

Assignment 3

Author – Mohit R

1. What is JSX?

- It is a JavaScript syntax which is easier to create react elements
- JSX is not part of react, both are very different
- We can write react without JSX also
- JSX- HTML-like or XML like syntax
- JSX is an extension of JavaScript (more or less) and it has to be trans piled by babel before the browser can use it

JSX - React.createElement - ReactElement - Object - HTMLElement(render)

creating a JSX:

```
const jsxHeading = <h1 id="heading">Hello World from JSX!</h1>
```

2. Superpowers of JSX

- **JSX as variables:**
markup (html-like) syntax can be set in a variable. This creates a react element (object).
- **JavaScript expressions in JSX:**

JSX supports all js expressions by wrapping them in {}

- **Attributes in JSX:**
 - We can pass all the html attributes inside jsx tag (attributes must be CamelCased)
- **Props in JSX:** The values of each attribute can be passed as properties (props) to a react element. It can handle dynamic data to create react elements.

3. Role of type attribute in script tag? What options can I use there?

- type attribute of the <script> tag indicates the type of script. Until HTML 4, type is a required attribute. The value of type can be any of the following:
`<script type="" src="app.js"></script>`
- In HTML5, type attribute is not mandatory. If type attribute is not present(default), or an empty string (type="") or javascript MIME type (text/javascript or application/ecmascript), it is treated as classic "javascript" file.
`<script type="" src="app.js"></script>`
- If the type attribute is set module, then the code in that js file is treated as module.
`<script type="module" src="app.js"></script>`
- If the type attribute is set importmap, the body of the element contains importmap ie an JSON object
`<script type="importmap" src="app.js"></script>`

- If the type attribute contains anyother value, then the code is treated as data block and will not be processed by the browser.

`<script type="{ $anyothervalue }" src="app.js"></script>`

4. { TitleComponent } vs {<TitleComponent />} vs {<TitleComponent> </TitleComponent>} in JSX

{ TitleComponent } -

This value in jsx is considered as jsx expression or variable. If no such variable is present, no output will be shown in the browser. Console throws the following warning

index.js:1 Warning: Functions are not valid as a React child. This may happen if you return a Component instead of <Component /> from render. Or maybe you meant to call this function rather than return it.

{<TitleComponent />} -

This value in jsx is meant for rendering a component (i.e) function that returns jsx. This is self-closing tag.

{<TitleComponent> </TitleComponent>} -

This is same as {<TitleComponent />} if there are no child inside TitleComponent. If there are children, then those values come inside {<TitleComponent>} and </TitleComponent>.