React class components can have hooks for several lifecycle events.

*Hooks allow function components to access them too, in a different way.*

During the lifetime of a component, there’s a series of events that gets called, and to each event you can hook and provide custom functionality.

What hook is best for what functionality is something we’re going to see here.

First, there are 3 phases in a React component lifecycle:

* Mounting
* Updating
* Unmounting

Let’s see those 3 phases in detail and the methods that get called for each.

**Mounting**

When mounting you have 4 lifecycle methods before the component is mounted in the DOM: the constructor, getDerivedStateFromProps, render and componentDidMount.

**Constructor**

The constructor is the first method that is called when mounting a component.

You usually use the constructor to set up the initial state using this.state = ....

**getDerivedStateFromProps()**

When the state depends on props, getDerivedStateFromProps can be used to update the state based on the props value.

It was added in React 16.3, aiming to replace the componentWillReceiveProps deprecated method.

In this method you haven’t access to this as it’s a static method.

It’s a pure method, so it should not cause side effects and should return the same output when called multiple times with the same input.

Returns an object with the updated elements of the state (or null if the state does not change)

**render()**

From the render() method you return the JSX that builds the component interface.

It’s a pure method, so it should not cause side effects and should return the same output when called multiple times with the same input.

**componentDidMount()**

This method is the one that you will use to perform API calls, or process operations on the DOM.

**Updating**

When updating you have 5 lifecycle methods before the component is mounted in the DOM: the getDerivedStateFromProps, shouldComponentUpdate, render, getSnapshotBeforeUpdate and componentDidUpdate.

**getDerivedStateFromProps()**

See the above description for this method.

**shouldComponentUpdate()**

This method returns a boolean, true or false. You use this method to tell React if it should go on with the rerendering, and defaults to true. You will return false when rerendering is expensive and you want to have more control on when this happens.

**render()**

See the above description for this method.

**getSnapshotBeforeUpdate()**

In this method you have access to the props and state of the previous render, and of the current render.

Its use cases are very niche, and it’s probably the one that you will use less.

**componentDidUpdate()**

This method is called when the component has been updated in the DOM. Use this to run any 3rd party DOM API or call APIs that must be updated when the DOM changes.

It corresponds to the componentDidMount() method from the mounting phase.

**Unmounting**

In this phase we only have one method, componentWillUnmount.

**componentWillUnmount()**

The method is called when the component is removed from the DOM. Use this to do any sort of cleanup you need to perform.

**Legacy**

If you are working on an app that uses componentWillMount, componentWillReceiveProps or componentWillUpdate, those were deprecated in React 16.3 and you should migrate to other lifecycle methods.