جواب سوال ۱:

NPM = Manages packages but doesn't make life easy executing any.

NPX = A tool for executing Node packages.

npm :

یک پکیج منیجیر برای Node.js

هدفش خودکار کردن روند کارکرد وابستگی‌های یک پروژه و مدیریت پکیج‌های آن است.

به عبارت دیگر شما می‌توانید تمامی وابستگی‌ها و یا پکیج‌های مربوط به پروژه‌تان را در فایل package.json مشخص کنید

هرگاه شخص دیگری خواست پروژه شما را به اجرا درآورد، می‌تواند تنها با نوشتن دستور npm install یا npm i به نصب تمامی آن وابستگی‌ها که برای ران کردن پروژه موردنیاز اند، بپردازد.

همچتین نشان می‌دهد که پروژه‌ی شما از کدام نسخه‌ی فلان کتاب‌خانه بهره برده است و بدین ترتیب شما با متناسب‌ترین نسخه‌ی ممکن از پکیج‌های استفاده شده، پروژه را به اجرا در میاورید تا از مشکلات احتمالی و تداخلات زیادی جلوگیری شود .

npx :

ابزاریست برای به اجرا درآوردن پکیج‌های Node و از نسخه‌ی npm ۵.۲ به بعد در اختیار کابران این پکیج منیجر قرار گرفته است.

به طور پیش‌فرض ابتدا بررسی می‌کند که آیا پکیجی که می‌خواهد اجرا شود در مسیر پروژه‌تان موجود است یا خیر.۱

.۲اگر در مسیر پروژه وجود داشت، آن را اجرا می‌کند

در غیر این‌صورت، یعنی پکیج نصب نشده است؛ پس۳ npx به نصب آخرین نسخه از آن وابستگی می‌پردازد و سپس آن را به اجرا در می‌آورد.

در انتها لازم به ذکر است که ویژگی منحصر به فرد npx که آن را برای توسعه‌دهندگان جذاب‌تر کرده، این است که به شما اجازه می‌دهد که یک پکیج را بدون نصب اولیه و گلوبال، به اجرا درآورید.

NPM is a package manager, you can install node.js packages using NPM

NPX is a tool to execute node.js packages.

Simple Definition:

npm - Javascript package manager

npx - Execute npm package binaries

جواب سوال ۲ :

state :

حالت نمونه ای از React Component Class است

که می تواند به عنوان یک شی از مجموعه ای از ویژگی های قابل مشاهده تعریف شود که رفتار جزء را کنترل می کند. به عبارت دیگر، وضعیت یک جزء، یک شی است که اطلاعاتی را در خود نگه می‌دارد که ممکن است در طول عمر کامپوننت تغییر کند.

کامپوننت های React دارای یک آبجکت حالت داخلی هستند.

آبجکت state جایی است که مقادیر ویژگی متعلق به کامپوننت را در آن ذخیره می کنید.

هنگامی که شیء حالت تغییر می کند، کامپوننت دوباره ارائه می شود.

برای تغییر یک مقدار در شی state، از متد ()this.setState استفاده کنید.

هنگامی که یک مقدار در شیء حالت تغییر می کند، مولفه دوباره رندر می شود، به این معنی که خروجی مطابق با مقدار(های) جدید تغییر می کند.

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.stateرا به عنوان سازنده مطرح کرد.

Component :

کامپوننت ها بیت های کد مستقل و قابل استفاده مجدد هستند. آنها همان هدف توابع جاوا اسکریپت را انجام می دهند، اما به صورت مجزا کار می کنند و HTML را برمی گردانند.

کامپوننت ها در دو نوع Class Components و Function Component ها هستند که در این آموزش بر روی Function Component ها تمرکز می کنیم.

در پایگاه‌های کد React قدیمی‌تر، ممکن است اجزای کلاس را بیابید که عمدتاً استفاده می‌شوند. در حال حاضر پیشنهاد می شود از کامپوننت های Function به همراه Hooks استفاده کنید که در ۱۶.۸ React اضافه شده اند.

componentها به دلیل اینکه به React می‌گویند باید چه مواردی را به کاربر نمایش دهد، اهمیت زیادی دارند. مهم‌تر از همه اینکه اگر در این اطلاعات تغییری ایجاد شود، Reactهم به شکلی بهینه عناصر را به‌روزرسانی و بازنویسی خواهد کرد.

هر عنصر پارامترهایی به نام) props ویژگی، کوتاه شده‌ی کلمه (properties را می‌پذیرد و از طریق متد render سلسله‌مراتب مشاهدات را به صفحه برمی‌گرداند.

جواب سوال ۳ :

همانطور که می دانیم React JS یک کتابخانه جاوا اسکریپت منبع باز است که برای ساخت رابط های کاربری به طور خاص برای برنامه های تک صفحه ای استفاده می شود. همچنین به دلیل ارائه تجربه کاربری سریع تنها با به‌روزرسانی بخش‌هایی از رابط کاربری که تغییر کرده‌اند، شناخته شده است. کامپوننت‌های رندر در دست کاربر نیست، بخشی از چرخه حیات React Component است و توسط React در مراحل مختلف برنامه فراخوانی می‌شود، معمولاً زمانی که React Component برای اولین بار نمونه‌سازی می‌شود. رندر دوم یا بعدی برای به روز رسانی حالت به عنوان re-rendering. نامیده می شود.

کامپوننت‌های React هر زمان که تغییری در

Stateیا propsآن‌ها ایجاد شود، re-renderingمی‌شوند.

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

بیایید نگاهی عمیق‌تر به سه دلیل بالا، داشته باشیم:

Update in state :

تغییر وضعیت می تواند از یک تغییر prop یا setState برای به روز رسانی یک متغیر (مثلا) باشد. کامپوننت حالت به روز شده را دریافت می کند و React کامپوننت را مجدداً ارائه می کند تا تغییر را در برنامه منعکس کند.

Update in prop:

به همین ترتیب تغییر در prop منجر به تغییر حالت و تغییر حالت منجر به رندر مجدد کامپوننت توسط React می شود.

Re-rendering of parent component:

هر زمان که تابع رندر کامپوننت ها فراخوانی شود، تمام مولفه های فرزند بعدی آن، صرف نظر از اینکه پروپ های آنها تغییر کرده یا نه، دوباره رندر می شوند.