Introduction

* JS library for frontend web applications
* DOM: Document Object Model

Adding React

* Add 3 script tags (1 to enable JSX: Syntax extension to JS) to HTML head
* <script src = https://unpkg.com/react@16/umd/react.development.js crossorigin></script>

<script src="https://unpkg.com/react-dom@16/umd/react-dom.development.js" crossorigin></script>

<script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

* Now, we can build react app
* Below code prints “Hello, React!”
* <!DOCTYPE html>

<html>

    <head>

        <title>Test</title>

        <script src="https://unpkg.com/react@16/umd/react.development.js" crossorigin></script>

        <script src="https://unpkg.com/react-dom@16/umd/react-dom.development.js" crossorigin>

        </script>

        <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

    </head>

    <body>

        <div id="container"></div>

        <script type="text/babel">

        ReactDOM.render(

          <h1>Hello, React!</h1>,

          document.getElementById('container')

        )

        </script>

    </body>

</html>

* Here, code finds container div and adds h1 heading to it

Create React App

* You need create react app and node
* Use following commands to create ‘my-app’ react application
* npx create-react-app my-app

cd my-app

npm start

* All CSS and JS files should be added to src folder
* HTML file should be in public folder
* Open src/index.js and change code as follows to change default output
* ReactDOM.render(

<h1>Hello, React!</h1>,

document.getElementById('root')

);

* Here, ReactDOM.render is used to add React app inside element
* React’s render method is used
* Create React App has Fast Refresh feature
* Uses Webpack as bundling tool
* package.json has the list of project dependencies
* Localhost: Local computer

JSX

* JSX allows HTML elements inside JS
* Ex: <h1>Hello, React!</h1>
* We can use any JS expression inside JSX
* const name = "David";

const el = <p>Hello, {name}</p>;

ReactDOM.render(

el,

document.getElementById('root')

);

* JSX has attributes like in HTML
* <div id = “name”></div>
* For JS expression as attributes, don’t use quotes
* <div id = {user.id}></div>
* React DOM uses camelCase property naming convention instead of HTML attribute names
* Ex: class becomes className in JSX
* const x = “myClass”;

const el = <div className = {x}></div>;

* When JSX code is compiled it’s converted to JS objects representing React elements
* React then uses them to build HTML DOM
* let counter = 0;

function show() {

counter++;

const el = <p>{counter}</p>;

ReactDOM.render(

el, document.getElementById('root')

);

}

setInterval(show, 1000);

* In this counter app, only the paragraph text gets updated
* setInterval is used to call show function
* Below code calls calculate function every 5 seconds
* setInterval(calculate, 5000);

Virtual DOM

* React use Virtual DOM (lightweight representation of DOM)
* DOM: Document Object Model

Components