

## ۱- state management چیست ؟

State management یکی از ویژگی های مهم و اجتناب ناپذیر هر برنامه پویا است . React یک API ساده و انعطاف پذیر برای پشتیبانی از state management در یک component ارائه می دهد. در واقع React یک data store داینامیک برای هر component فراهم می کند. اگر کاربر با تعامل با برنامه ، state را تغییر دهد ، ممکن است رابط کاربری بعد از آن کاملاً متفاوت بنظر برسد زیرا component با فراخوانی متد render() به همراه state جدید خود را دوباره رندر می کند.

برنامه های React، state خود را بصورت داخلی مدیریت می کنند و برای برنامه هایی با component کم به خوبی کار می کنند. اما زمانی که برنامه بزرگتر می شود پیچیدگی مدیریت state های به اشتراک گذاشته شده در بین component ها دشوار می شود و پیدا کردن اینکه وقتی مشکلی پیش می آید چه اتفاقی می افتد واقعا سخت است . به همین دلیل است که به state management نیاز داریم و برای این کار از Redux, Hooks, Recoil می توان کمک گرفت .

## ۲- functional components چیست ؟

Functional components از رایج ترین component هایی هستند که هنگام کار با در React با آن ها مواجه می شویم . این توابع ممکن است داده ها را بعنوان پارامتر دریافت کنند یا نکنند . در functional components مقداری که return می کنیم کد JSX است تا به درخت DOM رندر شود .

چندین نوع react component در گذشته رایج بود . قبل از معرفی Hook بعنوان یک component بدون state نیز شناخته می شد . با روی کار آمدن React Hooks، اکنون این امکان وجود دارد که کل برنامه را فقط از طریق function ها بعنوان React component ها بنویسید . و در صورت نیاز از props بعنوان ورودی استفاده می شود. functional components خواندن و تست کردن و دیباگ کردن آسان تری دارند.

### ۳-Hook چیست ؟

Hook ها توابعی هستند که شما را به React state و ویژگی های function component ها متصل می کنند. Hook ها در class ها کار نمی کنند و به شما اجازه می دهند تا از ویژگی های React بدون نوشتن class استفاده کنید.

Hook ها backwards-compatible هستند ، به این معنا که هیچ تغییر (break change) را نگه نمی دارند. React چند Hook built-in مثل useState, useEffect, useReducer دارد . هم چنین می توانید Hook های سفارشی بسازید .

منابع :

<https://www.devaradise.com/react-functional-component-with-hooks>

[https://www.tutorialspoint.com/reactjs/reactjs\\_state.htm](https://www.tutorialspoint.com/reactjs/reactjs_state.htm)

<https://www.loginradius.com/blog/async/react-state-management/>

<https://www.dotnettricks.com/learn/react/functional-component-react>