

defer , async در جاوا اسکریپت

هنگامی که قرار است یک صفحه در مرورگر لود شود ابتدا کدهای html لود میشوند. در صورتی که اسکریپت هم داشته باشیم مرورگر ابتدا به سراغ این اسکریپت ها میرود و تا زمانی که لود و اجرای آنها تمام نشده کدهای اچتمل را لود نمیکند. حال فرض کنید یک اسکریپت سنگین در صفحه داشته باشیم و این باعث میشود اجرای کدهای اچتمل بلاک شود. البته این مشکلات در حالتی اتفاق می افتد که اسکریپت ها بالا یا بین کدهای اچتمل قرار گیرند.

روش اول: قرار دادن اسکریپت ها در پایین صفحه

اولین روش برای حل این مشکل این است که اسکریپت ها را در پایین کدهای اچتمل قرار دهیم. مثلا قبل از تگ `body`. مشکل این روش این است که ممکن است کدهای اچتمل بسیار طولانی داشته باشیم که باعث تاخیر در اجرای اسکریپت ها شوند. و روش چندان مناسبی نیست.

روش دوم استفاده از خاصیت defer

این خاصیت باعث میشود که اسکریپت ها در بک گراند لود شوند و کدهای اچتمل و dom بدون تاخیر اجرا شوند و وقتی لود آنها تمام شد اسکریپت ها اجرا شوند.

نکته

خاصیت defer فقط برای اسکریپت های اکسترنال یا خارجی استفاده میشود. در واقع اسکریپتی که خاصیت src داشته باشد.

روش سوم استفاده از خاصیت async

این خاصیت باعث میشود اسکریپت در یک حالت مستقل از همه چیز قرار گیرد. در واقع سایر اسکریپت ها و اچتمل و dom منتظر اجرای اسکریپت های Async نمی مانند و برعکس. این اسکریپت ها در بک گراند لود میشوند و باعث بلاک لود اچتمل نمی شوند. اما به محض لود شدن اجرا میشوند.

اسکریپت های داینامیک

گاهی اوقات لازم است یک سری اسکریپت به صورت داینامیک به صفحه اضافه شوند. این اسکریپت ها بالاافاصله بعد از اضافه شدن شروع به لود و اجرا می کنند. به صورت پیش فرض این اسکریپت ها از خاصیت async استفاده میکنند. البته میتوانیم این خاصیت را false کنیم تا اسکریپت در حالت defer قرار گیرد. مثال:

```
function loadScript(src) {
```

```
let script = document.createElement('script');  
script.src = src;  
script.async = false;  
document.body.append(script);  
}
```

```
// long.js runs first because of async=false  
loadScript("/article/script-async-defer/long.js");  
loadScript("/article/script-async-defer/small.js");
```