**React Lifecycle**

**Learning Objectives**

* Understand that React has lifecycle methods
* Look at a few of the methods in action

React automatically looks for functions throughout the life of a component.

We can think of a component as having a birth, life and death, just like us!

You won’t be using all of these but they are handy to know, especially for getting AJAX data.

We are not required to implement these ‘lifecycle’ methods but if we choose to, React will run them automatically for us in the order of the lifecycle.

Think of them like hooks that we can use to run some code if need be.

Open up piggy bank / counter app from first day. Add the following methods in to the main component and see how the lifecycle methods work within it.

**Looking at the lifecycle**

Firstly add a console.log to the render method. render is called every time the component re-renders.

Next let’s look at where componentDidMount and componentWillMount are called, and the difference between them:

*// ./src/PiggyBank.jsx*

componentWillMount() {

console.log('Component WILL MOUNT!');

**var** button **=** document.querySelector('button');

console.log("Button:", button);

}

componentDidMount() {

console.log('Component DID MOUNT!');

**var** button **=** document.querySelector('button');

console.log("Button:", button);

}

As we can see, before the component has mounted we don’t have access to any of the DOM elements created by the component.

Two more of the lifecycle methods we can look at are:

*// ./src/components/PiggyBank.js*

componentWillUpdate(nextProps, nextState) {

console.log('Component WILL UPDATE!');

console.log("Next State:", nextState);

console.log("Next Props:", nextProps);

}

componentDidUpdate(prevProps, prevState) {

console.log('Component DID UPDATE!')

console.log("Previous State:", prevState)

console.log("Previous Props:", prevProps)

}

These allow us to look at the props and state before and after the component has rendered.

They don’t get called on the initial render. There is a similar method called componentWillReceiveProps() that gets called when a child component is passed new props from its parent.

Similarly we can access the nextProps and do any changes to state etc that are needed.

**Resources**

* Dan Abramov’s lifecycle diagram.
* More lifecycle overview:

<http://buildwithreact.com/article/component-lifecycle> <http://busypeoples.github.io/post/react-component-lifecycle/>

* Details of all methods (official docs):

<https://facebook.github.io/react/docs/component-specs.html>

P