React State Management \*

اپلیکیشن های rect با استفاده از کامپوننت ها ساخته شده اند. . *برای انتقال استیت و مقادیر، از props استفاده و مقادیر را از بالا به پایین منتقل می‌کنیم.* هنگامی که نیاز است تا تمام کامپوننت‌ها از یک استیت استفاده کنند کار دشواری می شود. اما زمانی که برنامه بزرگ‌تر می‌شود، پیچیدگی مدیریت state های به اشتراک گذاشته شده در بین کامپوننت ها دشوار می‌شود. مفهوم context مطرح شد که در واقع یکی از ویژگی های ریکت است که به ما این اجازه را می‌دهد که داده را بین کامپوننت‌ها تبادل و به اشتراک بگذاریم.

هنگامی که نیاز است یک استیت را همه جا استفاده کنیم از context استفاده می کنیم.

Function Component \*

استفاده از توابع از نظر سینتکسی خیلی راحت تر از کلاس ها هستند . با ایجاد hooks در ری اکت 16.8 این قابلیت پیش آمد که تمام life cycle ری اکت را در functional Components داشته باشید . useEffect یک هوک برای سنجش تغییرات کامپوننت در کلاس های تابعی هست که کاری مشابه componentDidUpdate, componentWillUnmount , componentDidmount را انجام میدهد .

Hook in React \*

هوک (Hook) تابع خاصی است که با استفاده از آن می‌توانیم به کامپوننت‌ها، استیت (State) و منطق اضافه کنیم. نام‌گذاری هوک‌ها با use آغاز می‌شود.

با استفاده از هوک، می‌توانید از ویژگی‌هایی که ری‌اکت دراختیارتان قرار داده استفاده کنید، مانند نگه‌داری مقادیر، انجام کار‌های خاص در زمان اولین اجرا، یا انجام عملی خاص در زمانی که یک متغیر تغییر کرد.

برخی از هوک‌هایی که ریکت در اختیار ما قرار داده به شرح زیر می‌باشند:

useState // useEffect // useRef

useState

این هوک برای نگه‌داشتن مقدارهای کامپوننت استفاده می‌شود.

هوک دو مقدار به ما می‌دهد. مقدار اول، خود استیت و مقدار دوم، تابع مربوطه که با دادن مقدار جدید به این تابع، استیت، این مقدار جدید را می‌گیرد.

const [state, setState] = useState(0);