**مفهوم کامپوننت :**

کامپوننت ، در واقع بلاکی از ساختمان reactهست و یک پروژه ریکت شامل، کامپوننت های زیادی می باشد که یک کامپوننت، یک کلاس در جاوااسکریپت یا تابع است که به صورت اختیاری ورودی می پذیرد و props ها و یک المنت react برمی گرادندکه توضیح می دهد چگونه بخشی از User Interface (رابط کاربری ) باید ظاهر شود. در واقع کامپوننت ریکت به ما اجازه می دهد که interface را به قسمت های مجزا و قابل استفاده مجدد تقسیم نماییم و در مورد هر بخش به صورت جداگانه فکر نماییم. و در واقع کامپوننت ها داده های خام را به html تبدیل و یا ترجمه می نمایند.

منابع :

<https://medium.com/the-andela-way/understanding-react-components-37f841c1f3bb>

<https://reactjs.org/docs/components-and-props.html>

**مفهوم state :**

State یک نوع آبجکت جاوا اسکریپت هست که توسط react برای نشان دادن اطلاعات درباره وضعیت فعلی کامپوننت ارائه شده است. و در کامپوننت مدیریت می شود.(مانند متغیری های که در یک فانکشن تعریف می شوند) اما تفاوت بین این دو وجود دارد که متغیرهای نرمال داخل تابع با خروج از تابع ناپدید می شوند. اما state ها در ریکت حفظ می شوند. از آنجایی که نقش کامپوننت ها در واقع ترجمه داده های خام به html می باشد ما می توانیم در نظر بگیریم که این داده خام از از props و آبجکت state ساخته شده است . حتی می توانیم بگیم که props و آبجکت ها ورودی های داده برای فانکشن render هستند. بنابراینstate ها برخی از ویژگی ها یا خصوصیات را نشان می دهند که رفتار کامپوننت ها را کنترل می کند و برخی از ویژگی ها (یا خصوصیات) را نشان می دهد که رفتار جزء را کنترل می کند.

منابع :

https://medium.com/edonec/state-in-react-an-overview-a182675cee2c#:~:text=State%20is%20a%20plain%20JavaScript%20object%20used%20by%20React%20to,variable%20declared%20in%20a%20function).

since the role of the component is to translate raw data into HTML, we can consider that this raw data is made out of props and state objects. We could even say that props and state are the input data for the render() function. So the State represents some property (or properties) that controls the behavior of the component.

State is a plain JavaScript object used by React to represent an information about the component’s current situation.  
It’s managed in the component (just like any variable declared in a function). The difference is while a “normal” variable “disappears” when their function exits, the state variables are preserved by React.