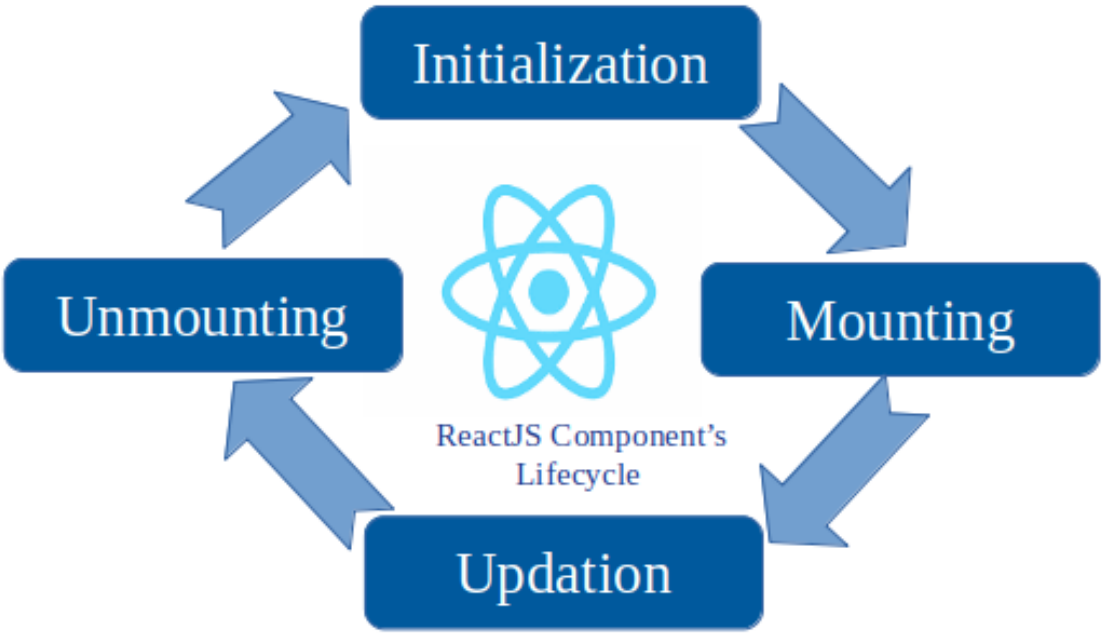
Mounting



Lifecycle of Components

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

The **three** phases are:

- **Mounting**
- **Updating**, and
- **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:

- **constructor()**
- **getDerivedStateFromProps()**
- **render()**
- **componentDidMount()**

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

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 allow the component to inherit methods from its parent (React.Component).

Example:

The constructor method is called, by React, every time you make a component:

```
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'));
```

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 the 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:

Example:

The **getDerivedStateFromProps** method is called right before the render method:

```
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'));
```

Render

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

Example:

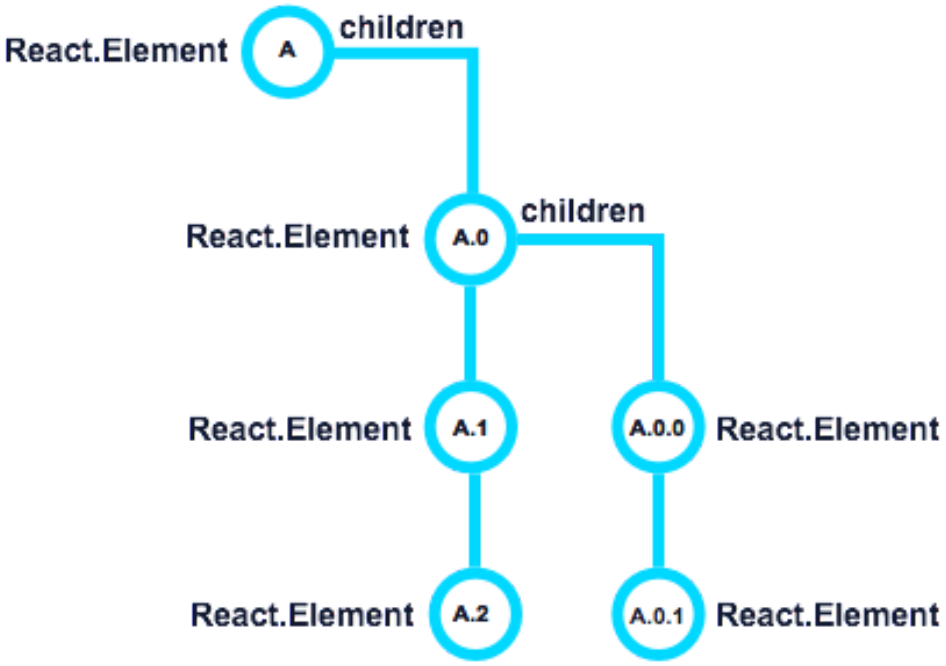
A simple component with a simple **render**() method:

```
class Header extends React.Component {

  render() {

    return (

      <h1>This is the content of the Header component</h1>

    );

  }

}


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

componentDidMount

The **componentDidMount**() method is called after the component is rendered.

This is where you run statements that require that the component is already placed in the DOM.



Example:

At first my favorite color is **red**, but give me a second, and it is **yellow** instead:

```
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'));
```