JSX - JavaScript XML

JSX

To make react work, the two libraries is needed which is

React - Responsible for creating components / views
React-DOM - Responsible for rendering.

2

We will see with below example.

JS

```
class Message extends React.Component{
  render()
  {
  return(
    React.createElement('div',null,
                        React.createElement('h1',null,"React"),
                        React.createElement('h1',null,"React DOM"),
                        ) )
   }
}
ReactDOM.render(React.createElement(Message),
                document.getElementById('reactcontainer'));
```

JS CODE

HTML

```
<div>
  <h4>Hello</h4>
</div>
<div id = 'reactcontainer'>

</div>
```

HTML

- I imported two library react and react dom in codepen via CDN.
- In reactcontainer, we inserted two h1 tag grouped under div like this.

```
<div>
 <h1>React</h1>
 <h1>React Dom</h1>
</div>
```

- The **react.createElement** method construct the above structure.
- The **reactdom** is taking responsibility of rendering under destination.

- The react.createElement will look simple for this example. But imagine it for large projects. Thing become difficult there. 😱
- Think like instead of coding like this tree structure, what if we do like our html code (Like Below). The thing become easier now. 🤔
- It is what is called **JSX**

```
return
(
 <div>
 <h1>React</h1>
 <h1>React Dom</h1>
</div>.
)
```

- It looks like HTML, but not. It is a JSX.
- JSX will not understandable by browser.
- In end it will be complied to normal react JS code what we write before.

Who is behind it?

BABEL

BABEL.

- It is JavaScript compiler that transforms advanced JS code into browser understandable format.
- Babel is not only for JSX. Used for many other things like ECMA2016 JS code into older format.
- It makes the life easier.

Here you can see what babel can do. It translate our JSX into JS code.

```
class Message extends React.Component {
  render()
  {
  return(
   <div>
      <h1>React</h1>
      <h1>React Dom </h1>
  </div>

)
  }
}
```

```
"use strict";

class Message extends React.Component {
  render() {
    return /*#__PURE__*/React.createElement("div", null,
/*#__PURE__*/React.createElement("h1", null, "React"),
/*#__PURE__*/React.createElement("h1", null, "React Dom "));
  }
}
```

BABEL SANDBOX

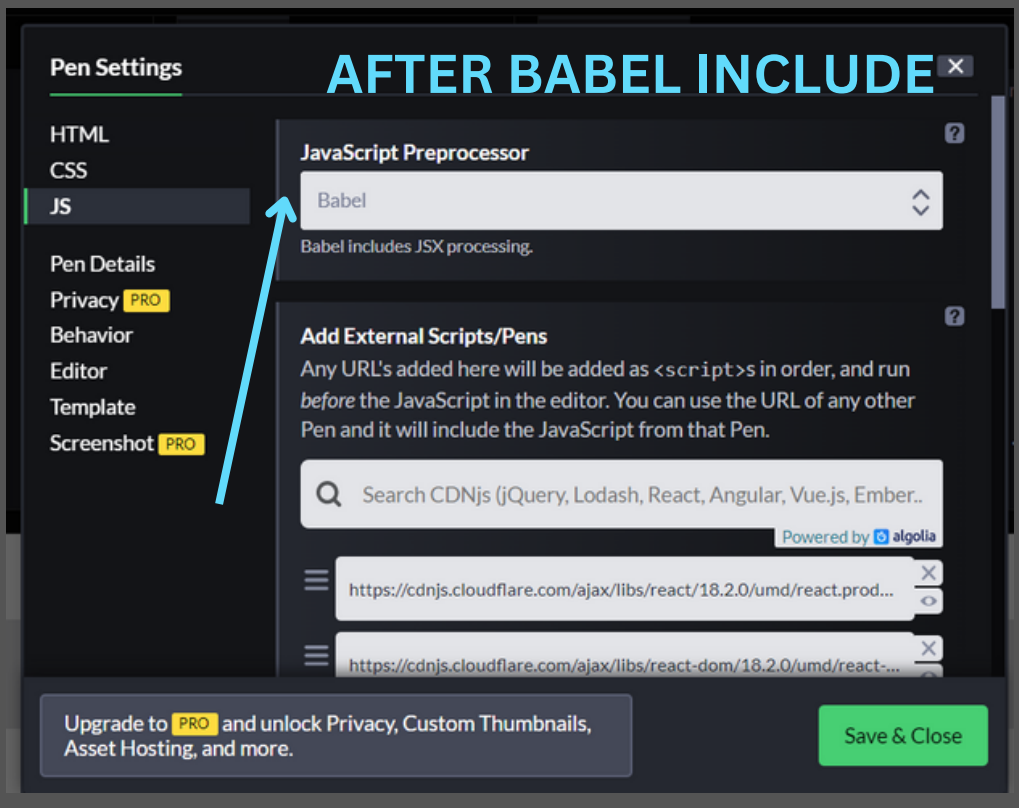
- As earlier said JSX is not supported by browser. It required some compiler translate into browser understandable format.
- Here in code pen without babel you can see the code is giving error.

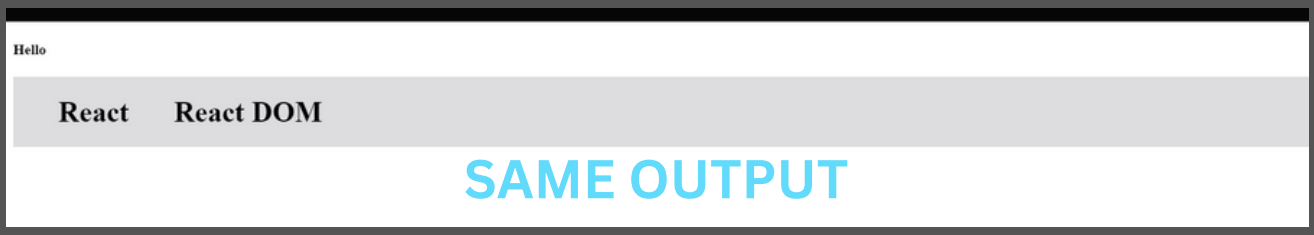
JS

```
class Message extends React.Component{
  render()
  {
  return(
    <div>
Uncaught SyntaxError: Unexpected token '<'
      <h1>React</h1>
      <h1>React Dom </h1>
    </div>
    )
      }
}
ReactDOM.render(<Message/>,document.getElementById('reactcontainer'));
```

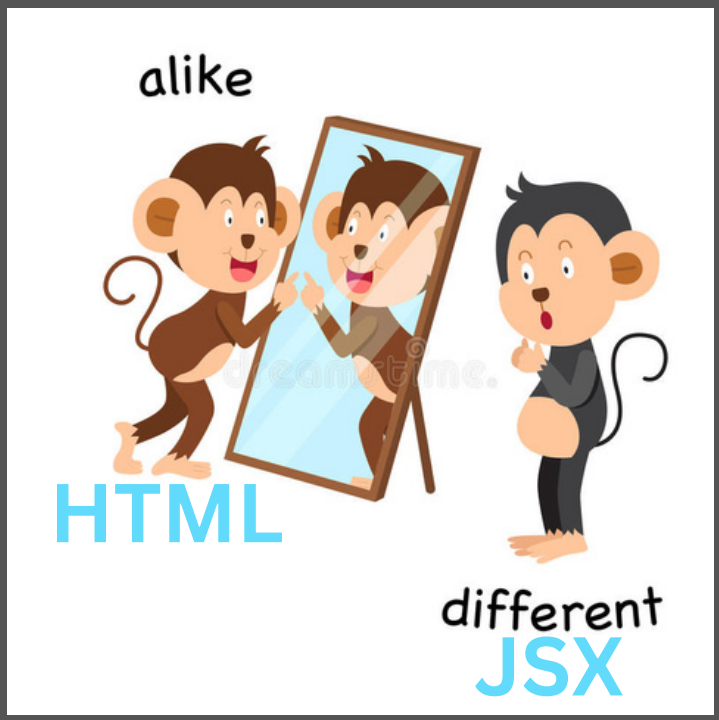
WITHOUT BABEL

AFTER BABEL INCLUDE





- As earlier said JSX is not HTML. It look like HTML but not.
- Simple Example. We will use class in HTML.
- But in JSX if we use this it will throw error. Because class is a reserved keyword in JS.
- We have to use className. This is one example, so some other difference also there in JSX.



- Anyways the ClassName is translated into class at final.
- In simple words, in one thing it is called class. And on another thing it is called ClassName. It is Like Language difference. But meaning is same.