Lifecycle of Components. **Day** 42/100

Each component in React has a lifecycle which you can monitor and manipulate during its three main phases.

The three phases are:

1). **Mounting**.

2). **Updating**.

3). **Unmounting**.

Mounting

Mounting means putting elements into the DOM.

React has four built-in methods that gets called, in this order, when mounting a component:

1. constructor()
2. getDerivedStateFromProps()
3. render()
4. componentDidMount()

The render() method is required and will always be called, the others are optional and will be called if you define them.

1). **constructor**

The constructor() method is called before anything else, when the component is initiated, and it is the natural place to set up the initial state and other initial values.The constructor() method is called with the props, as arguments, and you should always start by calling the super(props) before anything else, this will initiate the parent's constructor method and allows the component to inherit methods from its parent, React.Component.

```
class Header extends React.Component {
```

Example:

```
class Header extends React.Component {
  constructor(props) {
    super(props);
    this.state = {favoritecolor: "red"};
  }
  render() {
    return (
      <h1>My Favorite Color is {this.state.favoritecolor}</h1>
    );
  }
}
ReactDOM.render(<Header />, document.getElementById('root'));
```

2). **getDerivedStateFromProps**

The getDerivedStateFromProps() method is called right before rendering the element(s) in the DOM.

This is the natural place to set the state object based on the initial props.

It takes state as an argument, and returns an object with changes to the state.

The example below starts with the favorite color being "red", but the getDerivedStateFromProps() method updates the favorite color based on the favcol attribute:

Note that: The getDerivedStateFromProps method is called right before the render method.

Example:

```
class Header extends React.Component {
 constructor(props) {
    super(props);
    this.state = {favoritecolor: "red"};
  }
  static getDerivedStateFromProps(props, state) {
    return {favoritecolor: props.favcol };
  }
  render() {
    return (
      <h1>My Favorite Color is {this.state.favoritecolor}</h1>
    );
  }
}
ReactDOM.render(<Header favcol="yellow"/>,
document.getElementById('root'));
```

3). **render**

The render() method is required, and is the method that actually outputs the HTML to the DOM.

In the exampe below we have a simple component with a simple `render()` method:

Example:

```
class Header extends React.Component {
render() {
    return (
      <h1>This is the content of the Header component</h1>
    );
  }
}
ReactDOM.render(<Header />, document.getElementById('root'));
```

4). **componentDidMount**.

The componentDidMount() method is called after the component is rendered.This is where you run statements that requires that the component is already placed in the DOM.

Example

```
class Header extends React.Component {
 constructor(props) {
    super(props);
    this.state = {favoritecolor: "red"};
  }
  componentDidMount() {
    setTimeout(() => {
      this.setState({favoritecolor: "yellow"})
    }, 1000)
  }
  render() {
    return (
      <h1>My Favorite Color is {this.state.favoritecolor}</h1>
    );
  }
}

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

In the example above, at first my favorite color is red, but give me a second, and it is yellow instead.

Updating

The next phase in the lifecycle is when a component is updated. A component is updated whenever there is a change in the component's state or props.

React has five built-in methods that gets called, in this order, when a component is updated:

1). getDerivedStateFromProps()

2). shouldComponentUpdate()

3). render()

4). getSnapshotBeforeUpdate()

5). componentDidUpdate()

Unmounting

The next phase in the lifecycle is when a component is removed from the DOM, or unmounting as React likes to call it.

React has only one built-in method that gets called when a component is unmounted:

1). componentWillUnmount()

Tomorrow we will discus more on Updating and unmounting react component.

All the best, **Lux Tech Academy**.