

Class based component → It javascript class that returns a

Episode 8 th

23/02/2024

① created User.js

```
return (  
  <div className = "user-card">
```

② UserClass.js
import React from "react";

⇒ class UserClass extends React.Component {

render method → render () {

```
  return (  
    <div className = "user-card">  
      <h2> Name : Mandar </h2>  
      <h3> Location : Deheradun </h3>  
      <h4> Contact: 9) mandartule </h4>  
    </div>);  
}
```

export default UserClass;

③ Import it in about us page and add ~~it~~ it there.

• Props in class based Component

Props in function component

• Argument class

```
<UserClass name = {"Mandar"} />
```

• Parameter

```
class UserClass extends React.Component {  
  constructor(props) {
```

```
    super(props);  
    console.log(props);  
  }
```

```
  render() {
```

```
    return (  
      <div>  
        <h1> {props.name this.props.name} </h1>  
      </div>
```

Use this key word always

• Arguments

```
<User name = {"Mandar"} />
```

• Parameter

```
const User = (props) => {
```

```
  return (  
    <div>
```

```
      <h2> {props.name} </h2>  
    </div>
```

Q why do we use super props →

Earlier there was no use state it was created in constructor.

Syntax

```
constructor(props) {
```

```
  super(props);
```

```
  this.state = {
```

```
    count: 0,
```

```
    count2: 2,
```

```
  };
```

→ this contain All our state variables

```
  render() {
```

```
    const {name, location} = this.props;
```

```
    const {count, count2} = this.state;
```

```
    return(
```

```
      <div> className = "user-card">
```

```
        <h1> Count : {count} </h1>
```

```
        <button onClick = {() => {
```

```
          this.setState({
```

```
            count: this.state.count + 1,
```

```
          };
```

```
        } </button>
```

```
      </div>
```

```
    )
```

Function given by react to update state variable.

Never Update state variables Directly

How the component is mounted on the webpage?
loading

- ⇒
- 1) When class loads first then the constructor is called
 - 2) then render is called.
 - 3) then the componentDidMount() called.

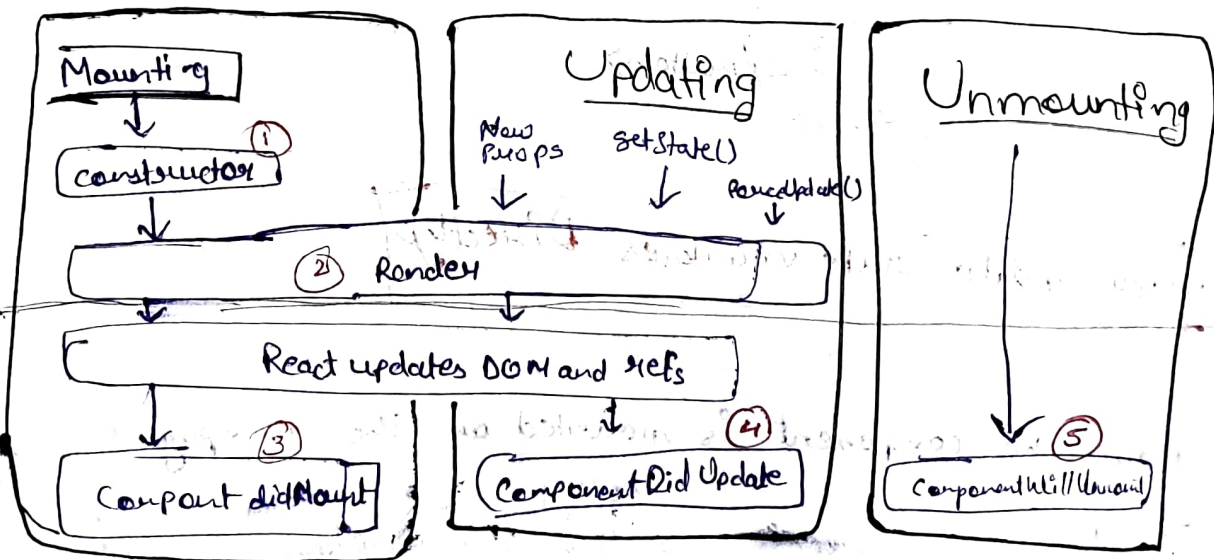
ComponentDidMount() { } → this is a function in class component that is called when the component is mounted.
used to make api call

*** Component Did Mount** The api call is made ~~the~~ after the component is rendered because to make the React fast. API call will take time so first render the component then fill the data inside. Otherwise React will have to wait to render.

When there are multiple child this it work like this

- ① Parent constructor
- ② Parent render
- ③ first child constructor
- ④ first child Render.
- ⑤ second child constructor
- ⑥ second child Render.
- ⑦ First child ComponentDidMount is called.
- ⑧ second child component Did Mount is called
- ⑨ Parent component Did Mount is called.

React Life cycle diagram www.wojtekmaej.pl/react-life-cycle



* React is doing this because → Because dom manipulation is very expensive in time so React tries to batch up all the render process so that it can do all the things in one go. That's why react is fast.

→ Remove the count and state variables.

① Create the get user api from github
<https://api.github.com/user/Username>
get it from here

② Make the componentDidMount Async

```
async componentDidMount() {  
  const data = await fetch("https...");  
  const json = data.json();  
  console.log(json);  
}
```

③ Create a use state variable object userInfo : {
 name: "Dummy",
 location: "Default",
 },
};

```
this.setState({  
  userInfo: json,  
});
```

```
const {name, location} = this.state.userInfo;
```

```
return (  
  <div className="user-card">  
    <h2> Name: {name} </h2>  
    <h2> location: {location} </h2>  
  </div>  
);
```

cycle

Render Phase

Pure and has
no side effects

May be paused,
aborted or restarted
by react

Commit Phase

Can work with DOM
run side effects

① → First the component is made on
the page is rendered with
dummy data.

② → Then the ~~one~~ component mount is
called the update cycle
begins.

③ → the set state is called

④ → react re renders with new state variable

IV Then component Did update is called at last.

↳ after the componentDidMount

V ComponentWillUnmount will be called at last

ComponentWillUnmount()

→ this all is done easy in new version of react.
So react is popular