Elements and Components

HyperionDev

Muhammad Zahir Junejo

Lecture – Housekeeping

- ❑ The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly.
  - ❑ Please review Code of Conduct (in Student Undertaking Agreement) if unsure
- ❑ No question is daft or silly - **ask them!**
- ❑ Q&A session at the end of the lesson, should you wish to ask any follow-up questions.
- ❑ Should you have any questions after the lecture, please schedule a mentor session.
- ❑ For all non-academic questions, please submit a query: www.hyperiondev.com/support

Lecture Objectives

1. Basics of React.js Projects
2. Creating elements
3. Creating components
4. What is the JSX Syntax?
5. Learn about props

React

❑ Web developers often use libraries and frameworks to create web applications more quickly.

❑ React.js or React is a JavaScript library used for frontend development.

❑ With React, we use JavaScript to write HTML. Therefore it is easy to mix up logic and calculations with the HTML.

❑ JSX: JavaScript XML. JSX is a JavaScript extension syntax that helps to write HTML and JavaScript together easily.

❑ React Elements: Building blocks of React applications.

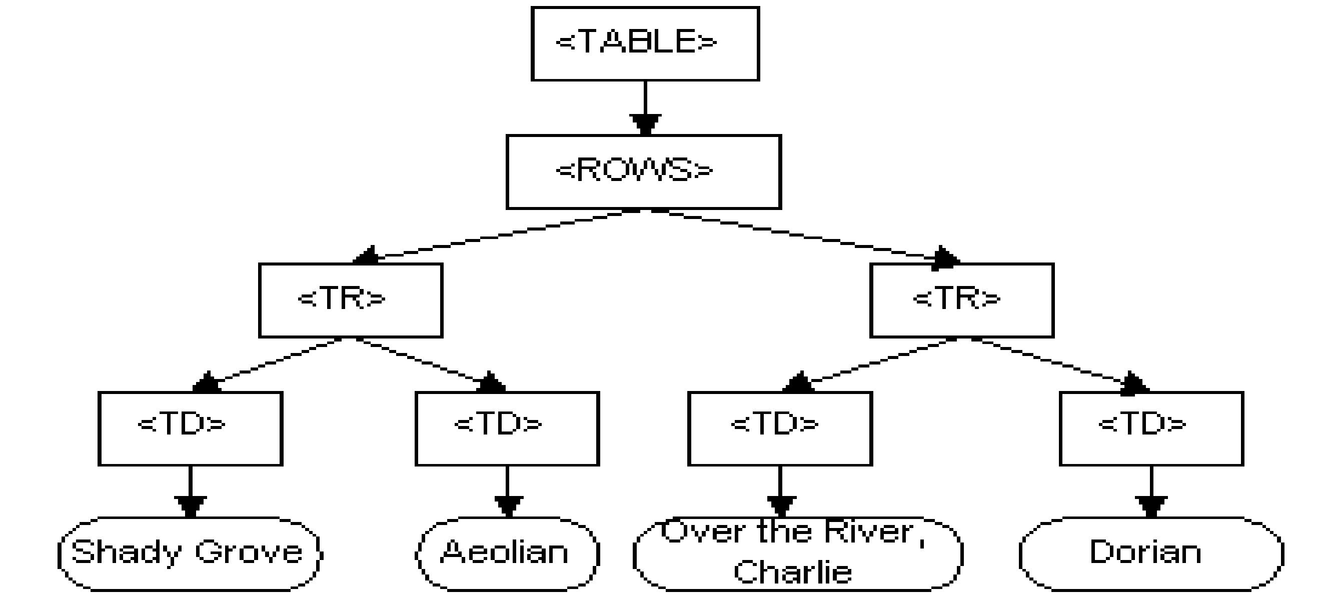
❑ Components: Reusable pieces of code that return an element to render to the page.

❑ Props: Inputs to a React component.

❑ State: Object that holds data related to a Component.

Virtual DOM

- ❑ The DOM is a programming API for HTML documents that defines the logical structure of documents and how they are accessed and manipulated.
- ❑ We can add or remove element from an HTML page using JavaScript, thanks to DOM.



Hyperiondev

Virtual DOM

- ❑ Rewriting the DOM every time the HTML document changes can significantly slow down your web application or site. To address this problem, React uses a virtual DOM.
- ❑ Virtual representation of the HTML document in memory.
- ❑ Works by taking a *snapshot* of the DOM before changes are made.
- ❑ As changes are made, the virtual DOM is updated.
- ❑ After the changes are complete, the virtual DOM and the snapshot of the DOM (which is also saved in memory) are compared to see where changes were made.
- ❑ Instead of rewriting the entire DOM, the real DOM is only updated with the changes that were made.

Elements

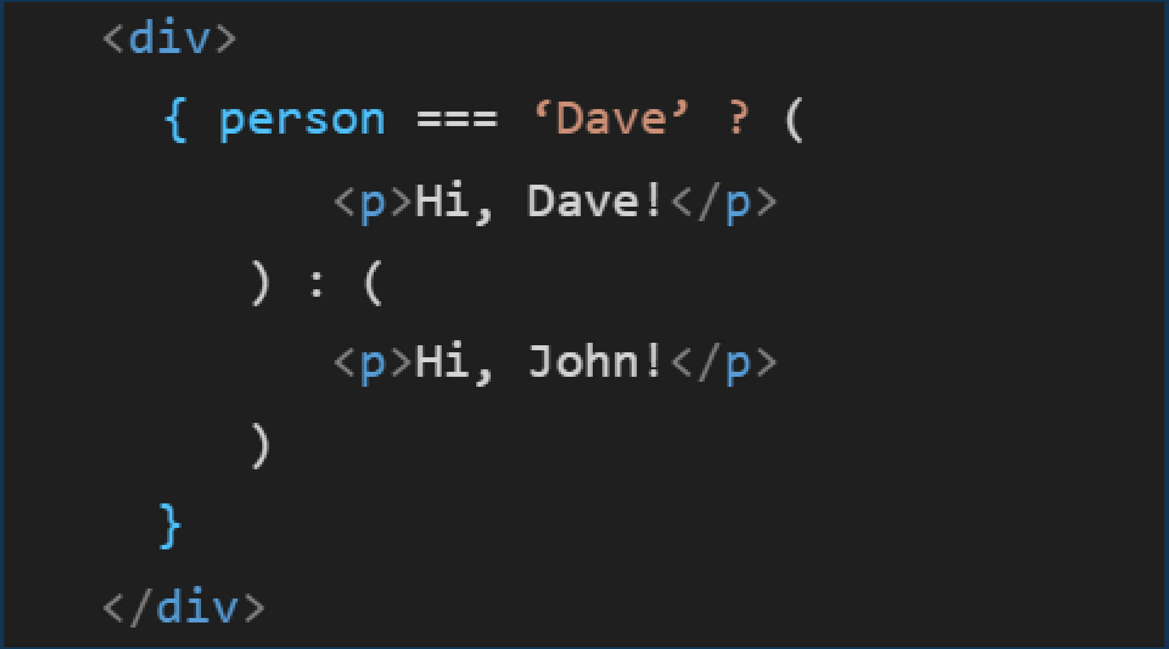
- ❑ Through elements, you basically tell React what you want to see on the screen.
- ❑ React elements are similar to DOM elements.
- ❑ Not used directly, instead they are returned from Components.
- ❑ Creating React elements:
  - ❑ Create the React and ReactDOM objects by importing React and React-DOM
  - ❑ Create the React element using JSX
  - ❑ Place the React element on the **render** function of the component.

JSX

- JavaScript Syntax Extension referred to as JavaScript XML.
- Makes creating React application a lot easier.
- JSX tags have a tag name, attributes and children.
- If an attribute value is enclosed within quotes (“”) it is a string literal however if the attribute value is wrapped by curly brackets ({}), it is considered an enclosed JavaScript expression.

Conditional Rendering

- ❑ Rendering an element or elements on the basis of a condition.
- ❑ Example:



```
<div>
  { person === ‘Dave’ ? (
        <p>Hi, Dave!</p>
     ) : (
        <p>Hi, John!</p>
     )
  }
</div>
```

Rendering arrays

- You can use the Array.map(...) function to render lists to the page.
- Example:

```
let gameList = [“cricket”, “football”, “badminton”];
<div>
        { gameList.map( ( item, index ) => {
          return (
            <li key = { index } > { item } </li>
           )
          })
        }
       </div>
```

Props

- ❑ React components use props to communicate with each other.
- ❑ Every parent component can pass some information to its child components by giving them props.
- ❑ Props might remind you of HTML attributes, but you can pass any JavaScript value through them, including objects, arrays, and functions.

Props

- Props let you think about parent and child components independently. You can think of props like “knobs” that you can adjust.

- Passing props:

```
<Avatar
    person={{ name: 'Lin Lanying', imageId: '1bX5QH6' }}
    size={100} />
```

- Reading props:

```
function Avatar(props) {
  let person = props.person;
  let size = props.size;
}
```

Questions and Answers

References

- ❑ https://react.dev/learn/your-first-component
- ❑ https://react.dev/learn/importing-and-exporting-components
- ❑ https://react.dev/learn/writing-markup-with-jsx
- ❑ https://react.dev/learn/javascript-in-jsx-with-curly-braces
- ❑ https://react.dev/learn/passing-props-to-a-component
- ❑ https://react.dev/learn/conditional-rendering
- ❑ https://react.dev/learn/rendering-lists

Thank You!