

### 3. 3. Class / VS functional Components

06 May 2024 14:09

Just see the structure & don't try to understand anything

Class Based functional components:, counter application:

```

    </div>
  </div>
);
}

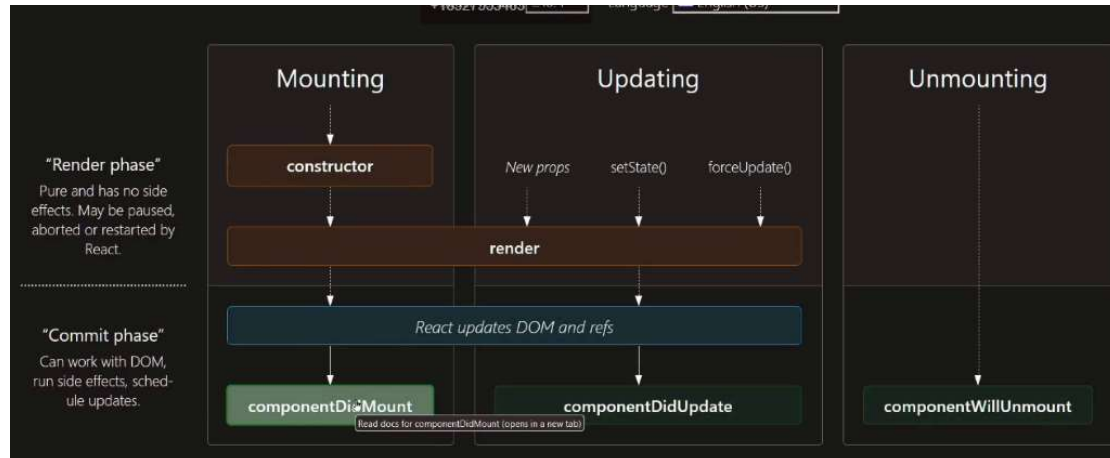
class Counter extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      count: 0,
    };
  }

  incrementCount = () => {
    this.setState({count: this.state.count + 1});
  };

  render() {
    return (
      <div>
        <p>Count: {this.state.count}</p>
        <button onClick={this.incrementCount}>Increment</button>
      </div>
    );
  }
}

```

We use super keyword to access the parent class constructor in our case it's (React.C



Lifecycle

Functional Based components & are hooks

```

    </div>
  </div>
);
}

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

  incrementCount = () => {
    this.setState({count: this.state.count + 1});
  };

  render() {
    return (
      <div>
        <p>Count: {this.state.count}</p>
        <button onClick={this.incrementCount}>Increment</button>
      </div>
    );
  }
}

```

```

const PhotoGallery = () => {
  const [photos, setPhotos] = React.useState([]); // useState Hook

```

```

    React.useEffect(() => {
      fetch("https://jsonplaceholder.typicode.com/photos")
        .then((res) => res.json())
        .then((data) => setPhotos(data))
        .catch((err) => console.error(err));
      return () => {};
    }, []);

    return (
      <div>
        <h3>Photo Gallery</h3>
        <div>
          {photos.splice(0, 5).map((photo) => {
            return <img key={photo.id} src={photo.thumbnailUrl} />;
          })}
        </div>
      </div>
    );
  };
}

```

This displays 5 images

Convert Class based components -> Functional Based Components

```

class DataList extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      data: [],
      isLoading: true,
      error: null,
    };
  }
}

```

We've 3 states here now in a functional comp. Create these states->

```

function DataListFn() {
  const [data, setData] = React.useState([]);
  const [isLoading, setIsLoading] = React.useState(true);
  const [error, setError] = React.useState(null);
}

```

