**Elements & Functions in React JS**

* Rendering Elements in React.JS

Elements are the smallest building blocks of React apps.

An element describes what you want to see on the screen:

const element = <h1>Hello, world</h1>;

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

* Hands-on: How does render function work and update DOM

To render a React element, first pass the DOM element to [ReactDOM.createRoot()](https://reactjs.org/docs/react-dom-client.html" \l "createroot), then pass the React element to root.render():

const root = ReactDOM.createRoot(

document.getElementById('root')

);

const element = <h1>Hello, world</h1>;

root.render(element);

**Updating the Rendered Element**

React elements are [immutable](https://en.wikipedia.org/wiki/Immutable_object). Once you create an element, you can’t change its children or attributes. An element is like a single frame in a movie: it represents the UI at a certain point in time.

With our knowledge so far, the only way to update the UI is to create a new element, and pass it to root.render().

Consider this ticking clock example:

const root = ReactDOM.createRoot(

document.getElementById('root')

);

function tick() {

const element = (

<div>

<h1>Hello, world!</h1>

<h2>It is {new Date().toLocaleTimeString()}.</h2>

</div>

);

root.render(element);}

setInterval(tick, 1000);

It calls [root.render()](https://reactjs.org/docs/react-dom.html" \l "render) every second from a [setInterval()](https://developer.mozilla.org/en-US/docs/Web/API/WindowTimers/setInterval" \t "_blank) callback.

* Hands-on: Styling and CSS

There are many ways to style React with CSS, this tutorial will take a closer look at inline styling, and CSS stylesheet.

**Inline Styling**

To style an element with the inline style attribute, the value must be a JavaScript object:

**Example:**

Insert an object with the styling information:

class MyHeader extends React.Component {

render() {

return (

<div>

<h1 style={{color: "red"}}>Hello Style!</h1>

<p>Add a little style!</p>

</div>

);

}

}

**CSS Stylesheet**

You can write your CSS styling in a separate file, just save the file with the .css file extension, and import it in your application.

**App.css:**

Create a new file called "App.css" and insert some CSS code in it:

body {

background-color: #282c34;

color: white;

padding: 40px;

font-family: Arial;

text-align: center;

}

**index.js:**

Import the stylesheet in your application:

index.js:

import React from 'react';

import ReactDOM from 'react-dom/client';

import './App.css';

class MyHeader extends React.Component {

render() {

return (

<div>

<h1>Hello Style!</h1>

<p>Add a little style!.</p>

</div>

);

}

}

ReactDOM.render(<MyHeader />, document.getElementById('root'));