

جواب سوال ۱:

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

NPX = A tool for executing Node packages.

npm :

یک پکیج منیجر 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.

همچنین 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 می‌آورید تا از مشکلات احتمالی و تداخلات زیادی جلوگیری شود .

npx :

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

به طور پیش فرض ابتدا بررسی می کند که آیا پکیجی که می خواهد اجرا شود در مسیر پروژه تان موجود است یا خیر.

اگر در مسیر پروژه وجود داشت، آن را اجرا می کند

در غیر این صورت، یعنی پکیج نصب نشده است؛ پس npx به نصب آخرین نسخه از آن وابستگی می پردازد و سپس آن را به اجرا در می آورد.

در انتها لازم به ذکر است که ویژگی منحصر به فرد npx که آن را برای توسعه دهندگان جذاب تر کرده، این است که به شما اجازه می دهد که یک پکیج را بدون نصب اولیه و گلوبال، به اجرا در آورید.

**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 می دارد 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 شیء 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.

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 پایگاه های کد React قدیمی تر، ممکن است اجزای کلاس را بیابید که عمدتاً استفاده می شوند. در حال حاضر پیشنهاد می شود از کامپوننت های Function به همراه Hooks استفاده کنید که در React ۱۶.۸ 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، React هم 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 می‌دانیم React JS یک کتابخانه جاوا اسکریپت منبع باز است که برای ساخت رابط‌های کاربری به طور خاص برای برنامه‌های تک صفحه‌ای استفاده می‌شود. 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 به‌روزرسانی حالت به عنوان **re-rendering** نامیده می‌شود.

کامپوننت‌های React هر زمان که تغییری در State یا props آنها ایجاد شود، re-rendering می‌شوند.

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

بیا باید نگاهی عمیق‌تر به سه دلیل بالا، داشته باشیم:

Update in state :

تغییر وضعیت می‌تواند از یک تغییر prop یا setState برای به روز رسانی یک متغیر (مثلاً) باشد. کامپوننت حالت به روز شده را دریافت می‌کند و React کامپوننت را مجدداً ارائه می‌کند 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 می‌شود.

Re-rendering of parent component:

هر زمان که تابع رندر کامپوننت‌ها فراخوانی شود، تمام مولفه‌های فرزند بعدی آن، صرف نظر از اینکه پروپ‌های آنها تغییر کرده یا نه، دوباره رندر می‌شوند.

منابع :

<https://stackoverflow.com/questions/۴۷۹۷۰۷۷۸/how-to-remove-the-default-cra-react-favicon#:~:text=Just%۲۰Navigate%۲۰to%۲۰public%۲۰Folder,by%۲۰renaming%۲۰it%۲۰as%۲۰favicon>  
[.on](#)

<https://stackoverflow.com/questions/۵۰۶۰۵۲۱۹/difference-between-npx-and-npm>

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

<https://porsan.ir/question/q-۲۰۸۴>

<https://blog.faradars.org/what-is-state-in-react>

<https://www.geeksforgeeks.org/reactjs-state-in-react/#:~:text=The%20state%20is%20an%20instance,the%20lifetime%20of%20the%20component>

[https://www.w3schools.com/react/react\\_state.asp](https://www.w3schools.com/react/react_state.asp)

[/https://mag.hostiran.net/what-is-react](https://mag.hostiran.net/what-is-react)

[https://www.w3schools.com/react/react\\_components.asp](https://www.w3schools.com/react/react_components.asp)

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