

به نام خدا

تمرین سوم hw۱۴

۱- تفاوت npm و npx چیست ؟

npm	npx
npm مخفف Node Package Manager می باشد.	npm مخفف Node Package Execute می باشد.
npm با نصب Node.js در سیستم نصب می شود.	بعد از نصب npm از ورژن ۲,۵,۰ بطور خودکار npx نصب می شود.
npm یک repository آنلاین برای انتشار پروژه های این سورس Node.js است. هم چنین می توانیم با استفاده از آن package ها را بصورت global و local نصب کنیم .	npx یک اجر کننده بسته npm است که می تواند هر package را که می خواهید از رجیستری npm بدون نصب آن package اجرا کند.
اگر می خواهید بسته را از طریق npm اجرا کنید، باید آن بسته را در package.json در scripts خود مشخص کرده و آن را به صورت محلی نصب کنید. و برای اجرای آن از دستور npm run your--name package استفاده کنید . یا می توانید مسیر بسته را در ترمینال تایپ کنید. ./node_modules/.bin/your-package-name	اگر می خواهید یک بسته نصب شده محلی را اجرا کنید، از دستور npx your-package-name استفاده کنید و اگر بسته ای نصب شده باشد، npx بایزری های بسته را (به صورت محلی یا سراسری) جستجو می کند و سپس بسته را اجرا می کند. اگر بسته قبلاً نصب نشده باشد، npx بسته را در سیستم شما نصب نخواهد کرد. در عوض، یک کش موقت ایجاد می کند که بایزری های بسته را نگه می دارد. پس از اتمام اجرا، npx بایزری های کش نصب شده را از سیستم حذف می کند.
برای ساخت react app با npm از دو دستور create-react-app myApp و create-react-app استفاده می کنیم .	برای ساخت react app با npx از دستور npx create-react-app myApp استفاده می کنیم .

۲- مفهوم component و state در react چیست ؟

: Component

پیش از این، توسعه دهندگان بیش از هزاران خط کد را برای توسعه یک برنامه کاربردی تک صفحه ای می نوشتند. این برنامه ها از ساختار سنتی DOM پیروی می کنند و ایجاد تغییرات در آنها یک کار بسیار چالش برانگیز بود. اگر اشتباهی پیدا شود، به صورت دستی کل برنامه را جستجو می کند و بر اساس آن به روز می شود. رویکرد مبتنی بر کامپوننت برای غلبه بر این مشکل معرفی شد. در این رویکرد، کل برنامه به یک گروه منطقی کوچک از کدها تقسیم می شود که به کامپوننت ها معروف هستند. کامپوننت ها بلوک های سازنده هر برنامه React هستند و یک برنامه معمولی React بسیاری از آنها را دارد. به بیان ساده، یک کامپوننت یک کلاس یا تابع جاوا اسکریپت است که به صورت اختیاری ورودی ها می پذیرد، یعنی ویژگی ها (props). و React element را برمی گرداند (تبدیل JSX نوشته شده به React element توسط babel handles) که توضیح می دهد چگونه بخشی از UI (رابط کاربر) باید ظاهر شود. کامپوننت ها شامل دو نوع Class components و Function components می باشند. کامپوننت ها به طور مستقل از یکدیگر کار می کنند و همه در یک component والد ادغام می شوند، که رابط کاربری نهایی برنامه خواهد بود.

: State

stat یک ساختار و شی است که قابل به روز رسانی است که ویژگی ها، داده ها یا اطلاعات مربوط به کامپوننت را نگه داری می کند. state در یک کامپوننت می تواند در طول زمان و عمر کامپوننت تغییر کند. تغییر state در طول زمان می تواند به عنوان پاسخی به اقدام کاربر یا رویداد سیستم اتفاق بیفتد. state قلب کامپوننت ری اکت است که رفتار کامپوننت و نحوه render شدن آن را تعیین می کند. همچنین مسئول ایجاد یک کامپوننت پویا و تعاملی می باشد. یک state باید تا حد امکان ساده نگه داشته شود. می توان state را با استفاده از متد setState() تنظیم کرد و با فراخوانی متد setState() به روز رسانی UI را راه اندازی کرد. یک state نشان دهنده وضعیت یا اطلاعات local کامپوننت است و فقط در داخل کامپوننت یا مستقیماً توسط کامپوننت قابل دسترسی یا تغییر است.

۳-چه مواردی باعث re-render شدن کامپوننت می شوند ؟

یک به روز رسانی ساده state، از هر نقطه ای از کد، باعث می شود که تمام عناصر رابط کاربری (UI) دوباره به صورت خودکار render شوند. با این حال، ممکن است مواردی وجود داشته باشد که متد render() به برخی داده های دیگر بستگی دارد. render مجدد می تواند به هر یک از سه دلیل ذکر شده ایجاد شود:

۱. به روز رسانی در state : تغییر state می تواند از یک تغییر prop یا setState برای به روز رسانی یک متغیر باشد. کامپوننت state به روز شده را دریافت می کند و React کامپوننت را مجدداً render می کند تا تغییر را در برنامه منعکس کند.

۲. به روز رسانی در prop : به همین ترتیب تغییر در prop منجر به تغییر state و تغییر state منجر به render مجدد کامپوننت توسط React می شود.

۳. re-render شدن مجدد کامپوننت والد : هر زمان که تابع render کامپوننت ها فراخوانی شود، تمام کامپوننت های فرزند بعدی آن، صرف نظر از اینکه آیا props آنها تغییر کرده یا نه، دوباره render می شوند.

منابع :

<https://www.geeksforgeeks.org/what-are-the-differences-between-npm-and-term%۲۰long%۲۰the%۲۰for%۲۰that.pollution%۲۰tool%۲۰a%۲۰is%۲۰npx/#:~:text=Npm>

[/https://www.freecodecamp.org/news/npm-vs-npx-whats-the-difference](https://www.freecodecamp.org/news/npm-vs-npx-whats-the-difference)

<https://www.section.io/engineering-education/npm-vs-npx/#what-is-npx>

<https://www.javatpoint.com/react-components>

<https://medium.com/the-andela-way/understanding-react-components-bb۳f۱c۸۴۱۴۳۷>

<https://www.javatpoint.com/react-state>

<https://www.geeksforgeeks.org/re-rendering-components-in-re-%۲۰automatically%۲۰components%۲۰reactjs/#:~:text=React>

[be%۲۰the,to%۲۰of%۲۰update%۲۰simple%۲۰props.&text=A%۲۰or%۲۰state%۲۰their%۲۰render,in.automatically%۲۰re-rendered%۲۰](https://www.geeksforgeeks.org/re-rendering-components-in-re-%۲۰automatically%۲۰components%۲۰reactjs/#:~:text=React)