ASYNC & DEFER in

JS

Sahith Bodla

When the browser loads **HTML** and comes across a **<script>** tag, it can’t continue building the DOM.

the browser must wait for the script to download, execute the downloaded script, and only then can it process the rest of the page.

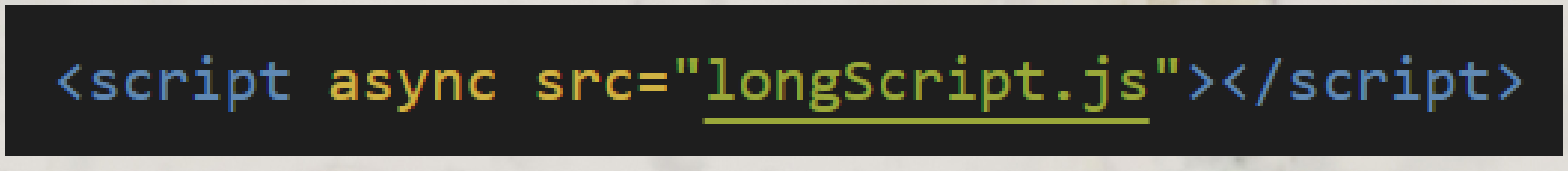
This leads to issues

- Scripts cannot access or interact with DOM elements below them, limiting their ability to add event handlers or perform actions.
- A large script at the top of the page blocks content rendering until downloaded and executed, delaying user visibility.

async

The solution is using **async** attribute inside **script** tag

The **async** attribute tells the browser not to wait for the script. Instead, the browser will continue to process the HTML, build DOM. The script loads **in the background**, and then runs when they are ready.

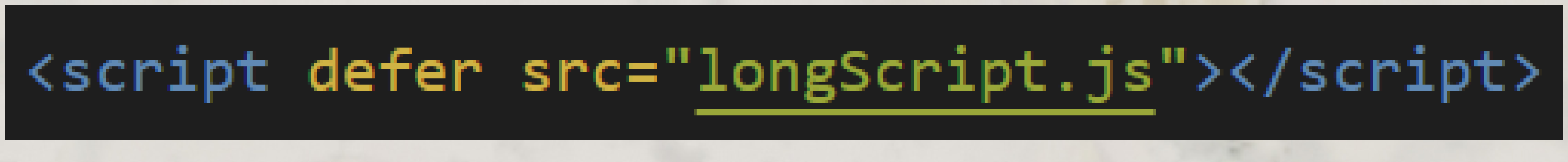


```
<script async src="longScript.js"></script>
```

defer

We can also use **defer** attribute inside **script** tag as a solution.

The **defer** attribute tells the browser not to wait for the script. Instead, the browser will continue to process the HTML, build DOM. The script loads **in the background**, and then runs when the DOM is fully built.



```html
<script defer src="longScript.js"></script>
```

So, what's the difference between async and defer?

Scripts with **defer** always execute when the DOM is ready (but before **DOMContentLoaded** event).

DOMContentLoaded and **async** scripts don't wait for each other, the one which is ready first will be executed first the other difference is that

Deferred scripts keep their relative order, just like regular scripts.

Where as **async** scripts don't maintain relative order, as async scripts run in the **load-first** order..

Just note that the async and defer attributes will be ignored if the src attribute is not present in the <script> tag.

async & defer

LIKE
&
FOLLOW

for more

Sahith Bodla