## [defer](https://learn.javascript.ru/script-async-defer" \l "defer)

Атрибут defer сообщает браузеру, что он должен продолжать обрабатывать страницу и загружать скрипт в фоновом режиме, а затем запустить этот скрипт, когда он загрузится.

* Скрипты с defer никогда не блокируют страницу.
* Скрипты с defer всегда выполняются, когда дерево DOM готово, но до события DOMContentLoaded.

**[async](https://learn.javascript.ru/script-async-defer" \l "async)**

Атрибут async означает, что скрипт абсолютно независим:

* Страница не ждёт асинхронных скриптов, содержимое обрабатывается и отображается.
* Событие DOMContentLoaded и асинхронные скрипты не ждут друг друга:
  + DOMContentLoaded может произойти как до асинхронного скрипта (если асинхронный скрипт завершит загрузку после того, как страница будет готова),
  + …так и после асинхронного скрипта (если он короткий или уже содержится в HTTP-кеше)
* Остальные скрипты не ждут async, и скрипты casync не ждут другие скрипты.

Так что если у нас есть несколько скриптов с async, они могут выполняться в любом порядке. То, что первое загрузится – запустится в первую очередь:

## [Итого](https://learn.javascript.ru/script-async-defer" \l "itogo)

У async и defer есть кое-что общее: они не блокируют отрисовку страницы. Так что пользователь может просмотреть содержимое страницы и ознакомиться с ней сразу же.

Но есть и значимые различия:

|  | **Порядок** | **DOMContentLoaded** |
| --- | --- | --- |
| async | Порядок загрузки (кто загрузится первым, тот и сработает). | Не имеет значения. Может загрузиться и выполниться до того, как страница полностью загрузится. Такое случается, если скрипты маленькие или хранятся в кеше, а документ достаточно большой. |
| defer | Порядок документа (как расположены в документе). | Выполняется после того, как документ загружен и обработан (ждёт), непосредственно перед DOMContentLoaded. |