React state management

برای ایجاد تغییرات در react نمی توانیم خود آن مقداررا مستقیما تغییر دهیم چرا که این تغییر باعث re-render به کاربر نمی شود. راه حل ما اضافه کردن useState است که یکی از hook های مهم در react می باشد.

Functional Components

دو نوع کامپوننت در React وجود داره Class Components و Functional Components. .Class components در واقع کلاس های ES6  هستند و Functional Components توابع هستند

### ۱- خوانایی فانکشنال کامپوننت ها بالاتر از کلاس کامپوننت هاست ( چون همون توابع ساده جاواسکرپیتی هستند )

### تست کردن فانکشنال کامپوننت ها ساده تر از کلاس کامپوننت هاست.

### Functional Components عملکرد بهتری دارند

اگر ما در function components از state ها استفاده نکنیم،کامپوننت ها راحت تر پیاده سازی می شوند و خیلی راحت در قسمت های دیگه پروژه و پروژه های دیگه قابل استفاده هستند.

Hook in react

تو تمرین هایی که انجام دادیم دیدیم  (this.setState) تنها راه تغییر state است اما در نسخه جدید قابلیتی به نام hook در ری اکت اضافه شده است که به ما اجازه می دهد state را از طریق کامپوننت های کاربردی (functional) نیز تغییر دهیم.

Hook ها توابعی ساده هستند که این امکان را به ما می دهند از قابلیت های state و lifecycle در react بدون استفاده از ساختار class در کامپوننت هایی که به صورت تابع هستند استفاده کنیم. Hook ها در داخل class ها غیر قابل استفاده می باشند. React تعدادی Hook همانند useState را به صورت پیش فرض تهیه کرده است که می توان از آن استفاده کرد.

useState یک آرایه با دو عنصر بر می گرداند که عنصر اول مقدار فعلی state و عنصر دوم یک تابع برای بروزرسانی مقدار state می باشد. این تابع شبیه this.setState می باشد با این تفاوت که state قبلی و جدید را با هم ادغام نمی کند. useState تنها یک پارامتر به عنوان مقدار اولیه state می گیرد.

از state hook می توانیم چندین بار در یک کامپوننت استفاده کنیم.

منابع :

reactjs.org

virgool.io