Ep 08: Let's Get Classy

apsara

Date: __/__/____

→ Creating class based components

Class based component in react is just only normal javascript class.

e.g

```
import React from 'react';

class UserClass extends React.Component {
    render() {
    return (
        <div>
            <h1> Aditi Deokar </h1>
            <h3> Pune </h3>
            <h4> @aditi17 </h4>
        </div>
    );
};};

export default UserClass;
```

→ Importing class component

```
Import UserClass from "./UserClass";

<UserClass/>
```

→ Passing props into class component:

- About.js

```
<UserClass name = "Aditi (class)"
        location = "Pune, Maharashtra" />
```

- UserClass.js

```
class UserClass extends React.Component{

    constructor (props) {
        super (props);
        console.log(props);
    }

render() {
 const {name, location} = this.props;

 return {
    <div>
      <h1> {name} </h1>
      <h2> {location} </h2>
      </div>
  );
 };
};
```

↳ Creating State Variable In Class Component ↳

Loading a class component on web page means creating Instance of class.

whenever we create Instance of class, the constructor get called.

Constructor is best place to create local state variables.

There is no any hook like functional component to create state variables

```
constructor (props) {
    super (props);

    this.state = {
      count: 0,
      count2: 2
    };
}

render() {
const { count, count2 } = this.state;

return (
   <h1> { count } </h1>
   <h1> { count2 } </h1>
)
}
```

↳ Updating state variable in class component

```
<button onClick = { () => {
    this.setState ( {
      count: this.state.count + 1 ;
  });
}} > Count </button>
```

Note: this.state() variable is an big object

↳ Life cycle of class based component

- UserClass.js

```
class Userclass extends React.Component{
  constructor (props){
    super(props);

    Console.log( "child Constructor");
  }

  componentDidMount() {
    Console.log ( " child component
    did Mount");
  }

  render() {

    console.log (" child render");

  }
```

- Parent Class (About.js)

```
class About extends React.Component{

  constructor (){
    console.log ("Parent Constructor");
  }

  componentDidMount() {
      Console.log ("parent component did
                    mount");  }
```

```
render() {
  Console.log ("parent render");
  }
}
```

Output:

```
Parent constructor
Parent Render
Child constructor
Child render
Child Component did Mount
Parent Component did Mount
```

→ componentDid Mount()

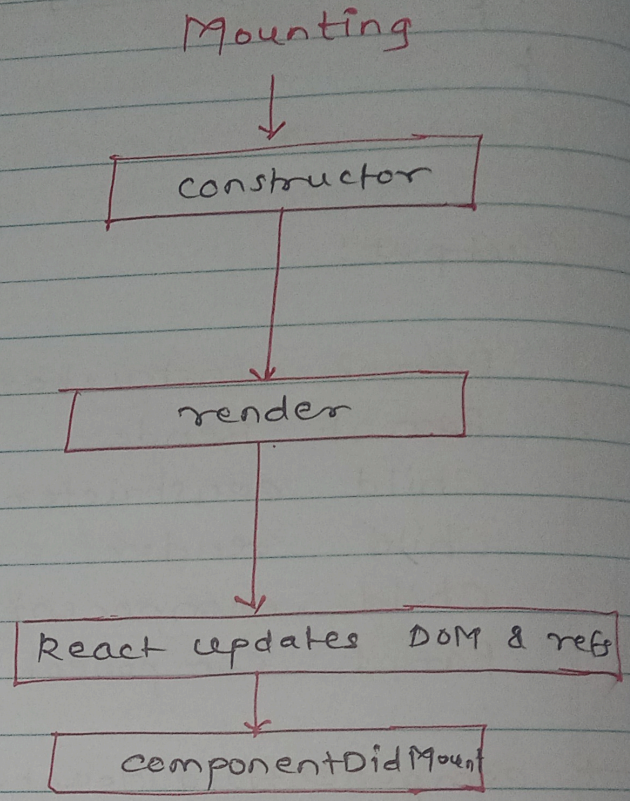
- Used for making API calls in class based components

→ If there are more than one children component in parent

~~First Chit~~

```
Parent Constructor
Parent Render
FirstChild Constructor
FirstChild render
SecondChild Constructor
SecondChild render
FirstChild Component did Mount
SecondChild component did Mount
Parent component did Mount
```

↳ React - Life Cycle Methods Diagram



Render phase
pure & has
no side effects,
maybe paused,
aborted or
restarted by
React

Commit phase
can work with
DOM, run side
effects, schedule
updates

In commit phase, DOM manipulation Begins and it happens in a single batch.

↳ Making api call in componentDidMount in class based component

```
async componentDidMount () {

const data = await fetch ("https://");
const json = await data.json();

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

- In class based component, we can make api call in componentDidMount() function by making it async function.

↳ componentDidUpdate()

- call after 2nd render with api data
- call when we update state variables

↳ Component Life cycle