What is JSX ?

Ans:-

Its Stand for Javascript XML. JSX allow us to write HTML in React. It makes it easier to write and add HTML in React. It is a syntax extension for JavaScript that allows HTML-like elements to be mixed with JavaScript code. **JSX is not the HTML**

Basically JSX is just the HTML light Syntax not HTML which we write into react. Because of following reasons

* React only understand or use this syntax to create the HTML DOM, React.createElement(‘h1’,{class:’heading’},”heading”). This is called **React Element.**
* But writing element like above can be problematic while dealing with large application.
* So we need something that write inform of html but later convert into react.createElement
* Then JSX help us to write normal html light syntax into react like
* *// writing html using jsx*
* const myName = () => <h1>Kunal Gautam</h1>
* and to convert this syntax we need a convertor that under this JSX and later convert it into react.createElement form. Which is knows as **Babel.**
* Which convert above code into this
* **Const myName = React.createElement(“h1”,{},”kunal Gautam”) , so that browser could understand this code , which is compile/transpile by babel.**
* Hence the normal flow we get as
* JSX **=>** React.createElement ( which gives us object of element) **=>** element object **=>** html dom(**babel** will also render/convert the element object into the html )

What are the super power or Advantage of JSX ?

Ans:-

* It uses the React.createElement behind the scenes. It just another way of creating React Elements.
* It keep our code simpler and elegant when wrting the code in large scale.
* It also allows react to show more useful error and warning messages.
* It faster then normal JavaScript as it’s performs optimizations while translating to regular JavaScript.
* It is secure protect the code from unwanted code injected in our code ,which is done by cross side scripting attack.
* Easy to maintain
* No repetition
* Enhance developer experience as visual aid .

What role of type attribute in script tag ?

Ans:-

* The type attribute specifies the type of the script.
* It identifies the content between the script tags

Q Difference between {TitleComponent} vs {<TitleComponent/>} vs {<TitleComponent></TitleComponent>}

Ans:-

{TitleComponent} :-

* It is just the JavaScript expression which is wrapped in a single curly brace ,which indicate where the variable will get evaluated in JSX.

<TitleComponent/> :-

* Self-closing tag, if we don’t have children element to specify then we can a self-closing tag also.

<TitleComponent> </TitleComponent>:-

* If we have children that come under our component then we have to specify children between the opening and the closing tag.

Q what is the Role of parcel in JSX?

Ans:-

* Parcel support JSX automatically when it detects you are using React. Means it enables the modern JSX transform, so that we don’t need to import React for JSX to work.

Q what is the Role of Babel in JSX?

Ans:-

* Babel is commonly used for converting JSX syntax in React Element. It makes easier for developers to write and understand React Componets.
* Why this compilation is required because browser only understand ECMAScript/React createElement , it does not understand the JSX hence due to this it require the compliler.

Q Difference between HTML and JSX ?

Ans:- JSX is not HTML , it’s just the light html syntax.

|  |  |
| --- | --- |
| HTML | JSX |
| * Multiple Elements can return e.g <ul><li></li><li></li><li></li></ul> <h1></h1> | * Nested JSX must return one element, which we call a parent element that wraps all other levels of nested elements. * Without wrapper element, JSX won’t transpile. * In React, JSX transpile into React createElement and this react element further render into hTMl dom using **React rendering api ( aka ReactDOM).** |
| * HTMl elements have attributes. * E.g <input maxlength = “16”/> | * JSX elemets have props. * E.g <input maxLength=’16’/> |
| * Not necessary to use camelCase for attributes, for naming ids,class, event etc. | * All id , event name become camelCase, * E.g onChange,className |
| * In here we give Id for the uniqueness for each element . e.g array element | * Here we gave Key attribute for uniqueness to each element , e.g array element or children of parent element. |

Q what are Component ?

Ans:-

* react component are independent and resuable bits of code. The whole Ui we build in react are based on component.
* They serve the same purpose as JavaScript functions but work in isolation and return HTML.

Q what are functional component?

Ans:-

* These are just javascript function which return JSX ( Html ).
* They are easy to read , build and maintain as compare to class based component.
* They don’t have state due to which they need hooks to use state.

Q what is the composite Component ?

Ans:-

* These are those component which are used in another component like comp b inside comp A and so on . hence this called composing components.