Npm vs Npx

NPM	NPX
مخفف Node package manager	مخفف Node package execute
اگر بخواهید پکیجی را از طریق npm اجرا کنید باید آن را در فایل package.json مشخص کنید و آن را به طور محلی نصب کنید	یک پکیج میتواند بدون اینکه نصب شود نیز اجرا شود. Npx یک اجراکننده پکیج npm است. بنابراین اگر هر پکیجی نصب نشده باشد به طور اتوماتیک نصب می‌شود.
Npm ابزاری است که برای نصب پکیج استفاده می‌شود	Npx ابزاری است که برای اجرای پکیج استفاده می‌شود.
پکیج‌هایی که از NPM استفاده می‌کنند، به صورت سراسری نصب می‌شوند و امکان تداخل در دراز مدت وجود دارد.	پکیج‌هایی که از NPX استفاده میکنند به صورت سراسری نصب نمی‌شوند و امکان تداخل در دراز مدت وجود ندارد.
	npxشما را محدود به پکیج‌های انتشار یافته روی رجیستری npm نمی‌کند.
	با استفاده از @ می‌توان اقدام به تعیین نسخه و ترکیب کردن آن‌ها با پکیج npm در node کرد.
	npx .به شما کمک می‌کند تا از ورژن‌سازی ، مشکلات dependency ها و نصب پکیج‌های غیرضروری، که فقط می‌خواهیم آن‌ها را امتحان کنیم، جلوگیری کنیم.

منابع:

https://www.geeksforgeeks.org/what-are-the-differences-between-npm-and-npx/

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

https://blog.faradars.org/node-js-tutorial-part-۵/

https://virgool.io/@happyycodee/comment/nitwv۸bj٤dajqqr

مفهوم component و state در react

Components

کامپونت‌ها کدهای مستقل و قابل استفاده مجدد هستند. آن‌ها همان هدف توابع جاوا اسکریپت را انجام می‌دهند با این تفاوت که به صورت ایزوله کار می‌کنند و html برمی‌گردانند. Componentsبه شما اجازه می دهد UI را به صورت مستقل تقسیم کنید، درمورد هر Components به قطعات قابل استفاده مجدد و هرقطعه جدا شده فکر کنید. از نظر مفهومی، componentsمانند functions ها در JavaScript هستند.

ری اکت یک لایبرری کامپوننت بیس است. به این معنی که هر قسمت از وب سایت را به صورت کامپوننت های جداگانه در نظر می گیریم و سپس آن ها را در کنار همدیگر قرار می دهیم و با در کنار هم قرار دادن آن ها وب سایت مورد نظرمان ساخته می شود.

به عنوان مثال اگر بخواهیم کامپوننت های صفحه اصلی یک فروشگاه اینترنتی را در نظر بگیریم، می توان به Header، NavBar، Products، PopularProducts، Footer و ... اشاره کرد.

زمانی که شما پروژه خودتان را به کمک ری اکت به صورت کامپوننت بیس توسعه می دهید، اگر قسمتی از وب سایت شما مشکلی داشته باشد در مدت زمان کمی می توانید متوجه شوید که مشکل موجود در کدام یک از کامپوننت ها رخ داده است و به راحتی می توانید آن را فیکس کنید.

مزایای بعدی استفاده از کامپوننت این است که شما می توانید پروژه خودتان را با کنترل بالاتر توسعه دهید و مدیریت بیشتری روی پروژه مورد نظر داشته باشید.

به عنوان مثال فرض کنید بعد از مدت ها از توسعه پروژه قصد دارید فیچری را به پروژه خودتان اضافه کنید. در این صورت به راحتی و در مدت زمان کم می توانید فیچر مورد نظرتان را به پروژه مورد نظر اضافه کنید.

مزایای بعدی استفاده از کامپوننت قابلیت استفاده مجدد و در نتیجه کاهش حجم سورس کد پروژه است. شما می توانید یک قسمت از وب سایت را یک کامپوننت در نظر گرفته و آن را چندین بار استفاده کنید.

اگر بخواهیم یک نتیجه گیری و جمع بندی کلی داشته باشیم، می توان گفت استفاده از کامپوننت مزایایی مثل:

۱. توسعه و مدیریت بهتر
۲. دیباگینگ راحت تر
۳. پرفورمنس بهتر
٤. تست نویسی راحت تر و...

State

Stateیک شیء جاوا اسکریپت است که داده‌های دینامیک کامپوننت را نگه‌داری می‌کند و آن را قادر می‌سازد که تغییرات بین رندرها را ردگیری کند. از آنجا که State دینامیک است، تنها به منظور ایجاد تعامل‌پذیری استفاده می‌شود و از این رو در مورد پروژه‌های استاتیک React کاربردی ندارد.

کامپوننت‌ها به صورت کلاس‌هایی تعریف می‌شوند که ویژگی‌های اضافی دارند State .محلی دقیقاً به معنای یک ویژگی است که تنها در کلاس‌ها وجود دارد State .صرفاً درون کلاس می‌تواند مورد استفاده قرار گیرد و معمولاً تنها جایی است که می‌توان this.state را به عنوان سازنده مطرح کرد.

Stateدرون کامپوننت مدیریت می‌شود، همان طور که متغیرها درون یک تابع اعلان می‌شوند State .در کامپوننت React در واقع حالت محلی خودش است، یعنی حالت نمی‌تواند خارج از کامپوننت مورد دسترسی یا تغییر قرار گیرد و تنها درون آن کاربرد دارد. این وضعیت شبیه تابعی است که حیطه محلی خود را دارد.

منابع:

https://www.w۳schools.com/react/react_components.asp

https://virgool.io/@learnreact/components-and-props-in-react-js-wmstoeae۲۳۱۷

https://sabzlearn.ir/what-is-component/#kampwnnt_dr_ry_akt

چه مواردی باعث re-render شدن کامپوننت میشود؟

به طور کلی re-render ممکن است به دلایل زیر اجرا شود:

۱- به روزرسانی در state
۲- به روزرسانی در prop
۳- رندر مجدد کامپوننت والد

بیایید با جزئیات ببینیم که چرا کامپونت‌ها دوباره رندر می‌شوند و چگونه از رندر مجدد ناخواسته برای بهینه‌سازی اجزای برنامه جلوگیری کنیم.

۱- به روزرسانی در state : تغییر state میتواند به خاطر تغییر در prop و یا setState برای به روز رسانی یک متغیر باشد. کامپونت به‌روز رسانی را دریافت می‌کند و React با رندر مجدد کامپونت، تغییر را در برنامه منعکس می‌کند.
۲- به روزرسانی در prop : به همین ترتیب ، تغییر در prop باعث تغییر در state و تغییر state باعث رندر دوباره کامپونت توسط React می‌شود.
۳- رندر مجدد کامپونت والد : هرزمان که تابع رندر کامپونت‌ها اجرا شود، تمامی کامپونت های فرزند آن re-render خواهد شد. صرف نظر از اینکه آیا prop آنها تغییر کرده است یا خیر.

منابع:

https://www.geeksforgeeks.org/re-rendering-components-in-reactjs/