1. تفاوت npx وnpm:

یک ابزاری برای نصب پکیج ها npm - Package manager

به عبارتی ما می‌توانید تمامی وابستگی‌ها و یا پکیج‌های مربوط به پروژه‌ را در فایل package.json مشخص کنیم و هرگاه شخص دیگری خواست پروژه شما را به اجرا درآورد، می‌تواند تنها با نوشتن npm installبه نصب تمامی آن وابستگی‌ها که برای ران کردن پروژه موردنیاز اند، بپردازد. همچنین نشان میده که پروژه‌ی از کدام نسخه‌ی فلان کتاب‌خانه است.

مسیر پکیج‌های نصب شده توسط npm :

* پکیج‌های locally‌ در دایرکتوری /node\_modules/.bin/. ایجاد می‌شوند.
* پکیج‌های globally در دایرکتوری bin/  ایجاد می‌شوند ـ (به طور‌مثال در لینوکس در مسیر /usr/local/bin  و در ویندوز در مسیر %AppData%/npm قرار دارند.)

یک ابزاری برای اجرا کردن پکیج ها npx - Package runner

به طور پیش‌فرض ابتدا بررسی می‌کند که آیا پکیجی که می‌خواهد اجرا شود در مسیر پروژه‌تان 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 و سپس آن را به اجرا در می‌آورد.

منابع:

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

<https://medium.com/codex/are-npm-and-npx-the-same-567104f13bf0#:~:text=NPM%20%E2%80%94%20Manages%20packages%20and%20it,we%20just%20want%20to%20test>.

1. react در state و component مفهوم

: State

در واقع یک شیء است که property های متخلفی را در خود نگه می دارد. شما می توانید هر نوع داده ای (آرایه، رشته و...) را در آن قرار دهید

خصوصیت state یک خصوصیت خاص است؛ از این جهت که اگر تغییری در آن ایجاد شود react تمام DOM را دوباره render می کند

react متوجه شد که state تغییر کرده است بنابراین DOM را دوباره render کرد. دو عنصر در react به این شکل عمل می کنند: prop و state. هر زمان که یکی از اینها تغییر کند DOM نیز به روز رسانی می شود

تغییر دادن state به صورت مستقیم اشتباه است

this.state.persons[0].name = 'Maximilian';

برای تغییر state باید از تابعی از پیش تعریف شده در Component به نام setState استفاده کنیم. این متد به عنوان آرگومان یک شیء جاوا اسکریپتی می گیرد. سپس آن را با state قبلی خودمان ادغام می کند.

Components :

یک کامپوننت یک کلاس یا تابع جاوا اسکریپت است که به صورت اختیاری ورودی‌ها را می‌پذیرد، یعنی ویژگی‌ها (props) و یک عنصر React را برمی‌گرداند که توضیح می‌دهد چگونه بخشی از UI (رابط کاربر) باید ظاهر شود.

“props” نحوه دستیابی component ها به property های خودشان است.

### [کی باید از Class Component بجای Function Component استفاده کنیم؟](https://react.sayjeyhi.com/#7f992f04b0eaeb26ce0569c68d51c1d6)میشه گفت هیچ لزومی به این‌کار نیست(مگر در مواقع خیلی خاص مثل error boundaryها) و از ورژن 16.8 ری‌اکت به بعد و با اضافه شدن هوک‌ها به فانکشن کامپوننت‌ها، شما می‌تونین از state یا lifecycle methodها یا تمامی فیچرهایی که قبلا فقط در کلاس کامپوننت ‌ها قابل استفاده بود، توی فانکشن کامپوننت‌هاتون استفاده کنین. ولی قبلا اگه کامپوننت نیاز به state یا lifecycle methods داشت از کلاس کامپوننت‌ها باید استفاده می‌کردیم و در غیر این صورت می‌رفتیم سراغ فانکشن کامپوننت‌ها.

.

منابع:

<https://www.roxo.ir/state-in-react>

<https://errorweb.ir/%DA%A9%D8%A7%D9%85%D9%BE%D9%88%D9%86%D9%86%D8%AA-component-%D9%87%D8%A7-%D9%88-%D9%BE%D8%B1%D8%A7%D9%BE%D8%B3-props-%D8%AF%D8%B1-%D8%B1%DB%8C-%D8%A7%DA%A9%D8%AA-react/>

<https://medium.com/the-andela-way/understanding-react-components-37f841c1f3bb#:~:text=Components%20are%20the%20building%20blocks,(User%20Interface)%20should%20appear>.

1. چه مواردی باعت re-render شدن کامپوننت میشود

Re-render؟ رندر دوم یا بعدی برای به روز رسانی حالت به عنوان رندر مجدد نامیده می شود

کامپوننت‌های React هر زمان که تغییری در وضعیت یا ویژگی‌های آن‌ها ایجاد شود، به‌طور خودکار دوباره ارائه می‌شوند.

یک به روز رسانی ساده حالت، از هر نقطه ای از کد، باعث می شود که تمام عناصر رابط کاربری (UI) دوباره به طور خودکار رندر شوند. با این حال، ممکن است مواردی وجود داشته باشد که متد render() به برخی داده های دیگر بستگی دارد. رندر مجدد می تواند به هر یک از سه دلیل ذکر شده ایجاد شود:

1. Update in State
2. Update in prop
3. Re-rendering of the parent component

<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>.