

سوال (۱)

Difference between npm and npx

npm - Package manager یک ابزاری برای نصب پکیج ها
npx - Package runner یک ابزاری برای اجرا کردن پکیج ها

برای مثال اگر از `npm create-react-app my-app` استفاده کنیم، `local package create-react-app` از روی سیستم اجرا میشود یعنی برای اجرا شدن این پکیج محلی، قبل اینکه این پروژه ری اکت رو با `npm` نصب کنیم، باید قبلاً بصورت `globally` روی سیستم با دستور `npm install -g create-react-app` نصب کنیم و بعد بتونیم پروژه های ری اکت رو روی سیستم نصب کنیم. و برای اجرای یک پکیج چه `local` چه `global` با `npm`، باید اون پکیج رو تو فایل `package.json` مشخص کنی. ولی اگه از `npx create-react-app my-app` استفاده کنیم و قبلاً `create-react-app` رو بصورت `globally` روی سیستم نصب نکرده باشیم، با این دستور `create-react-app` روی برنامه دانلود میشه ولی بصورت `globally` نصب نمیشه. این برای زمان هایی ست که نمیخواهیم همه ی پکیج ها رو روی سیستم نصب کنیم و فقط پکیج های مورد نیازمون رو تو برنامه داشته باشیم و اجرا کنیم خیلی خوبه. برای اجرای پکیج های `local` و پکیج هایی که قبلاً نصب نکردیم خیلی راحت میتونیم با دستور `npx` اجرا کنیم. در واقع با دستور `npx` بدون نیاز به نصب میتون مستقیماً از آن اجرا کرد.

منبع: <https://virgool.io/@happycodee/comment/nitwv8bj4dajqqr>

منبع ۲: <https://roocket.ir/articles/yes-its-npx-not-npm-the-difference-explained>

سوال (۲)

مفهوم component :

Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and return HTML.

Components come in two types, Class components and Function components.

کامپوننت ها تکه کد هایی هستند که میتوان از آن ها هر جایی در پروژه استفاده کرد و در اصطلاح `reusable` هستند.

در ری اکت دو نوع کامپوننت میتوان نوشت ، یکی به شکل تابع و دیگری به شکل کلاس (class components , functional components).

منابع:

https://www.w3schools.com/react/react_components.asp#:~:text=Components%20are%20independent%20and%20reusable,will%20concentrate%20on%20Function%20components.

<https://virgool.io/iran-react-community/%D%A9%D8%A7%D9%85%D9%BE%D9%88%D9%86%D9%86%D8%AA-%D9%87%D8%A7-%D8%AF%D8%B1-%D8%B1%DB%8C-%D8%A7%DA%A9%D8%AA-gxatvwucn1em>

مفهوم state :

React components has a built-in state object.

The state object is where you store property values that belongs to the component.

When the state object changes, the component re-renders.

در استیت زمانی که تغییری رخ میدهد render انجام میشه ولی زمانی که از کامپوننت استفاده میکنیم این اتفاق نمیوفته مطابق مثال
Todoist اولیه و زمانی که todoitem رو اضافه کردیم مگر اینکه استیت کامپوننت هم دچار تغییر بشه.
استیت یک ابجکت برای نگه داری دیتاهای مربوط به یک کامپوننت.

منابع: https://www.w3schools.com/react/react_state.asp

سوال ۳)

React components automatically re-render whenever there is a change in their state or props.

A simple update of the state, from anywhere in the code, causes all the User Interface (UI) elements to be re-rendered automatically

Re-render can be caused due to any of the three reasons listed:

Update in State

Update in prop

Re-rendering of the parent component

Update in state: The state change can be from a prop or setState change to update a variable(say). The component gets the updated state and React re-renders the component to reflect the change on the app.

Update in prop: Likewise the change in prop leads to state change and state change leads to re-rendering of the component by React.

Re-rendering of parent component: Whenever the components render function is called, all its subsequent child components will re-render, regardless of whether their props have changed or not.

منبع:

<https://www.geeksforgeeks.org/re-rendering-components-in-reactjs/#:~:text=React%20components%20automatically%20re%2Drender,in%20their%20state%20or%20props.&text=A%20simple%20update%20of%20the,to%20be%20re%2Drendered%20automatically>

