"Key should be On Parent JSX" while using MAP

Let's get Classy

Functional Component

1. A functional Component is just a plain Javascript pure function that accepts props as an argument and returns a React element (JSX)
2. There is no render method used in functional components.
3. Functional components run from top to bottom and once the function is returned it can't kept alive.
4. Also known as Stateless component as they simply accept data and display them in some form, they are mainly responsible for rendering UI.
5. React life cycle methods (e.g. componentDidMount()) can't be used in functional components

Class Component

1. A class component requires you to extend from React Component and create a render function that returns a React Element
2. It must have the render() method returning JSX (which is syntactically similar to HTML)
3. The class component is instantiated and different life cycle method is kept alive and is run and invoked depending on the phase of the class component
4. Also known as Stateful Components because they implement logic and state.
5. React lifecycle methods can be used inside class components (e.g componentDidMount()).

(6) Hooks can be easily used in functional components to make them stateful.

e.g.

```
const [name, setName] =
  React.useState('')
```

(7) Constructors are not used

(7) It requires different Syntax inside a class component to implement hooks.

e.g.

```
constructor (props){
  super(props);
this.state = { name: '' }
}
```

(7) Constructor is used as it needs to store state.

Functional Component:

```
const User = ({name}) => {
  return (
   <div className="user-card">
      <h2> Name : {name}</h2>
      <h3> Location : Dehradun</h3>
      <h4> Contact : @akshaymarch7</h4>
   </div>
  );
};
```

Class Component

```
Class UserClass extends React.Component {
    constructor(props){
        super(props);
        console.log(props);
    }
    render(){
        const { name, location } = this.props;

        return (
            <div class Name="user-card">
            <h2> Name : {name} </h2>
            <h3> Location : {location} </h3>
            <h4> Contact : @akshaymarch7 </h4>
            </div>
        );
    }
}
```

Way of passing the data through props is same. way of accepting is different b/w functional and Class Component.

Creating State Variables in Class based Component

```
class UserClass extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      count: 0,
      count2: 2,
    };
  }
```

← big object which stores all state variables

best practice to write

Note:
Never Update State Variable directly e.g

```
this.state.count = this.state.count + 1;
```

Instead we use

```
this.setState({
  count: this.state.count + 1,
});
```

← we can use this anywhere inside class

↓ React will only update the the state variable written here and will not touch any other state variable.

Lifecycle of class based Component:

First Constructor called whenever class based component ~~Renders~~ mounted then render called after that component Did Mount called.

- Parent Constructor
- Parent Render
- Child Constructor
- Child Render
- Child component Did mount
- Parent Component Did mount

Learn More from Notes of Akshay Saini

We did API call inside ComponentDidMount() because first we renders the basic structure of component then makes API call.

So

Constructor

Render

ComponentDid mount → Do API call → Then Re renders the component with data

Life Cycle of Multiple Childs.

Parent Constructor

Parent Render

- First child Constructor
- First child Render
- Second child Constructor
- Second child Render

} Render phase is fast
only Commit phase takes time.

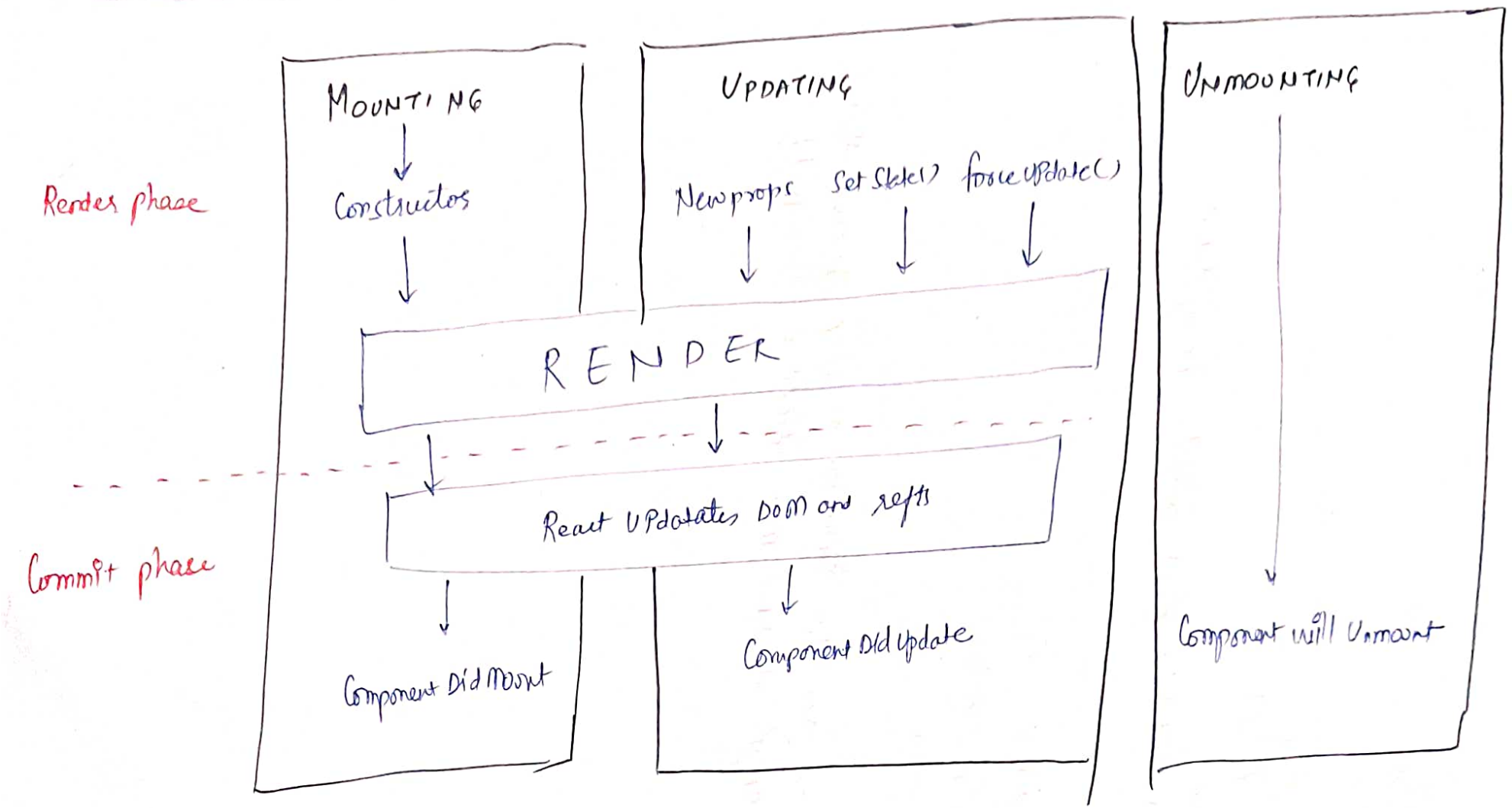
<DOM UPDATED - in single Batch> → ∵ Dom manipulation is most expensive.

- first child Component Did Mount
- Second child Component Did Mount

- Parent Component Did Mount.

For Effective Dom Manipulation first Render phase for all child then Dom updated in a single time and then Component Did Mount will happen for all the child.

Same for any no. of child.


Render phase
Commit phase
MOUNTING
Constructor
UPDATING
New props
setState()
forceUpdate()
RENDER
React updates DOM and refs
Component Did Mount
Component Did Update
UNMOUNTING
Component will Unmount

How to Make API Call in Class Based Component

MOUNTING

Constructor (dummy)

Render (dummy)

<HTML Dummy>

Component Did Mount

<API Call>

<this.setState> → State variable is updated.

--- UPDATE

render(API data)

<HTML (new API data)>

Component Did Update.

Skipped last 45 mins See before Interview or skip because that much Info is enough if need more deepdive then watch.