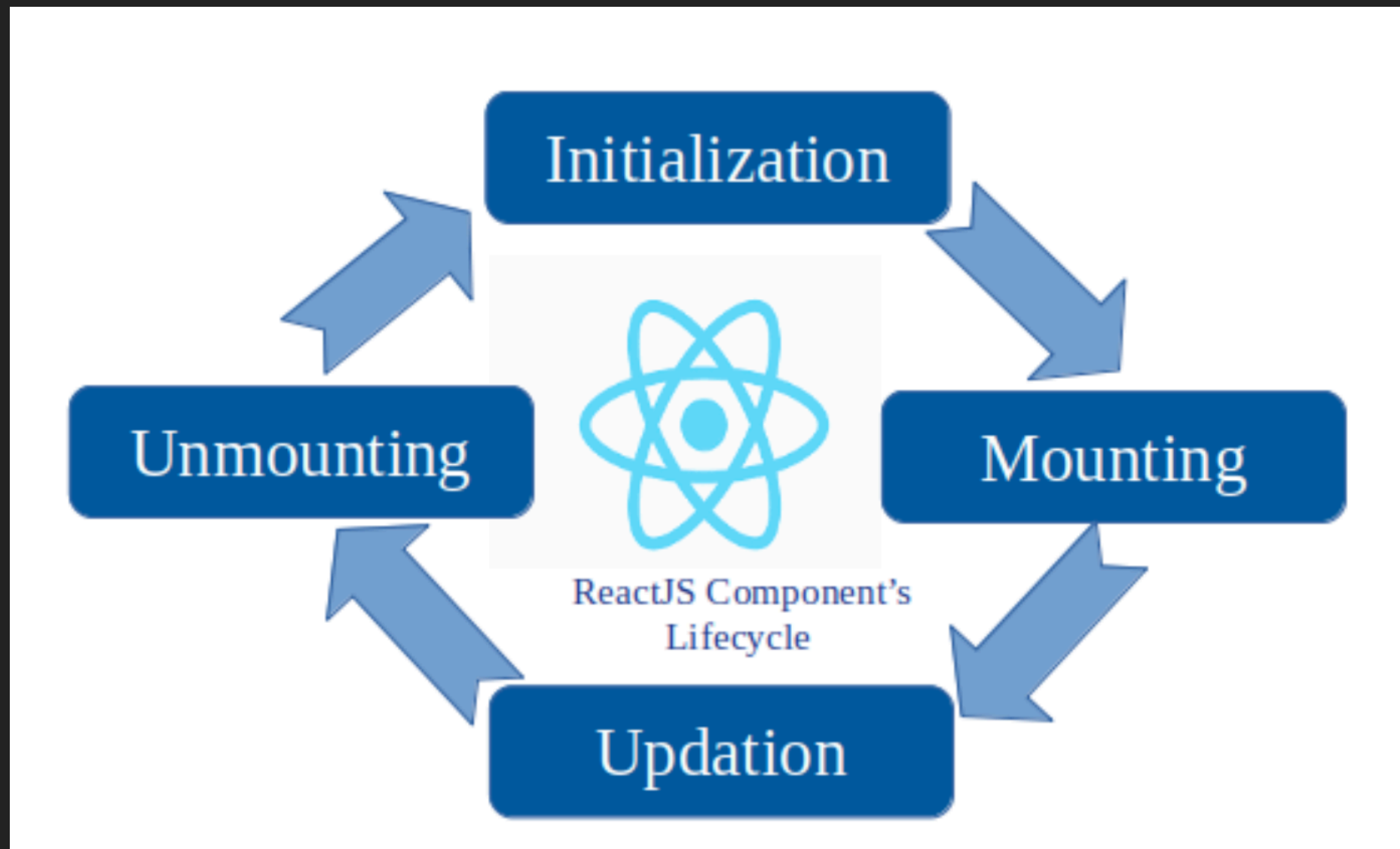


# REACT COMPONENT LIFECYCLE

BY QCC TECHWORKS

---

- ▶ A Components lifecycle is classified into four parts:
  - ▶ Initialization
  - ▶ Mounting
  - ▶ Updating
  - ▶ Unmounting



- ▶ Just like in life - Components in React are created (mounted on the DOM), grow by updating, and then die (unmount from the DOM).

### ► Initialization

- This is the phase in which the component is going to start its journey by setting up the state (see below) and the props. This is usually done inside the constructor method.

```
constructor() {  
  super();  
  this.state = {  
    items: [],  
  }  
  console.log("this is when the constructor is loaded");  
}
```

### ▶ Mounting

- ▶ Mounting is the phase in which our React Component mounts on the DOM.
- ▶ This phase is after the initialization is completed. In this phase our component executes the `render()` method for the first time.
- ▶ One lifecycle method we're going to use today is:

### 1. `componentDidMount()`

- ▶ This method is called after the component gets mounted on the DOM. Before the execution of this method, the render method is called (we can access the DOM). We can make API calls and update the state with the API response. It's called once in a lifecycle.

# REACT COMPONENT LIFECYCLE

---

```
class Lifecycle extends Component {
  constructor() {
    super();
    this.state = {
      items: [],
    }
    console.log("this is when the constructor is loaded");
  }

  componentDidMount () {
    console.log("This is when the componentDidMount method executed");
  }

  render () {
    console.log("this is when the render method executed")
    return (
      <div></div>
    );
  }
};
```

- ▶ What we should see in our console - is the `console.log()` message we placed in the render method first.
- ▶ Then, we should see our `componentDidMount()` lifecycle method `console.log()` execute after.

|  |                  |
|--|------------------|
| this is when the constructor is loaded             | <u>App.js:10</u> |
| this is when the render method executed            | <u>App.js:18</u> |
| This is when the componentDidMount method executed | <u>App.js:14</u> |



### ► Updating

- After the mounting phase where the component has been created, the update phase comes into the scene. This is where component's state changes and hence, re-rendering takes place.
- In this phase, the data of the component (state & props) updates in response to user events like clicking, typing etc... This results in the re-rendering of the component. The methods that are available in this phase are:

### 1. `shouldComponentUpdate()`

- ▶ This method determines whether the component should be updated or not. By default, it returns true. At some point, if you want to re-render the component on some condition, then `shouldComponentUpdate` method is the right place.
- ▶ For example, you want to only re-render your component when there is a change in props - then use the power of this method. It receives arguments `nextProps` and `nextState` which will help us decide whether to re-render by doing a comparison with the current prop value.

# REACT COMPONENT LIFECYCLE

```
class LifecycleTwo extends React.Component {
  constructor() {
    super();
    this.state = {
      value: true,
      countOfClicks: 0
    };
  }

  pickRandom = () => {
    this.setState({
      value: Math.random() > 0.5, // randomly picks true or false
      countOfClicks: this.state.countOfClicks + 1
    });
  }

  // comment out the below to re-render on every click
  shouldComponentUpdate(nextProps, nextState) {
    return this.state.value !== nextState.value;
  }

  render() {
    return (
      <div>
        shouldComponentUpdate demo
        <p><b>{this.state.value.toString()}</b></p>
        <p>Count of clicks: <b>{this.state.countOfClicks}</b></p>
        <button onClick={this.pickRandom}>
          Click to randomly select: true or false
        </button>
      </div>
    );
  }
}
```

### ► Unmounting

- This is the last phase in the component's lifecycle. The component gets unmounted from the DOM in this phase. The method that is available is:

#### 1. `componentWillUnmount()`

- This method is called before the unmounting of the component takes place. Before the removal of the component from the DOM, 'componentWillUnmount' executes. This method denotes the end of the component's lifecycle.

# REACT COMPONENT LIFECYCLE

---

- ▶ Links:

- ▶ [React Documentation](#)

- ▶ [MDN](#)