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

Npx => Node Package eXecute ابزاری برای اجرا کردن پکیج ها

Npm => Node Package Manager ابزاری برای نصب پکیج ها

وقتی یک پکیج را با npm نصب میکنیم . npm فایل های پکیج را در node\_module نصب میکند و اگر این پکیج فایلهای binary (فایل‌های اجرایی همان فایل‌های binary یا executable ها هستند) داشته باشد آنها را کامپایل میکند . و در دایرکتوری node\_modules نصب میکند . اگر پکیجی بصورت لوکال نصب شده باشد و بخواهیم آن را اجرا کنیم سه روش داریم :  
اول اینکه به صورت دستی آن را از مسیر bin اجرا کنیم .   
روش دوم که در قسمت script فایل package.json آن را بنویسیم و با npm پکیج را اجرا کنیم .

>> npm run packagename

روش سوم استفاده از npx است .

>> npx packagename

npx یک package runner است که از نسخه 5.2.0 این دستور باینری به همراه npm نصب میشود . در واقع خودش یک پکیج npm است و اگر بخواهیم از طریق package manager های دیگر استفاده کنیم میتوانیم آن را جداگانه نصب کنیم .

>> npm install –g npx

npx به طور پیش‌فرض ابتدا بررسی می‌کند که آیا پکیجی که می‌خواهد اجرا شود در مسیر پروژه‌ موجود است یا خیر.اگر در مسیر پروژه وجود داشت، آن را اجرا می‌کند.

در غیر این‌صورت، یعنی پکیج نصب نشده است؛ پس npx به نصب آخرین نسخه از آن وابستگی می‌پردازد (از طریق مخزن nodejs‌) و سپس آن را به اجرا در می‌آورد. این رفتار معمول npx است .  
  
npx کار با باینری ها را ساده تر میکند و دستوری که به آن پاس میدهیم به همراه همه ملزومات اجرای آن برایمان انجام میدهد .  
گاهی برای ایجاد پروژه به ابزارهایی نیاز داریم که خیلی کم استفاده میشود و نمیخواهیم آن ابزار در پوشه گلوبال سیستم عامل نصب شود وفضا اشغال کند . این هم کاربرد دیگری از npx است که ابزار را بصورت موقت نصب و بعد از بین میرود .

https://www.w3schools.com/react/react\_components.asp

کامپوننت ها شبیه فانکشن هایی هستند که اچ تی ام ال المنت برمیگردانند . کدهایی کوچک مستقل و قابل استفاده مجدد که به دو صورت استفاده میشود کلاس کامپوننت و فانکشن کامپوننت .

https://www.w3schools.com/react/react\_state.asp

استیت ها آبجکتهایی هستند که مقادیر پراپرتی مربوط به کامپوننت را نگهداری میکنند و با هربار تغییر در آن کامپوننت مجدد رندر میشود . استیت ها در کانستراکتور مقداردهی اولیه میشوند .

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

به سه دلیل کامپوننت ریرندر میشود :

1. تغییر در state : تغییرات استیت از یک setState یا تغییر props که باعث تغییر استیت میشود باعث ریرندر شدن کامپوننت میشود .
2. تغییر در props : تغییر در پراپسی که منجر به تغییر استیت میشود . باعث ریرندر شدن کامپوننت میشود .
3. آپدیت شدن کامپوننت والد : باعث میشود کامپوننت فرزند هم ریرندر شود .