JSX

.Lect 3

1

Objectives

- Review the DOM and the virtual DOM
- Introduce JSX
- Understand basic syntax for JSX
- Introduce the transpiler

2

What is React

- JSX is a JavaScript extension.
- JSX makes creating React elements much easier, faster and more compact.
- It also makes the reading of React elements much easier.
- JSX code is converted(transpiled) into JavaScript.
- React components can be written without JSX, but not recommended.

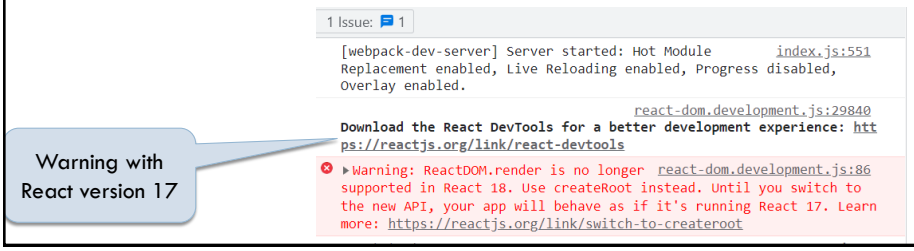
3

Rendering Elements

- ReactDom takes care of rendering of React components in web browser.

my-app > src > JS index.js

```
1  import React from 'react'
2  import ReactDOM  from 'react-dom'
3  import App from './App'
4
5  // version 17
6  ReactDOM.render( <App /> , document.getElementById('root'))
```



4

React version 18

5

- The **ReactDOM** component makes it possible for the user interfaces to handle different screen changes required by modern web applications.
- It does this with the help of the Virtual DOM.

6

- The Document Object Model (**DOM**) is an object-oriented representation of an HTML or XML document.
- The structure of an HTML / XML document is hierarchical, so the DOM structure resembles that of a tree

7

- The DOM models a document as a set of nodes.
- The DOM is not a programming language
- Several language have bindings to the DOM
  - JavaScript, PHP, JQuery
- Changes to the DOM cause changes in the web browser

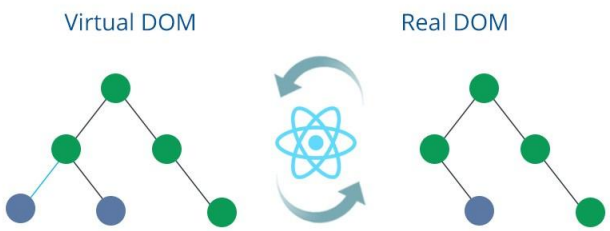
8

Virtual DOM

- DOM manipulation is slow
- DOM changes have to be checked on all levels of the DOM tree to see if the page need refreshed.
- React, takes away the details of how and when the DOM is modified from programmers.
- They created a layer between the code that the programmer writes and the DOM
- This layer is the Virtual DOM.

9

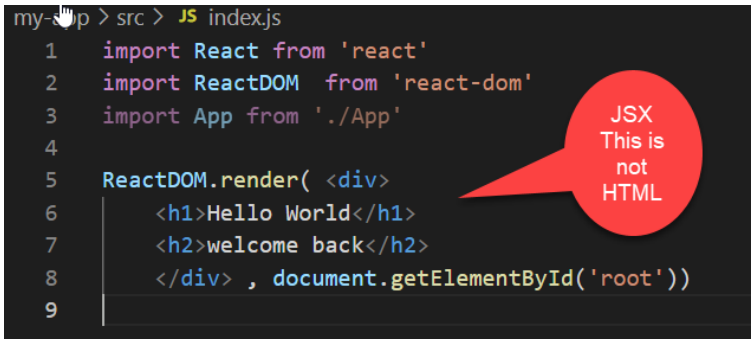
- ReactJS does not update the original DOM directly.
- In ReactJS, for every DOM object, there will be a corresponding in-memory copy created.
- This copy is called the Virtual DOM.



10

JSX

- HTML/XML-like syntax for composing React components.
- Extension of JavaScript that's used as a visual aid



11

JSX Benefits

- Code is easier to read and maintain.
- React assumes that you use JSX and reports helpful error messages as if you are.
- Faster code—the transpiler optimises the code on the fly.
- Casual developers (e.g., designers) can modify code more easily because JSX looks like HTML.
- Less code, less errors.

12

Without JSX

```
React.createElement(
    'h1',
    null,
    'Hello World',
 );
```

With JSX

```
<h1>
Welcome
</h1>
```

13

```
React.createElement(
    Title,
    {size: 8},
    'Welcome to ',
    React.createElement(
    'strong',
    null,
    'year four',
    ),
    );
```

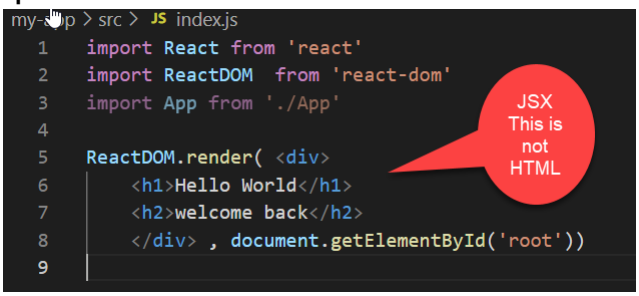
```
<Title size="8">
Welcome to
<strong>year four</strong>
</Title>
```

JSX looks like HTML (easier to read)

14

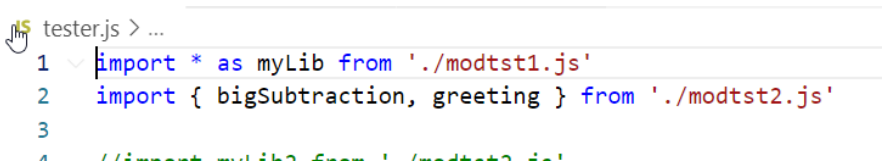
JSX

- XML-based syntax extension to JavaScript.
- Integral part of how React components are developed.
- React uses JSX elements to represent custom components.



15

- **import** and **export** should always be at the top level of your JavaScript file.
- The **import**(s) statements should be at the top of the module..



16

- React has components built into it for HTML5 elements.

```
function Links() {
    return (
      <div>
        <a href="//atu.ie">Donegal ATU</a>
      </div>
    );
  }

export default Links;
```

17

- Creating React components without using JSX (not recommended).

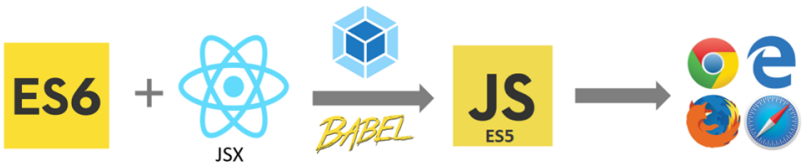
```
"use strict";

/*#__PURE__*/
React.createElement("div", null, /*#__PURE__*/React.createElement("h1", {
  className: "head"
}, "Head section"), /*#__PURE__*/React.createElement("h3", null, "Menu section"));
```

```
<div>
    <h1 className="head">Head section</h1>
    <h3>Menu section</h3>
</div>
```

18

- Reverse strategy on **separation of concerns**, seeks to keep related items together.
- JSX is compiled by various transformers into standard ECMAScript.

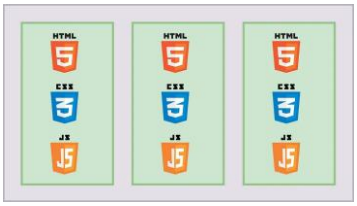


19

Separation of concerns

- Separate logic from presentation.
  - Logic mainly outside the return statement

Traditional SoC



Component SoC

20

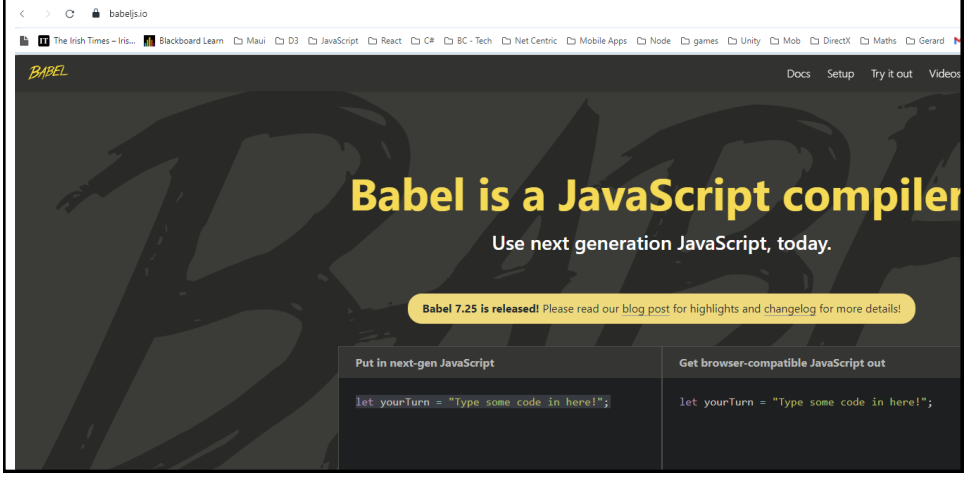
React transpiler

- Compilation of applications is different to other languages (like C++ or Java).
- With Java/C#/C++ the code is converted into low-level code that can be understood by the computer's software interpreter. (Java bytecode).
- **With React applications are compiled to another version of JavaScript.**
- Process called **transpilation**.
- One popular tool used for transpilation in JavaScript is called **Babel**.

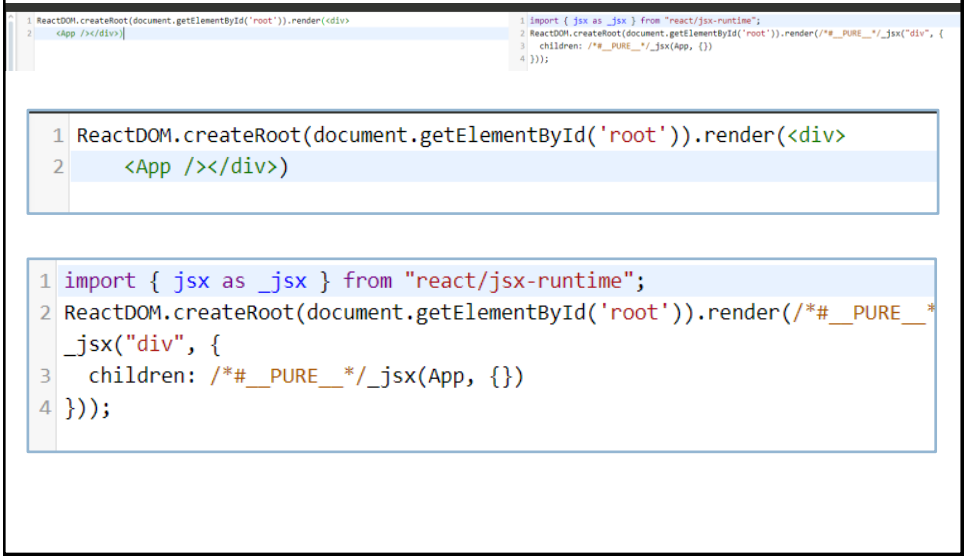
21

Babeljs.io

- Built into React App
- Babel converts JSX into JavaScript that can be



22



```
ReactDOM.createRoot(document.getElementById('root')).render(<div>
    <App /></div>)
```

```
import { jsx as _jsx } from "react/jsx-runtime";
ReactDOM.createRoot(document.getElementById('root')).render(/*#__PURE__*
_jsx("div", {
  children: /*#__PURE__*/_jsx(App, {})
}));
```

23

- When your browser executes a React application, it will only see the `React.createElement` statements required to generate the required structure.
- JSX needs to be **transpiled** into regular JavaScript before browsers can execute the code.

24

JSX syntax

- JSX similar to XML
  - All elements must be closed.
  - Elements that cannot have child nodes (so-called “empty” elements) must be closed with a slash.
  - Attributes that are strings must have quotes around them.
  - HTML elements in JSX must be written in all lowercase letters.
- Since JSX is closer to JavaScript than to HTML, React DOM uses camelCase property naming convention instead of HTML attribute names.
- React component can only return one thing

25

JSX tags may contain children:

my-app > src > JS App.js > App

```
1  function App() {
2      return <div>
3        <h2 className="tst">test image</h2>
4        <img src="tst.png" alt="img" />
5      </div>
6  }
```

Attribute names that contain more than one word are camel-cased

26

Comments

- HTML comments don't work.
- However, you can put comment inside curly braces to avoid transpilation.

```
return (<>
  {/* JSX comment  */}
```

27

- To include a variable or JavaScript in JSX.
- **Use curly braces**
  - Doesn't get interpreted by the transpiler.

```
const App = () => {
    console.log("React test msg");
    let today = new Date();
    let myImg = "tst.png"
    let favouriteNum = Math.floor(Math.random() * 100) + 1
    return (<>
      <h2>test2 arrow! on {today.toString()}</h2>
      <h3>my favourite number =  {favouriteNum}</h3>
      <img src={myImg} />
    </>)
}
```

28

JSX variable example

- Curly brace { } notation to output variables

```
function App() {
  const tomorrow = new Date().toDateString()
  return (
    <div className="App">
      <p>Hello <b>World!</b> {tomorrow}</p>;
      <Links />
      <Links />
    </div>
  );
}
```

29

Double braces for Objects

```
return (<>
  <h2>test2 arrow! on {today.toString()}</h2>
  <h3 style={{border:"solid", width:"200px", color:"red"}}>my favour
  <img src={myImg} />
</>)
```

my favourite number = 16

hello

30

Expressions

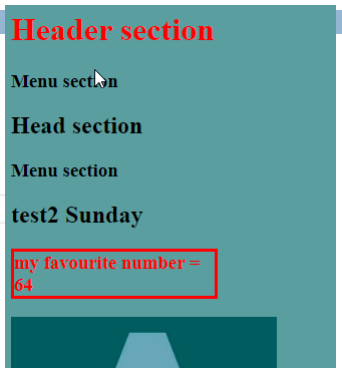
- Use any JavaScript expression inside JSX
- Insert into a React element attribute values by surrounding it with curly braces.
- Expression is any valid unit of code that resolves to a value.
- Make sure you understand the difference between expressions and statements.
- We can have statement but out the return

31

conditional statement

```
let today = new Date().getDay()
let dayOfWeek = ""

if (today === 0) {
  dayOfWeek = "Sunday"
} else {
  dayOfWeek = "Other day"
}
return (<>
 <Heading />
  <h2>test2 {dayOfWeek}</h2>
  <h3 style={{border:"solid", width:"200px", color:"red"}}>my favourite
```



32

- Tenary condition

```
let today = new Date().getDay()
let dayOfWeek = ""

return (<>
 <Heading />
  <h2>test2 { today === 0 ? "Sunday" : "other day" }</h2>
  <h3 style={{border:"solid", width:"200px", color:"red"}}>my
  <img src={myImg} />
</>)
}
```

33

Summary

- Reviewed the role of the virtual DOM
- JSX is an important tool used in the development of nearly every React component.
- JSX is not HTML
- JSX code with angle brackets (<>) is easier to read than code with a lot of `React.createElement()` statements
- Add JavaScript into JSX with curly braces

34

Webography

- Web Links
  - https://www.w3schools.com/nodejs/
  - https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm

35