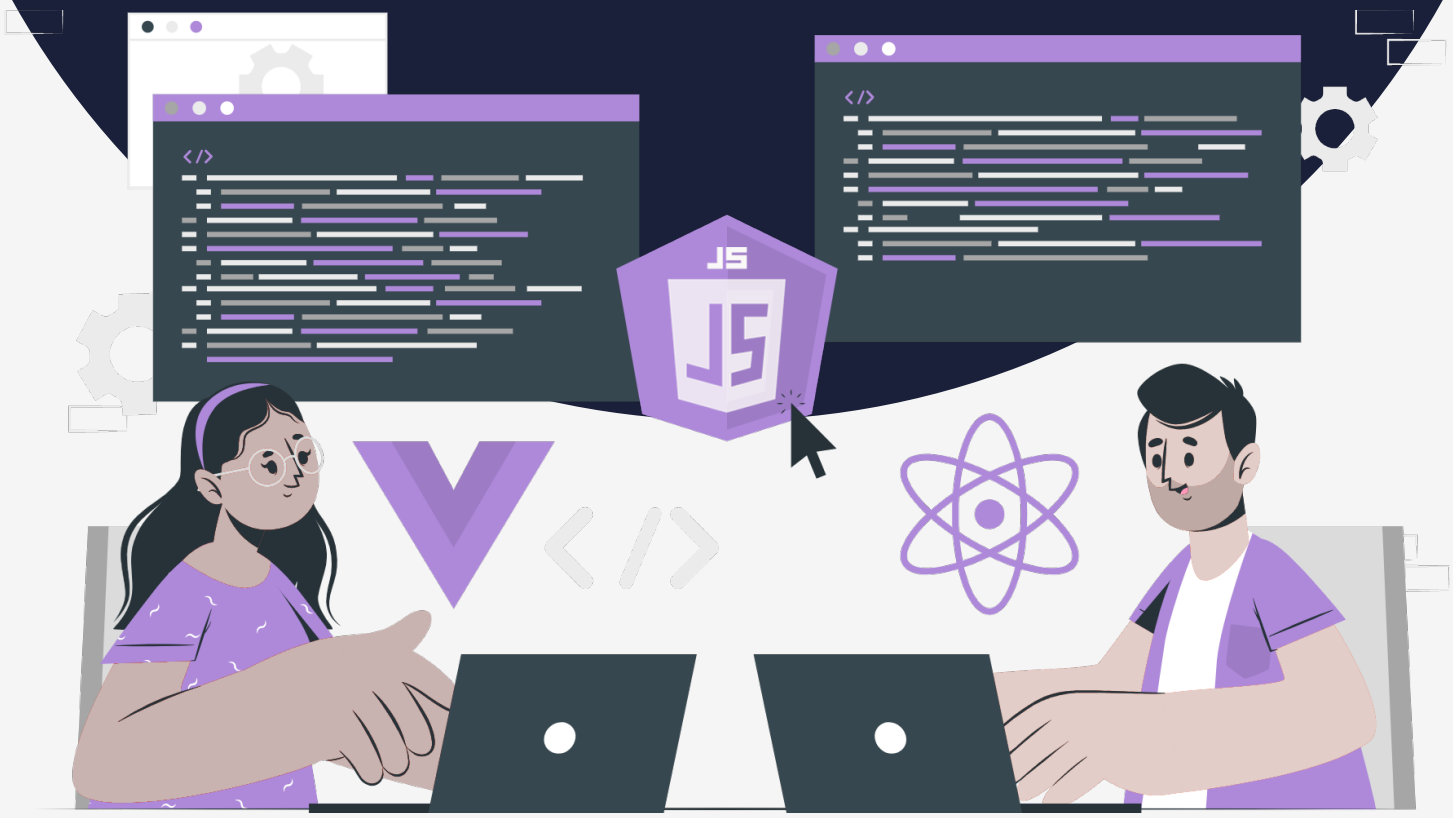
Lesson:

JSX



Topics Covered:

1. What is JSX?
2. Difference between JSX and HTML
3. Why JSX
4. Advantages of Jsx

What is JSX?
Jsx stands for JavaScript XML. It is a syntax extension for JavaScript that allows you to write HTML-like elements in our JavaScript code. It is used in React to describe the structure and content of a component.

One of the main benefits of JSX is that it makes it easy to create and manipulate the structure of a component. For example, instead of manually creating and appending DOM elements to build the structure of a component, you can use JSX to define the structure in a way that is similar to HTML, making it easier to read and understand.

Example of a JSX element:

```javascript
const name = <h1>Hello Bachho </h1>
```

A JSX element is converted into a regular JavaScript object during compilation so that it can be used to construct a genuine DOM element. For instance, the JavaScript generated from the JSX code above might be as follows:

```javascript
const name = React.createElement("h1", null, "Hello Bachho");
```

Basic difference between JSX and HTML

- Multiple elements can be returned in HTML. for e.g;

```javascript
<ul>
  <li>PW Skill
    <ol>
       <li>Web development</li>
       <li>Data Science</li>
       <li>Java with DSA</li>
     </ol>
 </li>
 <li>Physics</li>
 <li>Chemistry</li>
</ul>
```

- Nested JSX must return a single element that wraps all other levels of nested elements, which we'll refer to as the parent element:

```
JavaScript
<div>
   <h1>Welcome to PW SKILL </h1>
   <p>Thank you for choosing us</p>
   <p>Web development </p>
</div>
```

- All HTML attributes and event references in JSX become camelCase, this way, onclick event becomes onClick and onchange — onChange.

Browsers can't read JSX

JSX is a combination of HTML and Javascript..So it is not supported by browsers. So a transpiler called Babel converts the JSX into pure javascript . Then browsers understand the code and execute it.Browsers can't read JSX because there is no inherent implementation for the browser engines to read and understand them.

Why JSX?

With the help of JSX, we can write HTML code with javascript.React does not employ the createElement() method; instead, JSX elements are used to create HTML elements. As a result, JSX facilitates the writing and addition of HTML components in React. A transpiler called babel.js will convert JSX to JavaScript on the browser. **Babel is a library that converts JSX to pure JavaScript and newest JavaScript to previous versions.**

Advantages of JSX:

- Because JSX syntax is so similar to HTML syntax, which many developers are already familiar with, it is simpler to write and read.
- It enables you to specify your whole React component using declarative syntax in a single file by using JSX elements as the component's root.
- By using attributes, which are identical to HTML attributes, it makes it simple to give data to your React components.
- It enables you to create reusable parts that can be applied across your entire programme, which keeps your code clean and manageable.
- Using JavaScript expressions within JSX code makes it simple to construct dynamic components, allowing you to define components that can alter based on data or other variables.

Notes: JSX must have one parent element. Why? JSX is converted into plain JavaScript objects. In JavaScript, you can only return one object from a function. The same applies to JSX - if you want to return multiple JSX tags, you need to wrap them in a parent tag.

For example:

```javascript
const Jsx = <h1>Hello </h1><h1>World</h1>  //error
```

- So the above statement gives an error. To overcome this, we use **React.Fragment** which comes from react library.

```javascript
const Jsx =( <React.Fragment>
                <h1>Hello</h1>
                <h1>World</h1>
            </React.Fragment>
);
```

Also we can use a div tag to wrap both the elements.
Instead of using **React.fragment**, we use an empty tag(<> </>).

```javascript
const Jsx =( <>
              <h1>Hello</h1>
```

```
              <h1>World</h1>
            </>
);
```