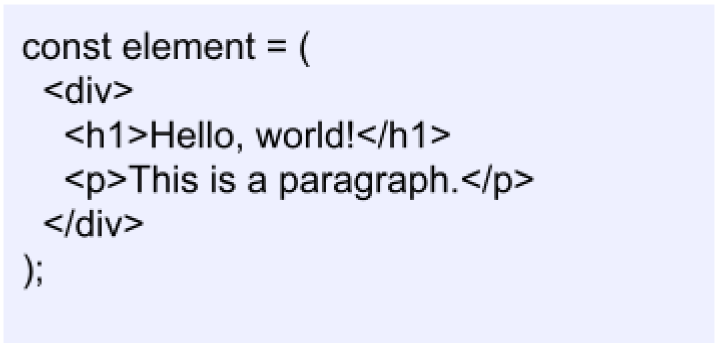
1. **What is JSX?**

   JSX stands for JavaScript XML.

   It's a JavaScript syntax that allows you to write HTML-like structures directly within your JavaScript code.

   JSX is a fast and efficient way to create react elements.

   Note : JSX is not a part of React



```
const element = (
  <div>
    <h1>Hello, world!</h1>
    <p>This is a paragraph.</p>
  </div>
);
```

2. **Superpowers of JSX**

   Enhanced Readability:
   Because of HTML-like Syntax, Visual Clarity

   Improved Developer Experience:
   Because of Faster Development, Reduced Boilerplate Code,Enhanced Error Handling.

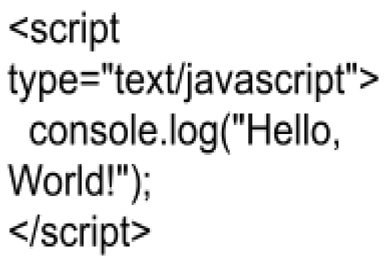
   JSX promotes the creation of reusable components, reducing code duplication and improving maintainability.

   JSX empowers developers to create clean, efficient, and maintainable user interfaces. By combining the structure of HTML with the power of JavaScript, JSX has become an indispensable tool for modern web development.

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

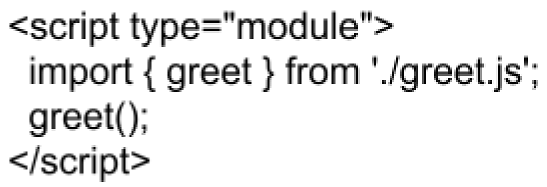
   The `type` attribute in the `<script>` tag specifies the scripting language of the code contained within the script element. While modern browsers assume the default scripting language is JavaScript (`text/javascript`).

   Historically, the `type` attribute was used to declare the scripting language (e.g., JavaScript or VBScript). For example:



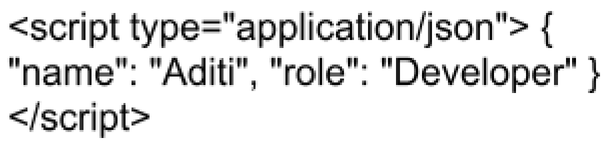
```
<script
type="text/javascript">
  console.log("Hello,
World!");
</script>
```

It allows you to specify ES modules using `type="module"`, which supports importing/exporting code and ensures the script is executed in strict mode. It Enable Modern JavaScript Features.



```
<script type="module">
  import { greet } from './greet.js';
  greet();
</script>
```

The `type` attribute can be used for non-JavaScript code, such as template engines, WebAssembly, or JSON.



```
<script type="application/json"> {
"name": "Aditi", "role": "Developer" }
</script>
```

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

   {TitleComponent}: This value describes the TitleComponent as a javascript expression or a variable. The {} can embed a javascript expression or a variable inside it.

   <TitleComponent/> : This value represents a Component that is basically returning Some JSX value. In simple terms TitleComponent is a function that is returning a JSX value. A component is written inside the {< />} expression.

   <TitleComponent></TitleComponent> : <TitleComponent /> and <TitleComponent></TitleComponent> are equivalent only when < TitleComponent /> has no child components. The opening and closing tags are created to include the child components.