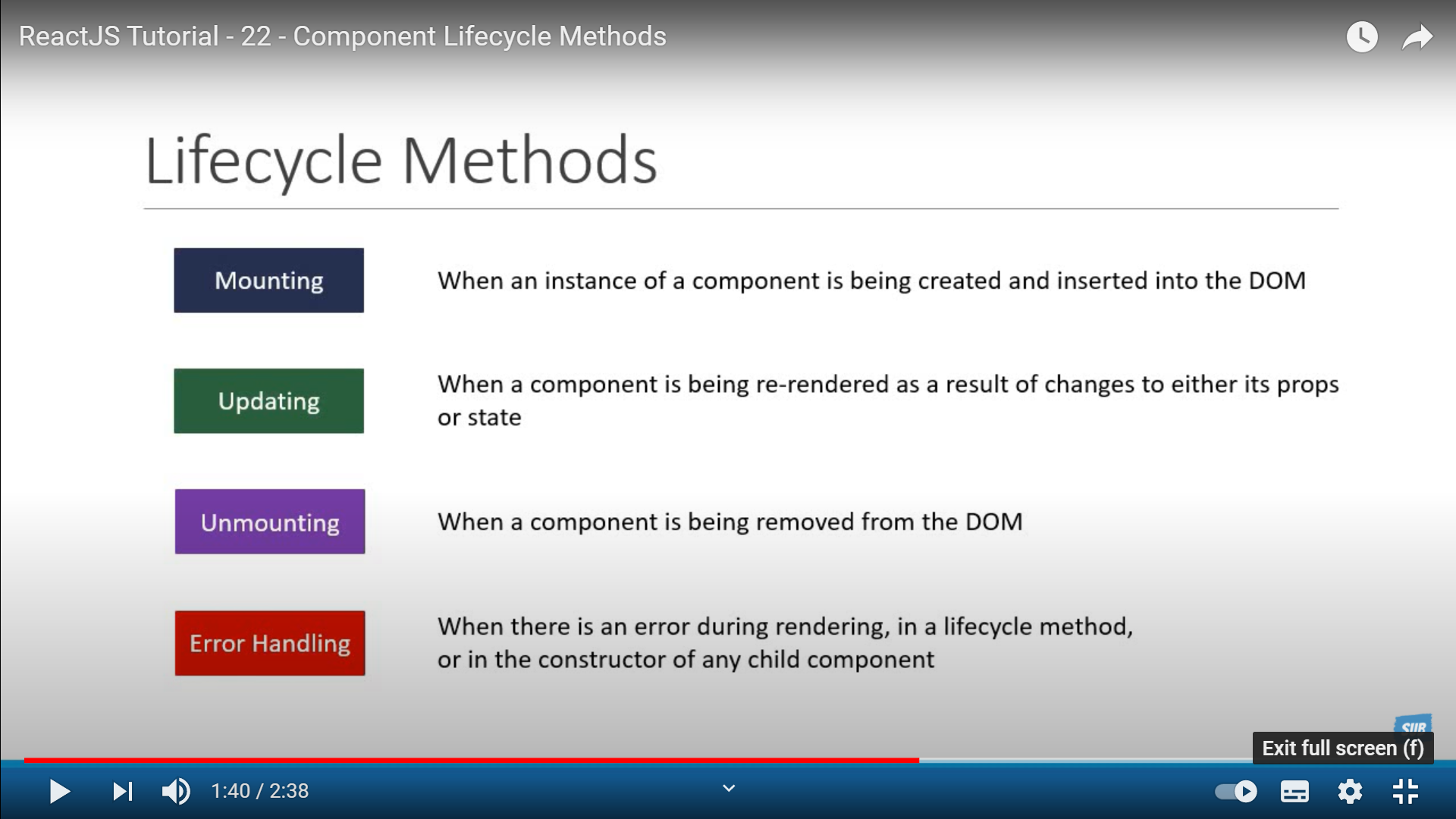
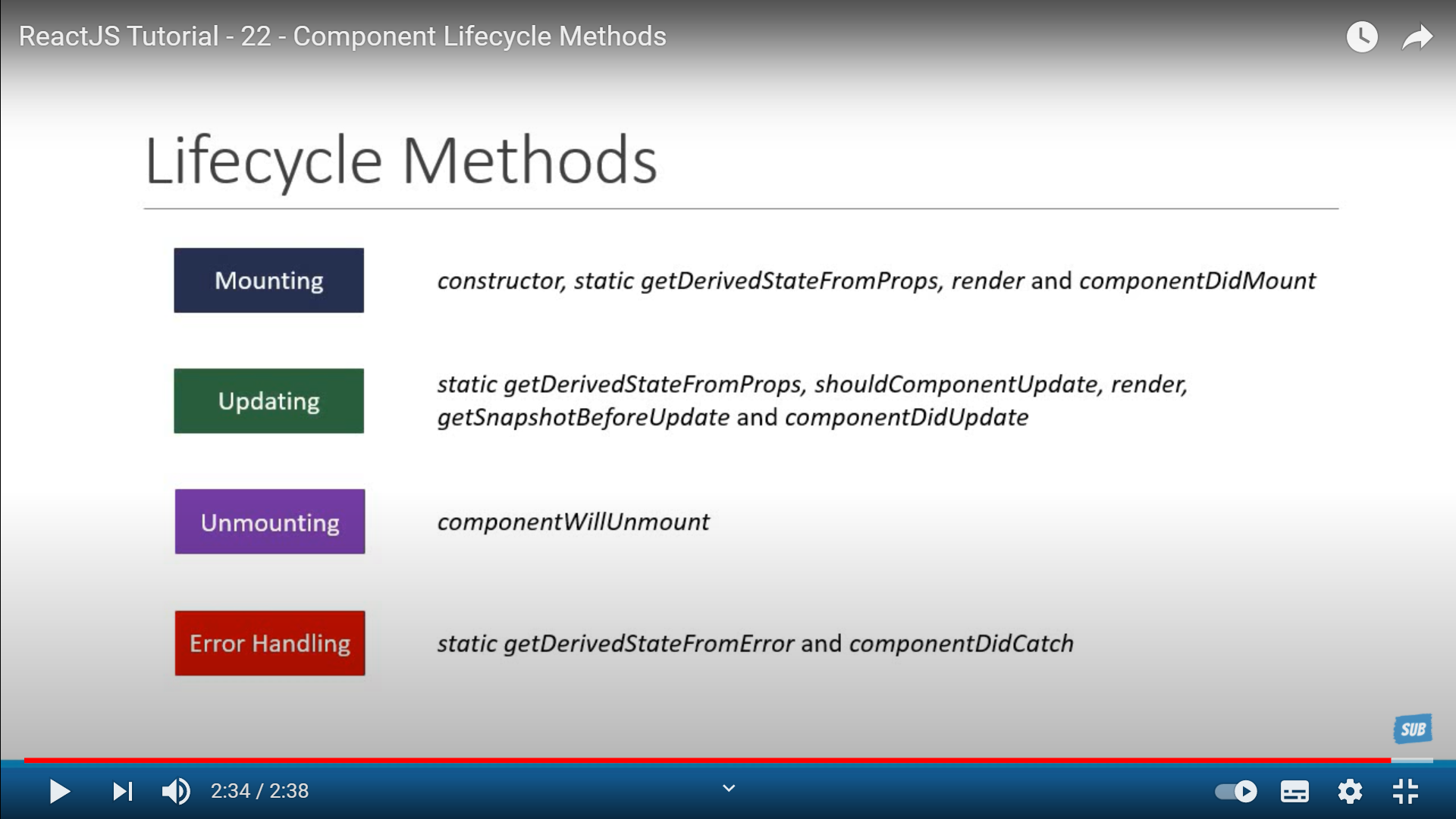
3Stages\Phases of lifecycle

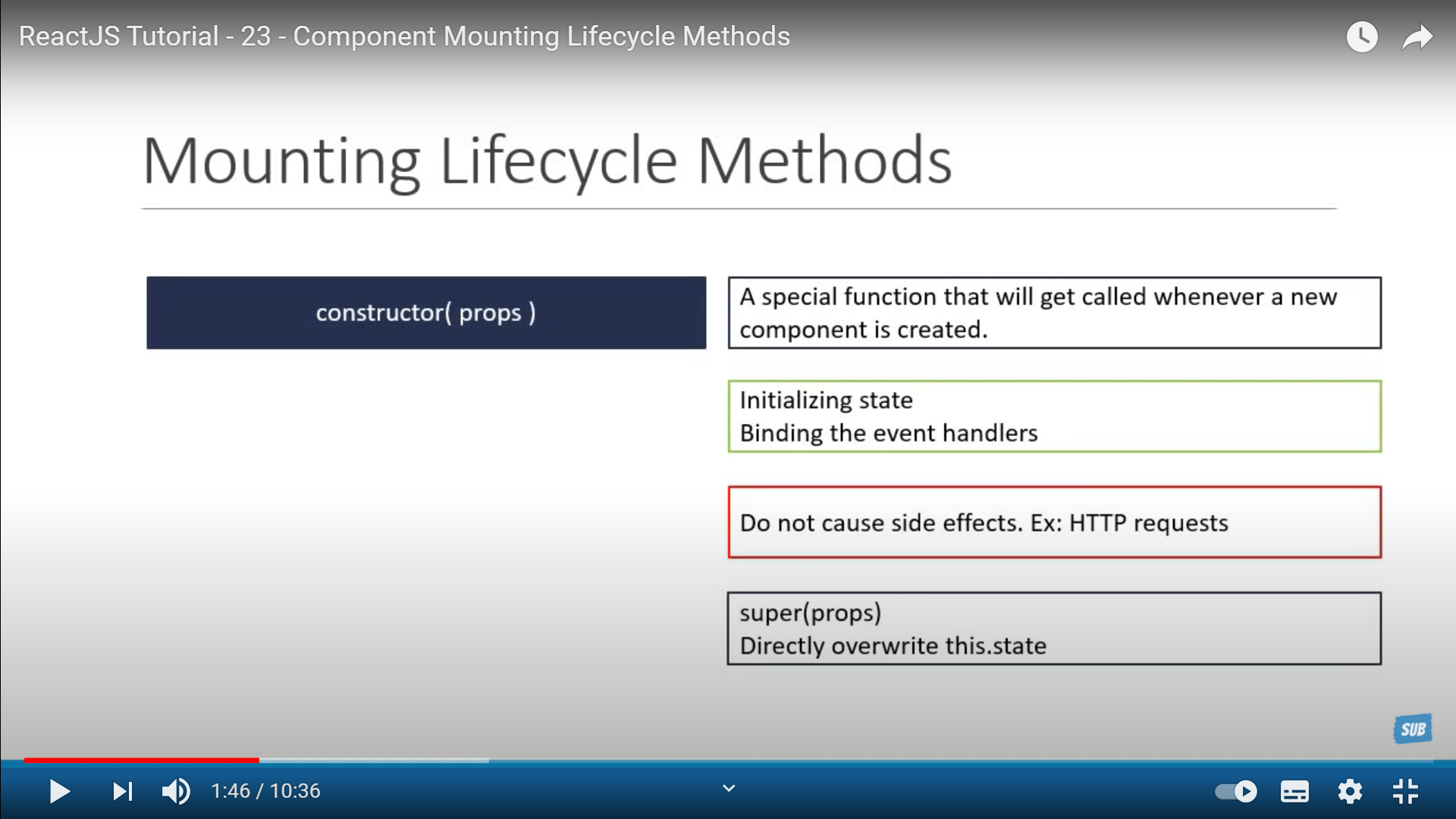


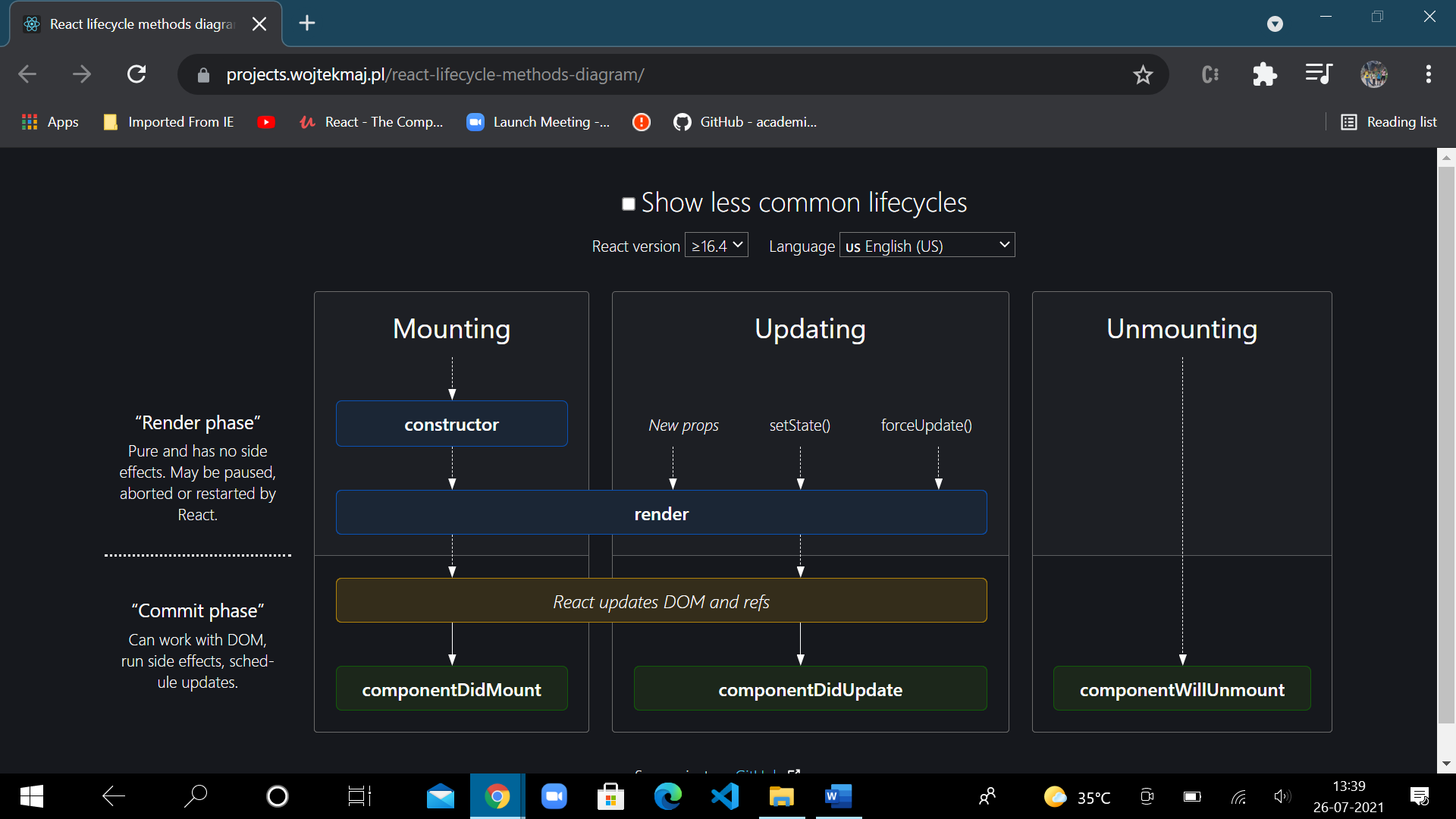
Mounting: First time it is shown/rendered on page. Each instance go thru its own life cycle.

Update: When the state or props of the component changes, the components get rerendered, this called update phase.

Unmounting: No longer want to show an react component on the page.Hiding a component or go to different page hence component is not shown







ComponentDidMount : As soon as component mounted, this method would be invoked

ComponentDidUpdate: As soon as component updated,no matter howmany times, this method would be invoked

ComponentWillUnmount: when the component is about to be unmounted and no longer shown this method would be invoked

If you’re familiar with React class lifecycle methods, you can think of useEffect Hook as componentDidMount, componentDidUpdate, and componentWillUnmount combined.

// componentDidMount—use 2nd parameter empty array[] for only mount phase

  // useEffect(() => {

  //   console.log("The use effect ran");

  // }, []);

  // // componentDidUpdate –side effects happen only when there is change in //count and not for all rerendering operation

  // useEffect(() => {

  //   console.log("The use effect ran");

  // }, [count]);

  // componentWillUnmount –use return section for unmount task

  // useEffect(() => {

  //   console.log("The use effect ran");

  //   return () => {

  //     console.log("the return is being ran");

  //   };

  // }, []);

//componentDidUpdate

//update phase alone code runs as we are using the variable didmountref to //know whether it is mounting or updating phase

    const didMountRef = useRef(false)

    useEffect(() => {

        if (didMountRef.current) {

            console.log("The use effect update ran --Update phase");

        } else didMountRef.current = true

    }, [count]);