**📘 Chapter 03 – Laying the Foundation**

**✅ 1. What is JSX?**

* **JSX (JavaScript XML)** is a **syntax extension of JavaScript** that looks like HTML but runs inside JS.
* It makes it easier to **create React elements** instead of using React.createElement().

📌 Example:

A screen shot of a computer program

AI-generated content may be incorrect.

**✅ 2. Superpowers of JSX**

1. Looks like HTML → but is actually **JavaScript under the hood**.
2. Easier to write and understand than React.createElement().
3. Supports **JavaScript expressions inside { }**.
4. Helps avoid injection attacks (React sanitizes inputs).
5. Can create complex UIs in a more **readable way**.

**✅ 3. React.createElement vs JSX**

* React.createElement() → Creates an element manually (verbose, hard to read).
* JSX → Cleaner, easier syntax that gets **compiled into React.createElement()** by Babel.

**✅ 4. Behind the Scenes of JSX**

* Browser doesn’t understand JSX.
* **Babel** converts JSX → React.createElement() calls.
* Then React creates a **JS object** (virtual DOM node).
* Finally, ReactDOM renders it to the browser DOM.

**✅ 5. Babel & Parcel role in JSX**

* **Babel** → Transpiles JSX into React code (React.createElement).

(**Transpiles** : process of converting Source code written in **one programming language** into **equivalent source code in another programming language**)

* **Parcel** → Bundles everything together, handles Babel, minification, HMR, etc.

**✅ 6. Components**

* Components = Reusable UI building blocks.
* Two types:
  + **Class components** (older, not much used now).
  + **Functional components** (modern standard).

**✅ 7. Functional Components**

* A **JavaScript function** that returns JSX.

📌 Example:

A screen shot of a computer code

AI-generated content may be incorrect.

**✅ 8. Composing Components**

* Using **one component inside another**.

📌 Example:

A screen shot of a computer code

AI-generated content may be incorrect.

**✅ 9. Role of type attribute in <script> tag**

* Defines the type of script.
* Common options:
  + text/javascript (default, can omit)
  + module → for ES6 modules
  + application/json → if embedding JSON
  + application/ld+json → for structured data

📌 Example:

A black background with colorful text

AI-generated content may be incorrect.

**✅ 10. {TitleComponent} vs {<TitleComponent/>} vs {<TitleComponent></TitleComponent>} in JSX**

* {TitleComponent} → Refers to the **function itself** (not rendered).
* {<TitleComponent />} → Executes the function → renders the component.
* {<TitleComponent></TitleComponent>} → Same as above (just longer syntax).

**💻 Coding Assignment Breakdown**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A computer screen shot of a computer code

AI-generated content may be incorrect.**

A screenshot of a computer program

AI-generated content may be incorrect.