Namaste React - Class 3 (Laying the Foundation)

Q) If we have sibling elements, we have to provide keys to each sibling element. Why?

---> render means updating something to the DOM tree.
---> Whenever we re-render on adding a new sibling, it has to modify everything inside that DOM.
Instead, when keys are given, it just injects the sibling wherever we want to. Keys are given as props.

Q) Why was React introduced?

---> When Facebook introduced React, the major concept behind introducing the React was that we want to unite a lot of HTML using JavaScript because JavaScript was very performant.

Q) What is JSX?

---> JSX is a HTML like syntax written inside the JavaScript.

Q) Why do we use JSX?

---> React was primarily built to create HTML tags using JavaScript. Because JavaScript is very performant.
---> Using JavaScript we use document.createElement() method.
Similarly, in React they came up with React.createElement() method.
This method made it more complicated while creating more tags.
So JSX syntax was introduced.
---> const heading = </h1> Hello World </h1>
This is JSX.
---> const heading = (
</h1> Hello World </h1>
</h3> Hello World </h3>
)
But when we add multiple tags to a variable, we enclose them using ( ) brackets.

Q) Does the browser understand JSX?

---> No
---> A browser can understand the HTML codes. So try writing this piece of JSX code on the dev console.
const heading = </h1> Hello World </h1>
It throws an error. Because browser does not understand JSX syntax codes.
---> So React uses a JavaScript compiler called Babel that transforms the JSX syntax to normal HTML like syntax.
---> Any piece of React code or React element behind the scenes is an object, or gets converted into an object.
---> JSX internally uses React.createElement() which generates an object and this object gets converted

JSX internally uses React.createElement() which generates an object and this object gets converted into HTML code and this HTML code gets rendered on the DOM.

Q) What is Babel?

---> Babel is a tool that converts the ES6 codes into a backwards compatible version of JavaScript so that older browsers can understand it.
---> Babel gets installed automatically and comes as a dependency when we install Parcel.

diff b/w babel and parcel?
--> Parcel is a bundler used to minify multiple files in a JavaScript project and comes up with a single output file and increase the overall efficiency.
--> Babel is a compiler that converts the ES6 version of js codes into older versions so that all the browsers can understand the code.

Q) What are some advantages of using JSX syntax?

---> Readability
---> Less code
---> Easy to maintain
---> Good developer experience
---> No repeated codes
---> Syntactical sugar

Q) What are React functional components?

---> Functional components are simple Javascript functions that returns some React elements.
---> Functional component's name starts with a capital letter. It's not mandatory but it is a good practice to use this convention.

```
ex: const Greet = () => {
      return (
        <h1></h1>
        <p></p>
      )
    }
```

The above code can be written like this:

```
ex: const Greet = () =>
      (
        <h1></h1>
        <p></p>
```

```
        )
```

Q) Difference between react element and functional component?

React element:

```
const greet = (
          <h1></h1>
          <p></p>
          )
```

Functional component:

```
const Greet = () =>
        (
          <h1></h1>
          <p></p>
        )
```

(or)

```
const Greet = () => {
        return (
          <h1></h1>
          <p></p>
        )
    }
```

Q) What are the different ways to render a functional component?

---> < Greet />

---> { Greet() }

Q) What is component composition?

---> When we use a component inside another component it is known as component composition.