WEB APP DEVELOPMENT WITH REACTJS (INT252)

Lecture 6: JSX & Rendering Basics

Unit I – JavaScript Refresher & React Foundations

Syllabus Mapping:

- JSX syntax
- Rendering elements in React
- Expressions and rules of JSX

Target Learner: Beginner (HTML, CSS, JavaScript knowledge)

Why JSX Exists

JSX looks like HTML ❌
JSX is **JavaScript syntax extension** ✅

It allows:

- Writing UI inside JavaScript
- Combining logic + markup

JSX vs HTML (Important)

JSX:

- Compiled to JavaScript
- Follows JS rules
- Case-sensitive

HTML:

- Browser parsed
- Lenient syntax

JSX Example

```
const element = <h1>Hello React</h1>;
```

This is NOT HTML

JSX Gets Converted To

```
React.createElement('h1', null, 'Hello React');
```

JSX is syntactic sugar

Why React Uses JSX

- Readable UI code
- Logic near UI
- Easier maintenance

JSX Rules (Must Remember)

1. Return **single parent element**
2. Use `className` not `class`
3. Close all tags
4. Use camelCase attributes

Single Parent Rule

❌ Wrong:

```
return <h1>Hello</h1><p>World</p>;
```

Correct: Wrapper Element

```
return (
  <div>
    <h1>Hello</h1>
    <p>World</p>
  </div>
);
```

React Fragment

```
return (
  <>
    <h1>Hello</h1>
    <p>World</p>
  </>
);
```

No extra DOM node

JSX Expressions

Use `{}` to embed JS

```
const name = 'Aman';

<h1>Hello {name}</h1>
```

What Can Go Inside `{}`

✅ Expressions
❌ Statements

Allowed:

- Variables
- Function calls
- Math operations

Invalid JSX Example

```
{if (x > 5) { ... }} // ❌
```

JSX does not allow statements

Conditional Rendering Preview

```
{x > 5 && <p>Greater</p>}
```

More later

JSX Attributes

```
<img src="logo.png" alt="logo" />
```

All tags must be closed

className Instead of class

```
<div className="container"></div>
```

`class` is reserved in JS

Inline Styling in JSX

```
<h1 style={{ color: 'red', fontSize: '20px' }}>
  Hello
</h1>
```

Style is an object

Rendering a Component

```
function App() {
  return <h1>Hello JSX</h1>;
}
```

Component must start with capital letter

Rendering Component Inside Component

```
function Header() {
  return <h2>Header</h2>;
}

function App() {
  return <Header />;
}
```

JSX Rendering Flow

```
JSX
 ↓
JavaScript
 ↓
Virtual DOM
 ↓
Real DOM
```

Common Beginner Mistakes

❌ Forgetting to close tags

❌ Using class instead of className

❌ Returning multiple elements

Practice Exercises

1. Create JSX heading with your name
2. Use expression inside JSX
3. Use fragment instead of div

✅ Answers – Practice Exercises

1. Heading

```html
<h1>My Name</h1>
```

✅ Answers – Continued

2. Expression

```
const age = 20;
<h2>Age: {age}</h2>
```

✅ Answers – Continued

3. Fragment

```
<>
  <h1>Hello</h1>
  <p>JSX</p>
</>
```

Key Takeaways

- JSX is JavaScript
- Must follow JSX rules
- Expressions go inside `{}`
- Components return JSX

Next Lecture

Lecture 7: Functional Components & Props

Unit II – Component Design & Styling