Fariba-Ghorbani

سوال اول: تفاوت npm و npx

Npm یا node package manager با نصب node.js به عنوان پکیج منیجر دیفالت نصب می‌شود و وظیفه مدیریت تمام پکیج‌ها و ماژول‌ها را در node به عهده دارد. پکیج‌های نصب شده را به کمک npm از دو روش می‌توان اجرا کرد.

- استفاده از آدرس فایل اجرایی آن در فولدر .bin مربوط به node_modules در محیط command-line
- استفاده از scripts در package.json و کامند npm run

در هر دو روش ذکر شده که npm برای اجرای پکیج مورد استفاده قرار می‌دهد، مشخصات این پکیج‌ها باید در package.json قرار بگیرد و به صورت محلی نصب شوند. به عبارتی، فایل اجرایی آن‌ها باید در سیستم موجود باشد.

npx که مخفف Node Package Execute است، ابزاری است که به همراه npm می‌آید و نصب می‌شود (از نسخه 5.2.0 به بالا). وظیفه آن اجرای پکیج‌ها از رجیستری npm است یعنی الزامی به نصب پکیج وجود ندارد و پکیج‌هایی که توسط npm نصب نشده اند نیز با npx می‌توانند اجرا شوند.

پکیج‌ها در npm به صورت گلوبال نصب می‌شوند و در طولانی مدت باید به پاک سازی پکیج‌های اضافی پرداخت اما پکیج‌ها در npx این گونه نصب نمی‌شوند و دغدغه ای از بابت وجود پکیج‌های ناخواسته وجود ندارد.

https://www.geeksforgeeks.org/what-are-the-differences-between-npm-and-npx/#:~:text=Npm%20is%20a%20tool%20that,pollution%20for%20the%20long%20term.

سوال دوم: مفهوم component و state در React؟

Component ها در حقیقت واحدهای ساختاری هر برنامه توسعه داده شده توسط React است. به عبارتی دیگر Component ها یک آبجکت یا فانکشن در جاوااسکریپت هستند که یک React element برمی‌گردانند. این React element ها رابط کاربری سکشن‌های مختلف برنامه را توصیف می‌کنند. یک component واحد می‌تواند چندین بار در طول برنامه با ویژگی‌های مختلف مورد استفاده قرار بگیرد تا اطلاعات متفاوتی را از خود نمایش دهد.

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

https://www.youtube.com/watch?v=Y2hgEGPzTZY

state یک آبجکت رزرو شده توسط React است که اطلاعاتی در مورد وضعیت حال component نظیر خود به ما میدهد. چون نقش component این است که داده‌های خام و جاوااسکریپتی را به HTML تبدیل کند، می‌توان در نظر گرفت که این داده‌های خام از state و props گرفته می‌شود و به عبارتی، این دو به عنوان ورودی برای component رفتار می‌کنند. پس تغییر state یکی از راه‌های تاثیرگذاشتن بر render و خروجی component های React است و با هر تغییر آن، component نظیر آن نیز re-render می‌شود.

https://www.youtube.com/watch?v=4ORZ1GmjaMc

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.(

سوال سوم: چه مواردی باعث re-render شدن component می‌شود؟

React یک کتابخانه open source است که یک تجربه کاربری سریع به واسطه تغییر حداقلی و لازم بخش‌هایی از UI که باید دست‌خودش تغییر شوند، به ما می‌دهد. Render شدن component ها خارج از اختیار کاربر است و بخشی از چرخه زندگی component ها محسوب می‌شود و در استیج‌های مختلفی از برنامه فراخوانی می‌شود. render ای که موخر بر اولین render برای آپدیت شدن UI صورت می‌گیرد، re-render نامیده می‌شود.

به دلایل زیر re-rendering رخ می‌دهد:

- آپدیت شدن ویژگی‌های state: تغییراتی که در state صورت می‌گیرد، چه از طریق props و چه از طریق setState()، باعث می‌شود که React آن component را re-render کرده تا تغییرات صورت گرفته را بازتاب دهد.
- آپدیت شدن ویژگی‌های props: به همین ترتیب آپدیت شدن ورودی‌های props در یک component نیز منجر به وقوع re-rendering می‌شود.
- Re-render شدن component های parent: هر زمانی که تابع render یک component فراخوانی می‌شود، تمام component های زیرمجموعه آن (child) فارغ از این که props آن ها تغییر داده شده یا نه، re-render می‌شود.
- تغییر متد shouldComponentUpdate(): هر موقع که React تصمیم به render کردن component می‌گیرد، این متد را چک می کند و در صورتی که true برگرداند، کار خود را انجام میدهد. پس با overwrite کردن این متد می‌توان نحوه رفتار React را برای آپدیت کردن یا نکردن یک component کنترل کرد.
- یک روش دیگر استفاده از props: key است که در این صورت React آن را watch می کند و در صورتی که تغییری در key صورت بگیرد، کاری می کند که React آن component را unmount کند و به عبارتی آن component مجبور می شود دوباره lifecycle را طی کند و componentDidMount دوباره trigger می شود و re-render صورت می گیرد.

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.

https://lucybain.com/blog/2017/react-js-when-to-rerender/

https://linguinecode.com/post/4-methods-to-re-render-react-component

https://linguinecode.com/post/understanding-react-componentdidmount