## React

A Javascript (Frontend) Framework

## Node

- What is Node.js?
- Install Node.js (> version 8.\*)

# **ES6**, **ES7**

## **ES6**, **ES7**

- Class
- Arrow Function
- let, const
- map, filter
- More on: <a href="http://es6-features.org">http://es6-features.org</a> & <a href="https://medium.freecodecamp.org/write-less-do-more-with-ja">https://medium.freecodecamp.org/write-less-do-more-with-ja</a> vascript-es6-5fd4a8e50ee2

# Create a simple project

- Create a folder
- Install live-server

```
npm install -g live-server
```

- Add index.html inside public
- Create a simple server

run live-server

#### Import react, react-dom library

```
<script crossorigin
src="https://unpkg.com/react@16/umd/react.dev
elopment.js"></script>

<script crossorigin
src="https://unpkg.com/react-dom@16/umd/react
-dom.development.js"></script></script>
```

- Create app.js and test

#### - Render using ReactDOM

```
1 var template = Rendering JSX
2 var myApp = document.querySelector('#my-app')
3
4 ReactDom.render(template, myApp)
```

#### - Install Babel

npm install -g babel-cli@6.24.1

- Initialize NPM and add React presets

```
npm init
npm install babel-preset-react@6.24.1
npm install babel-preset-env@1.5.2
```

- Create src folder and srcApp.js file
- Compile JSX using Babel presets

```
babel src/srcApp.js
--out-file=public/scripts/app.js
--presets=env,react --watch
```

- To install all the required packages for a project, run: npm install (it will install all the packages listed inside package.json file)

 To get proper syntax highlighting in VS Code, install Babel ES6/ES7 plugin

## **Dive into React**

# **JSX**

## **JSX**

### - Wrap adjacent JSX

```
1 // Error: Adjacent JSX elements must be wrapped in an
  enclosing tag
2 let template = Para.<h1>Heading</h1>
3
4 // Fix
5 let template = <div>Para.<h1>Heading</h1></div>
```

## **JSX**

- For better readability: indent

## JSX Exercise

```
Exercise 1: Create a template called personTemplate and add JSX for the followings:
```

```
_____
```

```
h1 -> Your name (i.e. Jane Doe)
p -> Youw batch (i.e. Batch: 35th)
p -> Your hometown (i.e. Hometown: Dhaka)
```

```
1 const person = {
    name: "Jane Doe",
    batch: "35th",
    hometown: "Dhaka",
 6
 7 let personTemplate = (
8 <div>
    <h1>{person.name}</h1>
    Batch: {person.batch}
10
    Hometown: {person.hometown}
12 </div>
```

```
1 // Conditional Rendering: Using if-else inside
  function:
2 |
3 Hometown: {getHometown(person.hometown)}
```

```
1 // Conditional Rendering: Using if-else inside
function:
2
3 Hometown: {getHometown(person.hometown)}
4
5 // Or,
6
7 {getHometown(person.hometown)}
```

```
// Conditional Rendering: Using Ternary Operator

// Conditional Rendering: Using Ternary Operator

// Sep>Hometown: {person.hometown : "Unknown"}

// Usecase: We want to keep the tag (here: ...)

// We have a value for both true and false
```

```
1 // Conditional Rendering: Using the && (AND) operator
2 
3 {person.hometown && Hometown: {person.hometown}}
4 
5 // Usecase: We want to render the element only if it meets our condition
```

#### JSX: Class, ID and Event Function

```
const counterTemplate = (

count count className="count">
count cou
```

- CSS ID is *id*, CSS class is *className* because of the conflict with JS *class* keyword and events are camel cased.
- To know more about what is supported and what is not, please visit: <a href="https://reactjs.org/docs/dom-elements.html">https://reactjs.org/docs/dom-elements.html</a>

#### JSX: Exercise

```
const counterTemplate = (

count count className="count">
count cou
```

Add another button with a label: Substract (-1)

Attach a method called substractOne with the click event

Print "substracted one" when the button is clicked

### JSX: Dom Binding/Rendering

```
1 const addOne = () => {
        count++;
        renderCounterApp();
 4
  5
   renderCounterApp = () => {
        const counterTemplate = (
          <div>
 10
          </div>
11
       ReactDOM.render(counterTemplate, myApp)
13 }
14
      Do Not Forget To Call It Initially
16
```

#### JSX: Exercise

Do the same for substractOne and resetCounter methods.

#### JSX: Form and Default Event

All types of react events: <a href="https://reactjs.org/docs/events.html#supported-events">https://reactjs.org/docs/events.html#supported-events</a>

### JSX: Array

- JSX Array's each element is treated as a separate element

- Falsy value is just being ignored by the renderer.

### JSX: Array (Quiz)

- Is this a valid JSX syntax?
- What will be it's output/error?
- (Check console *key* for error)

### JSX: Rendering Array using Map

```
1 const names = ["Jane", "John", "Doe"]
  const template = (
      <div>
             names.map(name => {
                return Name: {name}
             })
10
      </div>
```



- Render each item of the *tasks* array inside *li*s using a *map*