



ANECO ACADEMY

REACTJS - CLASS COMPONENT LIFE CYCLE METHODS

Lifecycle Methods

Mounting

When an instance of a component is being created and inserted into the DOM

Updating

When a component is being re-rendered as a result of changes to either its props or state

Unmounting

When a component is being removed from the DOM

Error Handling

When there is an error during rendering, in a lifecycle method, or in the constructor of any child component

CLASS COMPONENT - LIFECYCLE METHODS

Lifecycle Methods

Mounting

constructor, static `getDerivedStateFromProps`, `render` and `componentDidMount`

Updating

static `getDerivedStateFromProps`, `shouldComponentUpdate`, `render`, `getSnapshotBeforeUpdate` and `componentDidUpdate`

Unmounting

`componentWillUnmount`

Error Handling

static `getDerivedStateFromError` and `componentDidCatch`

Mounting Lifecycle Methods

`constructor(props)`

A special function that will get called whenever a new component is created.

Initializing state
Binding the event handlers

Do not cause side effects. Ex: HTTP requests

`super(props)`
Directly overwrite `this.state`

Mounting Lifecycle Methods

`constructor(props)`



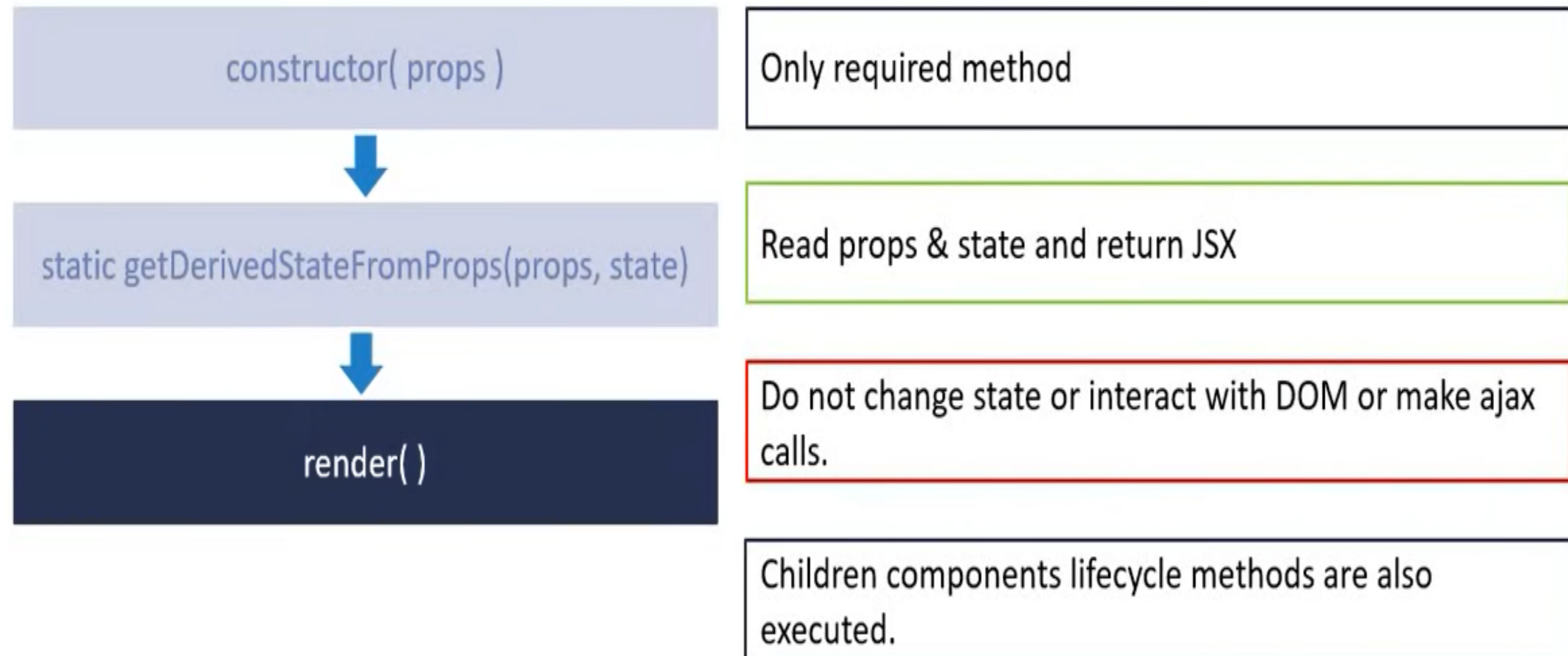
`static getDerivedStateFromProps(props, state)`

When the state of the component depends on changes in props over time.

Set the state

Do not cause side effects. Ex: HTTP requests

Mounting Lifecycle Methods



Mounting Lifecycle Methods

`constructor(props)`

Invoked immediately after a component and all its children components have been rendered to the DOM.



`static getDerivedStateFromProps(props, state)`

Cause side effects. Ex: Interact with the DOM or perform any ajax calls to load data.



`render()`



`componentDidMount()`

CLASS COMPONENT - LIFECYCLE METHODS - EXAMPLE

```
constructor(prop) {  
  super(prop);  
  console.log('[App.js] constructor');  
  this.state={  
    product:[  
      {name:"sony",price:"200"},  
      {name:"Samsung",price:"200"}  
    ]  
  }  
}  
  
//Change2  
static getDerivedStateFromProps(prop, state) {  
  console.log('[App.js] getDerivedStateFromProps', prop);  
  return state;  
}  
  
//Change3  
componentDidMount() {  
  console.log('[App.js] componentDidMount');  
}
```


Updating Lifecycle Methods

`static getDerivedStateFromProps(props, state)`

Method is called every time a component is re-rendered

Set the state

Do not cause side effects. Ex: HTTP requests

Updating Lifecycle Methods

`static getDerivedStateFromProps(props, state)`



`shouldComponentUpdate(nextProps, nextState)`

Dictates if the component should re-render or not

Performance optimization

Do not cause side effects. Ex: HTTP requests
Calling the `setState` method

Updating Lifecycle Methods

`static getDerivedStateFromProps(props, state)`

Only required method



`shouldComponentUpdate(nextProps, nextState)`

Read props & state and return JSX



`render()`

Do not change state or interact with DOM or make ajax calls.

Updating Lifecycle Methods

`constructor(props)`

Called right before the changes from the virtual DOM are to be reflected in the DOM



`static getDerivedStateFromProps(props, state)`

Capture some information from the DOM



`render()`

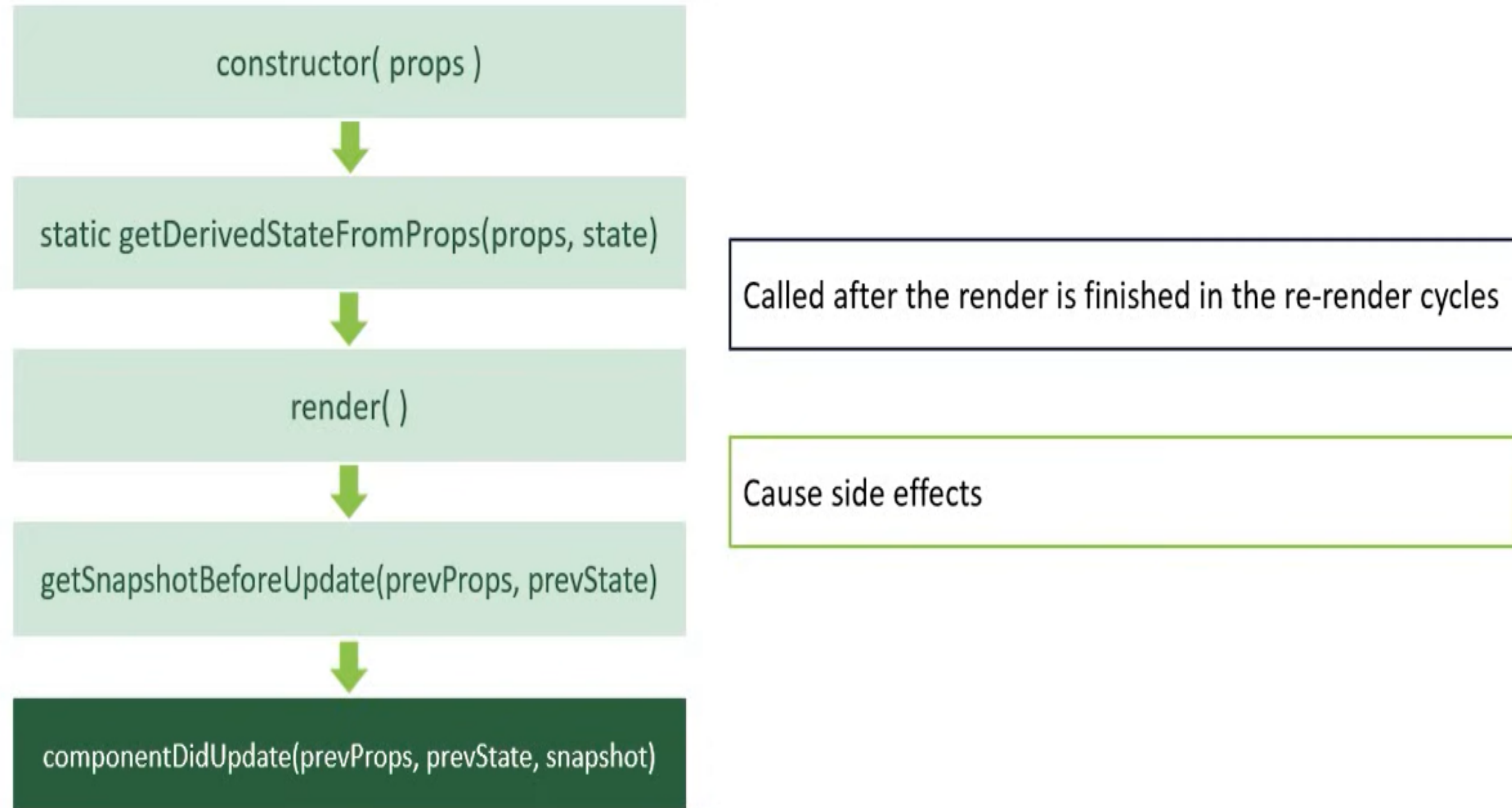
Method will either return null or return a value. Returned value will be passed as the third parameter to the next method.



`getSnapshotBeforeUpdate(prevProps, prevState)`

CLASS COMPONENT - LIFECYCLE METHODS - UPDATING

Updating Lifecycle Methods



CLASS COMPONENT - LIFECYCLE METHODS - EXAMPLES

```
// static getDerivedStateFromProps(props, state) {  
//   console.log('[Product.js] getDerivedStateFromProps', props);  
//   return state;  
// }
```

```
shouldComponentUpdate(nextProps, nextState) {  
  console.log('[Product.js] shouldComponentUpdate');  
  return true;  
}
```

```
componentDidUpdate(prevProps, prevState, snapshot) {  
  console.log('[Product.js] componentDidUpdate');  
  console.log(prevProps, prevState, snapshot);  
}
```

Unmounting Phase Method

`componentWillUnmount()`

Method is invoked immediately before a component is unmounted and destroyed.

Cancelling any network requests, removing event handlers, cancelling any subscriptions and also invalidating timers.

Do not call the `setState` method.

Error Handling Phase Methods

`static getDerivedStateFromError(error)`

`componentDidCatch(error, info)`

When there is an error either during rendering, in a lifecycle method, or in the constructor of any child component.