

About Document Object Model (DOM)

همانطور که می‌دانیم DOM به عنوان یک ساختار داده درختی نشان داده می‌شود. به همین دلیل، تغییرات و به‌روزرسانی‌های DOM سریع است. اما پس از تغییر، عنصری به‌روزرسانی می‌شود و در پی آن باید عناصری که به هر طریقی به آن مربوطند نیز مجدداً رندر شوند؛ تا در نهایت به‌روزرسانی کل UI محقق شود.

رندر مجدد و یا re-painting UI دقیقاً همان چیزی است که باعث کند شدن DOM می‌شود. بنابراین، هرچه مولفه UI شما بیشتر باشد، به همان میزان به‌روزرسانی DOM نیز زمانبرتر است. چرا که برای هر بروزرسانی DOM، تمامی مولفه‌ها مجدداً رندر می‌شوند

What is Virtual DOM?

مفهوم Virtual DOM فقط نمایش مجازی DOM است. هر زمان که وضعیت برنامه ما تغییر کند، Virtual DOM به جای Real DOM به‌روز می‌شود. هنگامی که عناصر جدیدی به UI اضافه می‌شوند، یک Virtual DOM ایجاد می‌شود که نحوه نمایش آن به صورت درختی است. هر عنصر گره‌ای بر روی این درخت است. اگر حالت هر یک از عناصر تغییر کند، یک درخت Virtual DOM جدید ایجاد می‌شود. سپس، این درخت با درخت DOM مجازی قبلی، مقایسه می‌شود. پس از انجام این کار، Virtual DOM این تغییرات را به بهترین روش ممکن بر روی Real DOM اعمال می‌کند. این کار تضمین می‌کند که حداقل عملیات بر روی Real DOM انجام می‌شود. از این رو، هزینه به‌روزرسانی آن کاهش می‌یابد.

Virtual DOM in React

در React، هر قطعه UI یک جزء است و هر یک از مولفه‌ها نیز حالتی دارند. React از الگوی قابل مشاهده پیروی می‌کند و گوش به زنگ است تا زمانی که حالت یکی از اجزاء تغییر کند. هنگام رخ دادن این اتفاق، React سریعاً درخت Virtual DOM را به‌روز می‌کند. پس از این به‌روزرسانی، React نسخه فعلی Virtual DOM را با نسخه قبلی آن مقایسه می‌کند (diffing).

هنگامی که React اطلاع پیدا می‌کند که اشیایی در Virtual DOM تغییر کرده‌اند، React تنها آن اشیاء را در Real DOM به‌روزرسانی می‌کند. این امر باعث می‌شود عملکرد مقایسه به مراتب بهتر از مواقعی باشد که Real DOM را به صورت مستقیم دستکاری می‌کنیم.

References

<https://reactjs.org/docs/faq-internals.html>

<https://www.geeksforgeeks.org/reactjs-virtual-dom/>