**React Udemy Theory**

* If we need to update state that is based on the previous state it is safer
* to save the state using a function in the setState so it will be based on the previous value
* Controlled/smart/stateful components vs dumb/presentation/stateless components - when the
* components manage the state or only present it
* JSX awaits for each components to return one element, so we can use a Wrapper that return only the props.children( React Fragment)
* refs shouldn't be used when we need to manipulate the dom... we should use useState for that... Use Refs only when we need to read a value. (ref holds the actual DOM element)
* useReducer is used when the one state is depending its value on the other state value
* useCallback to save a function through compoent rerender for React Memo
* useCallback have same conditiion for changing like useEffect
* useMemo is used when we need to store some data into a component that is very heavy and it doesn't change too much
* in class components the state needs to be initialized in constructor and the state is an object with all the data needed into the component(it can't be splitted)
* in class compoentent the setState merges previous state, in function components you need to preserve the last state if you store the state in an object
* error boundaries are using react components
* using bind here(createdTask.bind(null, taskText)) will pre-append a value, but it will not execute it. Any other argument will be appended after our pre-appended argument
* redux - never mutate the previous state(in reducers or in components) => return new object always
* -in react-router 6 the routes search for best fit (order doesn't matter and the exact is not needed anymore)
* activeClassName replaced with className((navData)=>{})
* link and nested routes are now relative, the previous path construction is not needed, only the path we want to append
* there is no more useHistory, it is used the useNavigate hook

Useful Links from the course:

* <https://academind.com/tutorials/angular-vs-react-vs-vue-my-thoughts>
* <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/Arrow_functions>
* <https://academind.com/tutorials/reference-vs-primitive-values>
* <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Closures>
* <https://academind.com/tutorials/this-keyword-function-references>
* <https://academind.com/tutorials/hide-javascript-code>
* <https://academind.com/tutorials/rest-vs-graphql>
* <https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>
* <https://academind.com/tutorials/function-bind-event-execution>
* <https://academind.com/tutorials/hide-javascript-code>
* <https://academind.com/tutorials/reactjs-a-custom-useform-hook>
* <https://academind.com/tutorials/localstorage-vs-cookies-xss>
* <https://www.w3.org/TR/html-aria/#docconformance>
* <https://academind.com/tutorials/angular-vs-react-vs-vue-my-thoughts>
* <https://academind.com/tutorials/what-is-state>
* <https://academind.com/tutorials/reactjs-image-upload>