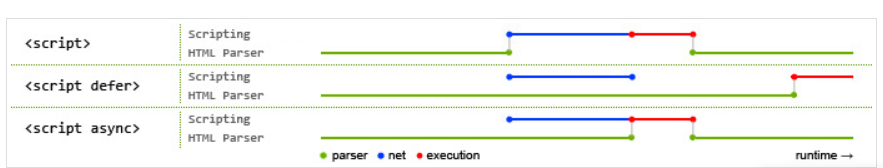
当HTML解析器解析到<script>元素时，默认先下载和执行脚本，DOM树解析立刻停止，也是先下载脚本，下载的时候不再解析HTML。

也就是说，脚本默认是同步和阻塞的。

异步脚本,<script async>,异步脚本下载的时候，DOM解析继续。异步脚本会在下载完后尽快执行，这样不用在下载脚本的时候阻塞文档解析。

延迟脚本，<script defer>，延迟脚本下载的时候，DOM树解析继续。延迟脚本下载完成之后，不会立即执行，而是等到整个DOM加载完毕，它再执行，也就是当document的readyState为DOMContentLoaded事件触发时开始执行延迟脚本。



当标签中同时又defer和async时，async>defer。延迟脚本会按照下载的顺序依次执行，但是异步脚本的执行可能是无序的。

为什么不能使用document.write()？

此方法是将html字符写入文档流。一旦文档流已经关闭了（document.close()），那么document.write（）会重新利用document.open()打开一个新的文档流，原来的文档流会被清空，已渲染的页面会被清除，浏览器重新构建DOM并渲染页面。这就是为什么使用write方法会照成原先的页面”丢失”。而当文档流还没关闭，并不会用document.open。

因此，异步脚本中最好不要有write，当然，你使用异步脚本就默认了不会使用write。