Notes on React

Creating and rendering elements

src > index.js > setInterval() callback

```
import React from 'react';
import ReactDOM from 'react-dom';

function time() {
    /**
     * React.createElement(
        type,
        [props],
        [...children]
        )
     */
    return React.createElement(
        'h1',
        null,
        <span>{new Date().toLocaleTimeString()}</span>,
        <h1> Children 2 </h1>
    );
}

const rootDiv = document.getElementById('root');

setInterval(() => {
    /**
        ReactDOM.render(
            element,
            container[,
            callback]
        )
     */
    ReactDOM.render(time(), rootDiv);
}, 1000);
```

type : argument can be either a tag name eg. div,span, A React.Component or A React.Fragment

[props] : a config for the the ReactElement such as style

[...children] : multiple children can be passed as it's rest operator

React.craeteElement : Returns a ReactElement Which is the smalles building block in react

ReactDOM/Native.render renders

React element/s in the supplied

container and execute

Callback if provided

with JSX

src > JS index.js > …

```
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3
4  function time() {
5      return (
6          <div style={{ color: 'red' }}>
7              <span>{new Date().toLocaleTimeString()}</span>
8              <h1> Children 2 </h1>
9          </div>
10     );
11 }
12
13 const rootDiv = document.getElementById('root');
14
15 setInterval(() => {
16     ReactDOM.render(time(), rootDiv);
17 }, 1000);
18
```

JSX syntax

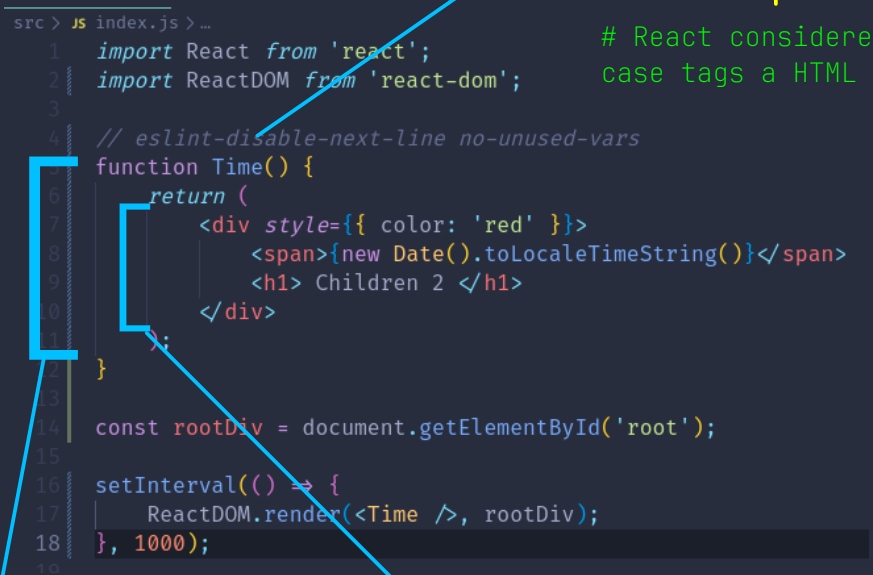
JS expression inside curly braces

ReactDOM escapes any values embedded in JSX before rendering them.This helps prevent XSS(cross-site-scripting) Attacks

Functional Component

User-Defined Components Must Be Capitalized

React considered lower-case tags a HTML tags



The Time function considered as a component

While the returned element by a component considered as a ReactElement

Props

src > JS index.js > …

```
import React from 'react';
import ReactDOM from 'react-dom';

function Time(props) {
    const { color, locale } = props;

    return (
        <div style={{ color }}>
            <span>{new Date().toLocaleTimeString(locale)}</span>
        </div>
    );
}

const rootDiv = document.getElementById('root');

setInterval(() => {
    ReactDOM.render(
        <div>
            <Time color="blue" locale="bn-BD" />
            <Time color="green" locale="ar-EG" />
            <Time color="red" locale="ko-KR" />
        </div>,
        rootDiv
    );
}, 1000);
```

Read props inside the child component

Don't miss the pair of { and } curlies inside of (and) when declaring props:

or destructring the objects received

Passing props to the child components as comma separated values

props are immutable—a term from computer science meaning "unchangeable". When a component needs to change its props (for example, in response to a user interaction or new data), it will have to "ask" its parent component to pass it different props—a new object!

Class Component