

: npm vs npx-1

npm (Node Package Manager):

npm بزرگترین نرم افزار رجیستری است. و یک online repository برای انتشار پروژه های Open-source در Node.js است.

مجموعه‌ای از کتابخانه‌ها و برنامه‌های Node.js در npm قرار دارد

رجیستری شامل بیش از 800000 packages است.

یک ابزار کاند لاین است

npx(Node Package Execute)

یک ابزار CLI است که هدف آن آسان کردن نصب و مدیریت وابستگی های میزبانی شده در رجیستری npm است.

وقتی npm را بالای نسخه 5.2.0 نصب کردید، به طور خودکار npx نصب می شود. Npm یک package runner است که می تواند هر package ای را که می خواهید از رجیستری بدون نصب آن package اجرا کند.

npm	npx
اگر می خواهید بسته را از طریق npm اجرا کنید، باید آن بسته را در package.json خود مشخص کرده و آن را به صورت local نصب کنید.	یک package می تواند بدون نصب شدن قابل اجرا باشد، package هایی که قبلاً نصب نشده باشند به طور خودکار نصب می شوند.
Npm ابزاری است که برای نصب package ها استفاده می شود.	اما در npx می توانید بدون نصب استفاده کنید:مثلاً: npx create-react-app myApp

منبع:

<https://nodejs.org/en/knowledge/getting-started/npm/what-is-npm/>

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

What are components in react -2

components

کامپوننت ها بیت های کد مستقل و قابل استفاده مجدد هستند. هر بخشی از کد که ماهیت مستقل و پر تکراری دارد را می توان یک کامپوننت کرد (یک بار کد زدن و چندین بار استفاده کردن) به صورت مجزا کار می کنند و HTML را برمی گردانند. و خروجی هر کامپوننت سبب `update` شدن درخت VDOM خواهد شد. عموماً برای هر کامپوننت یک فایل مستقل در نظر گرفته می شود و اینگونه کد ها منظم تر خواهد بود.

کامپوننت ها در دو نوع `Class Components` و `Function Component` ها هستند. `Function component` ها به روز تر و راحت تر هستند.

:State in components

`State` یک شی جاوا اسکریپت ساده است که توسط `React`، داده های پویای یک کامپوننت را ذخیره می کند و رفتار کامپوننت را تعیین می کند.

از آنجا که `state` پویاست، این پویایی، یک کامپوننت را قادر می سازد تا اطلاعات را بین رندر ها ردیابی کند و پویا و تعاملی باشد. پس هر داده ای که قرار نیست `update` شود و تغییر کند در `state` قرار نمی گیرد.

state فقط در یک Class Components قابل استفاده است. برای تغییر دادن مقادیر state می توانیم از تابع setState استفاده کنیم و آنها را به روز کنیم (State درست مانند هر متغیری که در یک تابع اعلام شده است). تفاوت در این است که یک متغیر "عادی" با پایان تابع "ناپدید می شود"، اما State توسط React حفظ می شود.

هر کامپوننت یک state دارد. تغییرات و به روز رسانی های محتوای state سبب update درخت react VDOM می شود و سپس react فقط بخش های تغییر یافته VDOM را در DOM به روز رسانی می کند. حالا این تغییرات دقیق و جزئی DOM باعث تغییر صفحه و پیچ خواهد شد.

منبع:

https://www.w3schools.com/react/react_components.asp#:~:text=Components%20are%20independent%20and%20reusable,will%20concentrate%20on%20Function%20components.

<https://medium.com/edonec/state-in-react-an-overview-a182675cee2c#:~:text=State%20is%20a%20plain%20JavaScript%20object%20used%20by%20React%20to,variable%20declared%20in%20a%20function.>

3- What causes the component to be rendered

یک به روز رسانی ساده state، از هر نقطه ای از کد، باعث می شود که تمام عناصر رابط کاربری (UI) دوباره به صورت خودکار رندر شوند. با این حال، 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

Update in state:1

تغییر state می تواند از تغییر prop یا setState برای به روز رسانی یک متغیر باشد. Component هم state به روز شده را دریافت کرده و react هم Component را re-render میکند تا تغییرات را به صفحه منتقل کند.

Update in prop:2

به همین ترتیب تغییر در prop منجر به تغییر state و تغییر state منجر به re-render شدن Component توسط React می شود.

Re-rendering of parent component:3

هر زمان که تابع render در Component ها فراخوانی شود، تمام مولفه های فرزند بعدی آن، صرف نظر از اینکه prop های آنها تغییر کرده یا نه، دوباره render می شوند. در کل React هر بار که state تغییر می کند render شدن را برنامه ریزی می کند (برنامه ریزی یک render به این معنی نیست که بلافاصله این اتفاق می افتد، ممکن است زمان ببرد و در بهترین لحظه انجام شود).

منبع:

<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>