

# ReactJS - JSX

As we learned earlier, React JSX is an extension to JavaScript. It enables developer to create virtual DOM using XML syntax. It compiles down to pure JavaScript (*React.createElement function calls*). Since it compiles to JavaScript, it can be used inside any valid JavaScript code. For example, below codes are perfectly valid.

Assign to a variable.

```
var greeting = <h1>Hello React! </h1>
```

Assign to a variable based on a condition.

```
var canGreet = true;
if(canGreet) {
  greeting = <h1>Hello React! </h1>
}
```

Can be used as return value of a function.

```
function Greeting() {
  return <h1>Hello React! </h1>
}
greeting = Greeting()
```

Can be used as argument of a function.

```
function Greet(message) {
  ReactDOM.render(message, document.getElementById('react-app'))
}
Greet(<h1>Hello React! </h1>)
```

## Expressions

JSX supports expression in pure JavaScript syntax. Expression has to be enclosed inside the curly braces, `{ }`. Expression can contain all variables available in the context, where the JSX is defined. Let us create simple JSX with expression.

## Example

```
<script type="text/babel">
  var cTime = new Date().toString();
  ReactDOM.render(
    <div><p>The current time is {cTime}</p></div>,
    document.getElementById('react-app') );
</script>
```

## Output

Here, `cTime` used in the JSX using expression. The output of the above code is as follows,

```
The Current time is 21:19:56 GMT+0530(India Standard Time)
```

One of the positive side effects of using expression in JSX is that it prevents *Injection attacks* as it converts any string into html safe string.

## Functions

JSX supports user defined JavaScript function. Function usage is similar to expression. Let us create a simple function and use it inside JSX.

## Example

```
<script type="text/babel">
  var cTime = new Date().toString();
  ReactDOM.render(
    <div><p>The current time is {cTime}</p></div>,
    document.getElementById('react-app')
  );
</script>
```

# Output

Here, `getCurrentTime()` is used get the current time and the output is similar as specified below –

```
The Current time is 21:19:56 GMT+0530(India Standard Time)
```

# Attributes

JSX supports HTML like attributes. All HTML tags and its attributes are supported. Attributes has to be specified using camelCase convention (and it follows JavaScript DOM API) instead of normal HTML attribute name. For example, class attribute in HTML has to be defined as `className`. The following are few other examples –

*htmlFor* instead of *for*  
*tabIndex* instead of *tabindex*  
*onClick* instead of *onclick*

# Example

```
<style>
  .red { color: red }
</style>
<script type="text/babel">
  function getCurrentTime() {
    return new Date().toString();
  }
  ReactDOM.render(
    <div>
      <p>The current time is <span className="red">{getCurrentTime()}</span></p>
    </div>,
    document.getElementById( ' react- app' )
  );
</script>
```

# Output

The output is as follows –

# Expression in attributes

JSX supports expression to be specified inside the attributes. In attributes, double quote should not be used along with expression. Either expression or string using double quote has to be used. The above example can be changed to use expression in attributes.

```
<style>
  .red { color: red }
</style>

<script type="text/babel">
  function getCurrentTime() {
    return new Date().toString();
  }
  var class_name = "red";
  ReactDOM.render(
    <div>
      <p>The current time is <span className={class_name}>{getCurrentTime()}</span></p>
    </div>,
    document.getElementById('react-app')
  );
</script>
```

---

Revision #1

Created 14 December 2022 10:38:56 by Admin

Updated 14 December 2022 10:39:29 by Admin