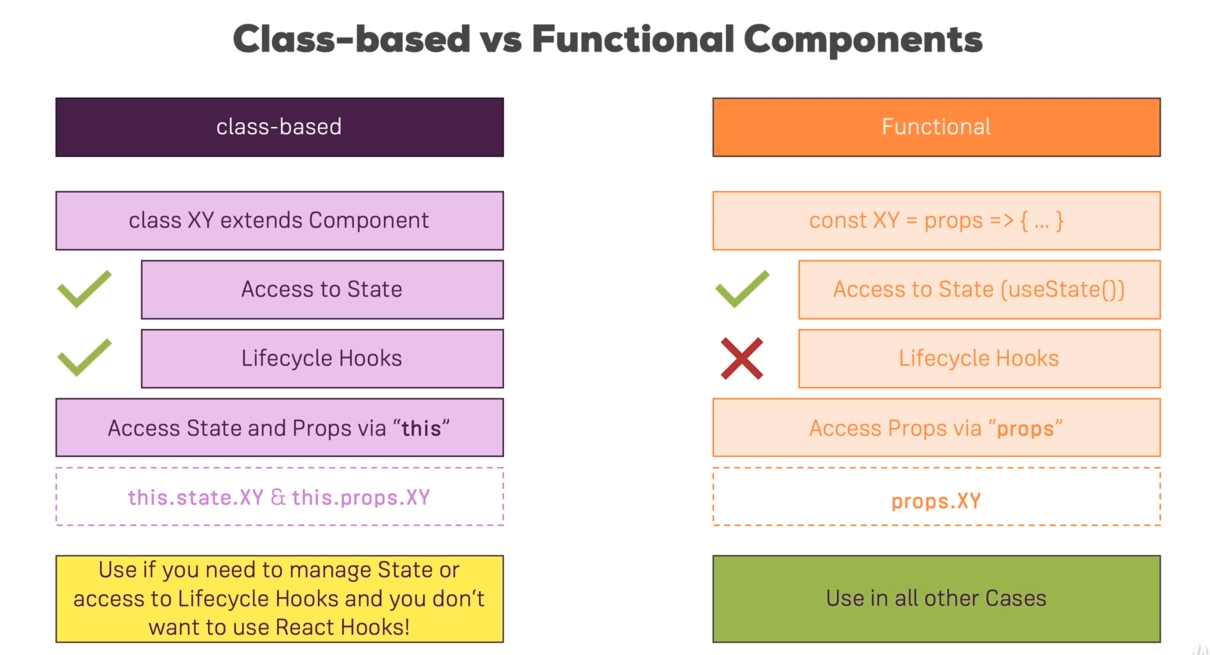
**Stateful and stateless:** same as flutter. Good practice is to use as many stateless as possible.

In the past, only class-based components could manage state. So, in older react project we couldn’t use functional component to manage state. Something to keep in mind when working on older react project.

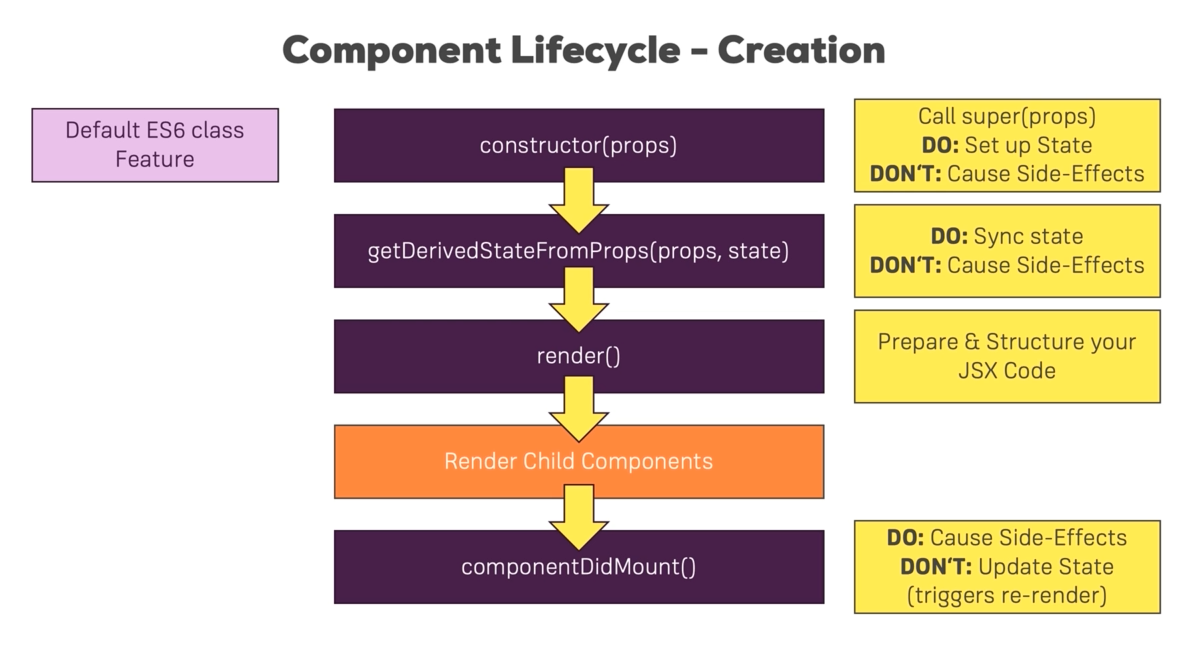
Functional components still can’t use lifecycle hooks yet. That’s probably only major different now. (Update: now they can using react hooks like useState, useEffect etc)



**Component Lifecycle:** Its only available in class-based components. When a class-based component is created, first the constructor execute. If we want, we can setup initial state here.

**Cause side effects**: things like http request, use local storage, send google analytics etc.

1. **Creation:**



There was other legacy lifecycle as well like **componentWillMount(), componentDidMount()** etc. They don’t have much use that we can’t do with other ones, so not much important.

1. **Update:**

* When state or prop changes only then react reevaluate the component.
* This lifecycle starts from **getDerivedStateFromProps**.This **getDerivedStateFromProps** method isn’t much useful other than the cases when we need to handle some external state information during each update. But there usually are better ways to do that.

