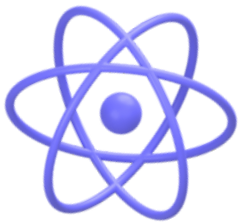


WHAT IS REACT JSX

HIMA VAMSI

Follow for more

Save

SUPERCHARGE YOUR REACT WITH ES6!

Before going to JSX lets understand

what is react ES6

ES6 stands for ECMAScript 6. ECMAScript is a JavaScript standard intended to ensure a common language across different browsers. ES6 is the 6th version of ECMAScript.

WHY ES6?

React uses ES6 and all of these new features will make your coding experience in react much much better. You will be able to do things with much more ease and in very less lines! Features like:

FEATURES OF ES6:

ARROW FUNCTIONS:

```
const hello = () => {
  return "Hello World!";
}
```

```
const hello = () => "Hello World!";
```

"With arrow functions, you don't have to worry about **this** keywords or binding functions!"

ARRAY METHODS:

```
const names = ["Alice", "Bob", "Charlie"];
const greetings = names.map(name => `Hello, ${name}!`);
console.log(greetings);

// Output: ["Hello, Alice!", "Hello, Bob!", "Hello, Charlie!"]
```

"map can be used for alot of things, one of it's use case is, we can make any number of cards through loop and just put it in jsx, like this"

HIMA VAMSI

DESTRUCTURING:

Classic Method:

```
const languages = ['JS', 'Python', 'Java'];
const js = languages[0]
const python = languages[1]
const java = languages[2]
```

Modern Method:

```
const languages = ['JS', 'Python', 'Java'];
const [ js, python, java ] = languages;
console.log(js);
console.log(python);
console.log(java);

//output:
//JS
//Python
//Java
```

“Easily get values from objects, like grabbing values from props or state.”

SPREAD OPERATOR:

```
const web = ['JS', 'Python', 'Java'];
const languages = ['C', 'C++', 'C#']

const allLanguages = [...web, ...languages]

//output:
["JS","Python","Java","C","C++","C#"]
```

CLASSES:

```
const web = ['JS', 'Python', 'Java'];
const languages = ['C', 'C++', 'C#']

const allLanguages = [...web, ...languages]

//output:
["JS","Python","Java","C","C++","C#"]
```

“Classes help you structure your code and organize how components behave.”

HIMA VAMSI

Save

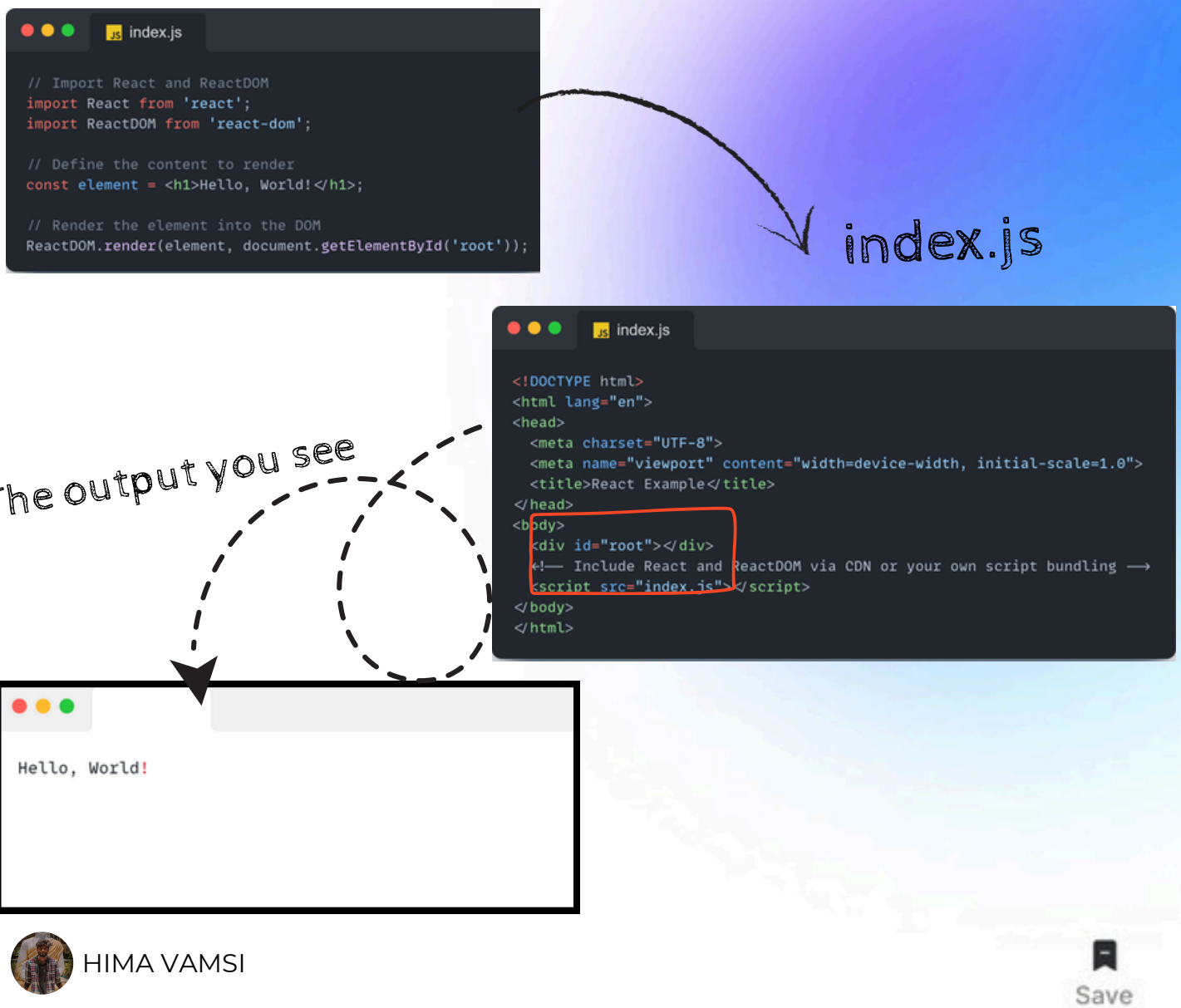
REACT RENDER HTML:

“React renders HTML to the web page by using a function called ReactDOM.render().”

REACTDOM.RENDER():

“This function takes two arguments, HTML content which you want to show on page and HTML element where you want to put the HTML content(first argument).

But where will it find that element? It will find it inside "index.html" located in "public" folder. There you will notice a div with id "root". That is where all this will be rendered! “



HIMA VAMSI

WHAT IS JSX?

JSX (JavaScript XML) is a syntax extension for JavaScript that looks a lot like HTML. It's used in React to describe what the UI should look like. Instead of using React.createElement every time you want to create an element, you can use JSX, which makes the code more readable and intuitive.

WHY USE JSX?

JSX makes it easier to write and visualize React components because it resembles HTML structure. It lets you combine HTML-like tags with JavaScript logic, so you can directly see what the UI will look like

LET'S SAY WE WANT TO DISPLAY A SIMPLE MESSAGE.

WITHOUT JSX:

```
const element = React.createElement('h1', null, 'Hello, World!');
```

WITH JSX:

```
const element = <h1>Hello, World!</h1>;
```

"Both examples do the same thing, but the second one with JSX is easier to understand and much closer to HTML"

- JSX is not HTML, but it looks very similar.
- It allows JavaScript expressions within {} braces. For example:

```
const name = "Alice";
const greeting = <h1>Hello, {name}!</h1>;
```

- JSX must have one parent element. If you have multiple elements, wrap them in a single or a React fragment (<> ... </>).

HIMA VAMSI

Save

BABEL THE JAVASCRIPT COMPILER

Babel is a JavaScript compiler that translates JSX and newer JavaScript features (like ES6+).Babel converts JSX and modern JavaScript into code that browsers can understand, making it compatible across all environments.

When Babel sees JSX code, it converts it into standard JavaScript code using **React.createElement**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>JSX Beginner Example</title>
</head>
<body>
  <!-- React code with Babel to handle JSX -->
  <script type="text/babel">
    const App = () => {
      return <h1>Hello, JSX and Babel!</h1>;
    };

    ReactDOM.render(<App />, document.getElementById('root'));
  </script>
</body>
</html>
```

index.js

```
</head>
<body>
  <div id="root"></div>
  <!-- Include React and ReactDOM via CDN or your own script bundling -->
  <script src="index.js"></script>
</body>
</html>
```

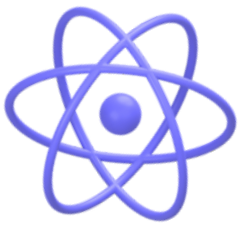
Renders **App** component in root element which is present in the public folder.

HIMA VAMSI

LEARNED SOMETHING NEW?

Follow for more such content like this

HIMA VAMSI



Save