Imagine if you didn’t have file names!
- Key rules:
  - The best keys are part of the data.
  - They must be unique
  - Should not be an index of the item in the array.
  - Should be a random number.

For You to Do

- Modify the code in “src/pages/qcomps/list_keys_id.js” such that a list of all images in the object people is displayed without any warnings.

- Modify the code in “src/pages/qcomps/recipes.js” such that it displays an unordered list of ingredients for each dish in recipes object.