Rendering Elements

React

Presented by : Pankaj Sharma

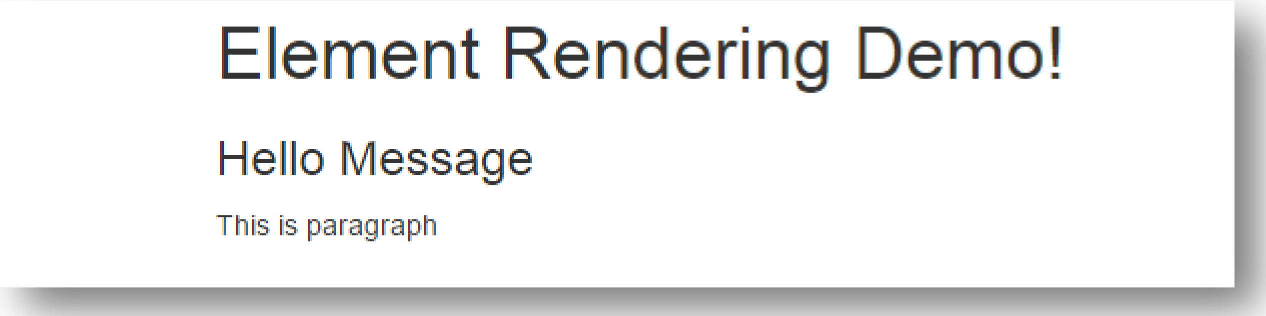
Topics to be covered

- Introduction
- Rendering Element into DOM
- Updating the rendered Element
- React DOM Feature

Introduction

- Elements are the smallest building blocks in React App.
- An element says what you want to see on screen.

Let’s create a simple application as shown below. In the next slide you will get the HTML template.



Element Rendering Demo!

Hello Message

This is paragraph

Introduction

HTML template :

```
Day3/elementrenderingdemo.html
--------------------------------------------------
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
/>

</head>
<body>
<div class="container">
<h1>Element Rendering Demo!</h1>
<div id="root">
<!-- React Component will render here-->
</div>
</div>
<!– Write React Code here -->
</body>
</html>
```

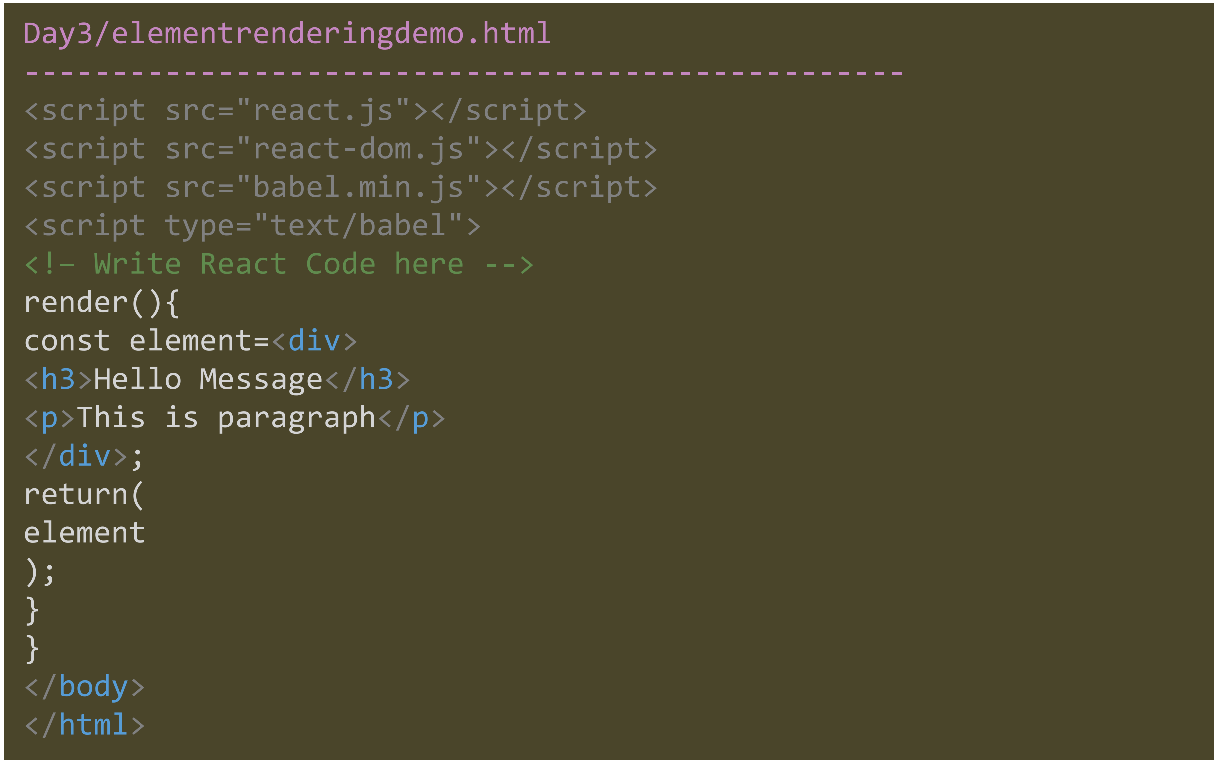
Write the code to generate the output as shared in last slide.

Rendering element into DOM

You would have written the code as below.

```
return(
<div>
<h3>Hello Message</h3>
<p>This is paragraph</p>
</div>
);
```

The red coloured code is the element. We can also write the above code as :



```
Day3/elementrenderingdemo.html
---------------------------------------------------
<script src="react.js"></script>
<script src="react-dom.js"></script>
<script src="babel.min.js"></script>
<script type="text/babel">
<!– Write React Code here -->
render(){
const element=<div>
<h3>Hello Message</h3>
<p>This is paragraph</p>
</div>;
return(
element
);
}
}
</body>
</html>
```

Unlike browser DOM elements, React elements are plain objects, and are easy to create. React DOM takes care of updating the DOM to match the React elements.

Updating the Rendered Element & React DOM Feature

React elements are immutable. Once you create an element, you can’t change its children or attributes.

The only way to update the UI is to create a new element, and pass it to ReactDOM.render().

```
Day3/clockdemo.html
--------------------------------------------------
<!DOCTYPE html>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>clock demo</title>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
/>
</head>
<body>
<div class="container">
<h1>Updating Rendered Element Demo!</h1>
<div id="root">
<!-- React Component will render here-->
</div>
</div>

-----Continue on next slice
```

Updating the Rendered Element & React DOM Feature

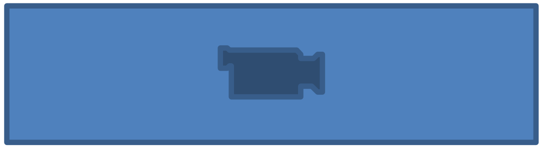
```
Day3/clockdemo.html
---------------------------------------------------
<script src="react.js"></script>
<script src="react-dom.js"></script>
<script src="babel.min.js"></script>
<script type="text/babel">
function clockTick(){
const element=(
<div>
<h3>Tick Tick Clock</h3>
<h2>This is {new Date().toLocaleTimeString()}.</h2>
</div>
);

/* Render React Component */
ReactDOM.render(element,document.getElementById("root"));
}
setInterval(clockTick,1000);
</script>
</body>
</html>
```

Now run your application, it should show output as running clock.

Updating the Rendered Element & React DOM Feature

React DOM compares the element and its children to the previous one, and only applies the DOM updates necessary to bring the DOM to the desired state.



Even though we create an element describing the whole UI tree on every tick, only the text node whose contents has changed gets updated by React DOM.

Thank You!

Email: info@yash.com
Web: www.yash.com

The information in this document is based on certain assumptions and as such is subject to change. No part of this document may be reproduced, stored or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of YASH Technologies Inc. This document makes reference to trademarks that may be owned by others. The use of such trademarks herein is not as assertion of ownership of such trademarks by YASH and is not intended to represent or imply the existence of an association between YASH and the lawful owners of such trademarks.

Presented by : Pankaj Sharma