

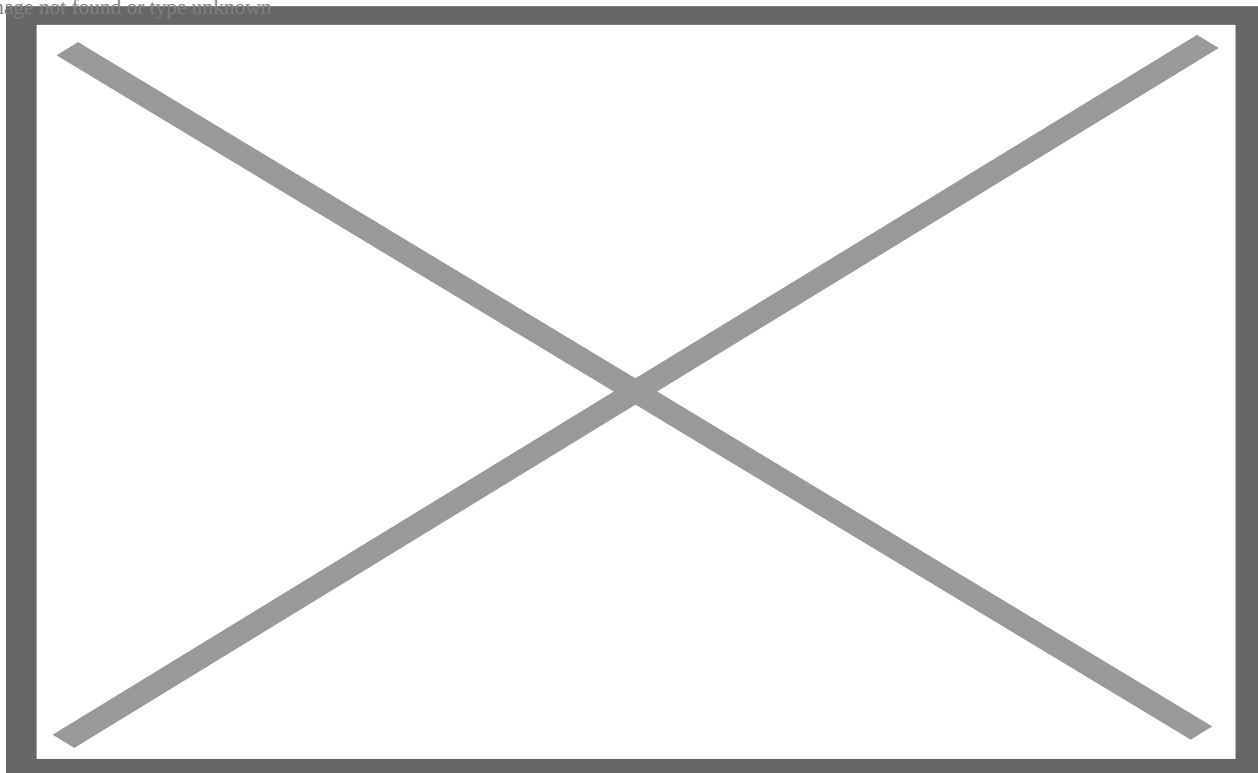


Как использовать HTML тег Script

Описание

Вы можете захотеть добавить на свои веб-страницы файлы JavaScript, библиотеки сторонних разработчиков или фреймворки. В этом случае важно знать, как быстро сделать эти добавления. В этом может помочь тег HTML script. Эта мощная функция тегов позволяет разработчикам без труда включать в HTML-документы распространенные библиотеки, плагины и скрипты для повышения производительности. Это не только повышает скорость разработки, но и способствует лучшей отладке и более плавному сотрудничеству с другими разработчиками.

Image not found or type unknown



Что такое тег script в HTML?

Вы можете вставить файл JavaScript или любой другой язык сценариев в HTML-документ с помощью тегов `<script>`. Этот код может быть размещен непосредственно в теге, или вы можете использовать внешний файл и ссылаться на него через атрибут `src`, чтобы сократить повторное кодирование и оптимизировать для лучшей производительности. Когда HTML-документ загружается, браузер анализирует и выполняет сценарий, добавляя динамическое поведение и интерактивность на страницу. Например, с помощью JavaScript можно реагировать на действия пользователя, обновлять содержимое и создавать анимацию.

Включая тег `<script>` в заголовок или тело HTML-документа, вы можете управлять различными функциями, такими как асинхронность и отсрочка, которые могут направлять сценарии на выполнение приказов. Кроме того, атрибут `src` позволяет определять внешние ресурсы для кода скрипта, оставаясь при этом очень настраиваемым. Теперь давайте рассмотрим некоторые популярные атрибуты тегов скриптов.

Атрибуты тега script

Тег script имеет множество атрибутов, которые можно использовать для управления загрузкой и выполнением кода. К этим атрибутам относятся:

- Куда поместить тег script в HTML.
- Что он делает (асинхронно, откладывая загрузку и т.д.).
- Какой приоритет он должен иметь при получении браузером.
- Какой криптографический хэш следует использовать для проверки.
- И включена ли блокировка или нет.

Все эти атрибуты служат разным целям и помогают поддерживать производительность и безопасность вашего сайта. Ниже приведены некоторые из наиболее используемых атрибутов тегов скриптов.

Async

Этот атрибут представляет собой булево значение, которое возвращает true, если оно присутствует. Атрибут async сообщает браузеру о необходимости асинхронной загрузки скрипта, так что после загрузки страница будет отображаться без помех. В результате вы можете наслаждаться высокой скоростью работы.

Crossorigin

Атрибут crossorigin является ценным преимуществом при загрузке скриптов с разных доменов. По умолчанию браузер не будет передавать определенную информацию в сообщениях об ошибках, отправляемых в window.onerror для скриптов, полученных из внешних доменов. Однако, используя этот атрибут, вы можете включить протоколирование любых ошибок, генерируемых этими типами запросов, что позволит повысить производительность и улучшить пользовательский опыт на всех страницах вашего сайта.

Defer

При использовании этого атрибута браузер будет ждать, пока вся страница не будет разобрана, прежде чем выполнить какой-либо сценарий. Это особенно полезно при наличии нескольких скриптов на одной странице, так как гарантирует,

что они будут выполняться в порядке их появления в HTML-коде. Воздержитесь от использования этого атрибута, если вы не будете включать атрибут `src`, так как это приведет к неэффективности встроенных скриптов.

Fetch Priority

Эта функция позволяет установить приоритет источника сценария при извлечении веб-браузером для повышения производительности. Приоритет запросов может быть изменен в зависимости от различных факторов, таких как размер и количество ресурсов, запрашиваемых одновременно, а также от условий сети:

- **Высокий.** Этот сигнал означает высокоприоритетный вызов внешних скриптов и должен иметь приоритет перед другими запросами на выборку.
- **Низкий.** Сигналы извлекаются с меньшим приоритетом, чем другие внешние сценарии.
- **Авто.** По умолчанию сигналы определяют приоритет выборки внешних скриптов по сравнению с другими автоматически.

Integrity

С помощью этой возможности вы можете передать браузеру криптографический хэш ресурса скрипта, чтобы убедиться, что никто не изменил его во время загрузки. Это гарантирует, что ваш скрипт должным образом защищен и безопасен для использования.

Nomodule

Недавно включенный в HTML5 атрибут `nomodule` тега `<script>` предоставляет разработчикам возможность указать, что современные браузеры не должны выполнять их сценарий. Это особенно полезно, если ваш код не следует стандартам модуля ECMAScript и должен выполняться только на старых веб-браузерах.

Nonce

Nonce означает число, используемое один раз. Эта функция позволяет создать непредсказуемую, однократно используемую строку (известную как “nonce”), которая подтверждает, что сценарий не был изменен во время загрузки. Этот

уникальный идентификатор затем включается в заголовок политики безопасности содержимого вашего ответа и проверяется любым используемым веб-браузером. Благодаря этой мере безопасности вы можете быть уверены, что в процессе передачи не произошло никаких вредоносных изменений.

Referrerpolicy

С помощью этой функции вы можете задать заголовок Referrer-Policy ответа, который контролирует отправку любой справочной информации при запросе сценария.

- no-referrer: заголовок Referrer не будет передаваться.
- no-referrer-when-downgrade: Если источник не имеет Transport Layer Security (TLS, также известный как HTTPS), заголовок Referer не будет передаваться.
- происхождение: Отправляемый на сервер реферер будет ограничен исключительно его источником: включая протокол, имя хоста и порт.
- origin-when-cross-origin: В качестве меры предосторожности реферер, отправляемый во внешние источники, будет включать только схему, имя хоста и номер порта. Однако при внутреннем переходе на страницу того же происхождения он также будет указывать путь к ней.
- same-origin: Запросы, сделанные в пределах одного домена, будут включать реферер. Однако запросы, сделанные в разных доменах, не будут содержать данных о реферере.
- strict-origin: При переходе от безопасного протокола (HTTPS) к другому безопасному направлению, убедитесь, что происхождение документа отправляется как часть referrer; однако не отправляйте его при переходе к небезопасной среде (HTTPS→HTTP).
- strict-origin-when-cross-origin: При выполнении запроса с одинаковым происхождением всегда предоставляйте полный URL. Отправляйте источник только в том случае, если оба пункта назначения имеют одинаковый уровень безопасности (например, HTTPS→HTTPS). Наконец, опускайте любой заголовок при доступе к менее безопасному сайту (например, HTTPS→HTTP).
- unsafe-URL: Ссылка будет включать происхождение и путь, но опускает фрагмент, пароль и имя пользователя. Это значение является небезопасным, раскрывая происхождение и путь из безопасных источников TLS в незащищенные места.

Src

Этот атрибут определяет точное местоположение внешнего ресурса скрипта, который необходимо загрузить. Вместо того чтобы включать скрипт непосредственно в документ, это отличная альтернатива.

Type

Этот атрибут определяет тип скрипта, обычно это text/javascript. Атрибут type необходим для того, чтобы браузер имел четкое представление о работе с вашими скриптами. Если опустить этот атрибут или установить его как пустую строку или MIME-тип JavaScript, браузер автоматически сообщит, что ваш сценарий содержит классический код JavaScript, что делает его более интерпретируемым и легким для обработки.

Blocking

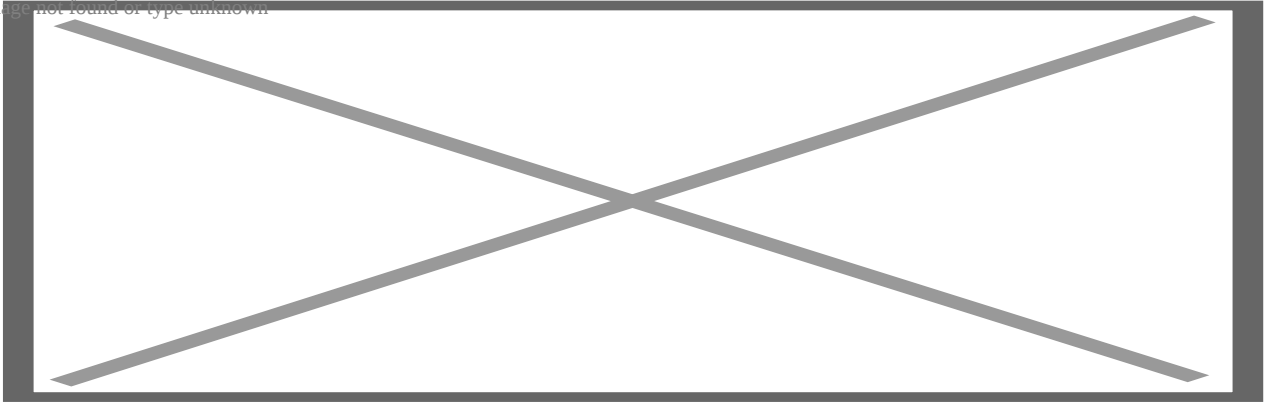
Этот атрибут используется для указания того, как сценарий должен загружаться и выполняться. Он блокирует определенные операции при получении сценария. Теперь, когда мы узнали об атрибутах тегов сценария, давайте рассмотрим несколько примеров.

Примеры HTML тегов script

Импорт сценария с помощью 'Src'

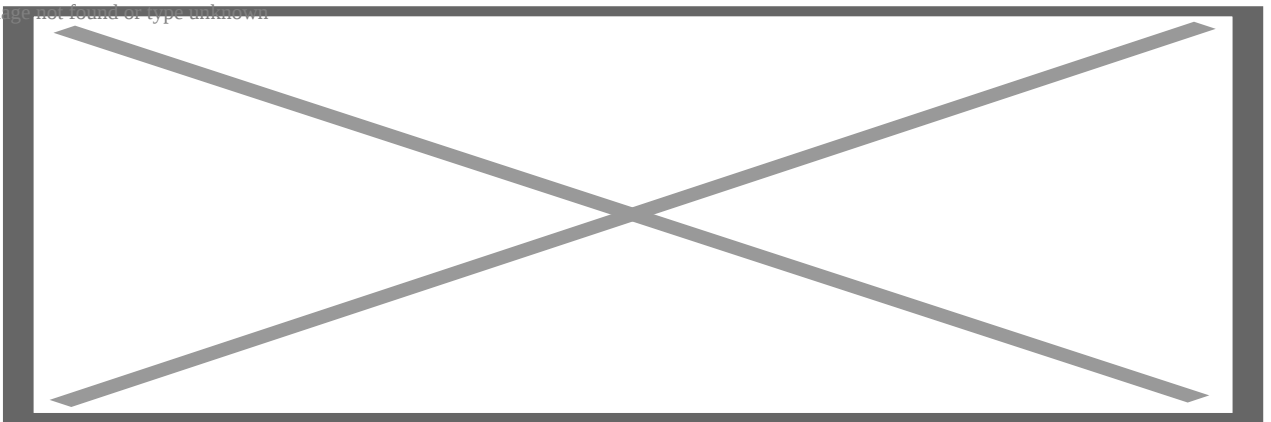
Конечно, вы можете включить код JavaScript в тот же документ, что и HTML. Однако вы также можете выделить код в отдельный файл и импортировать его с помощью тега script и атрибута src. Посмотрите на приведенный ниже пример, чтобы увидеть, как это делается.

Image not found of type unknown



Если вы хотите включить скрипт в строку, посмотрите пример ниже.

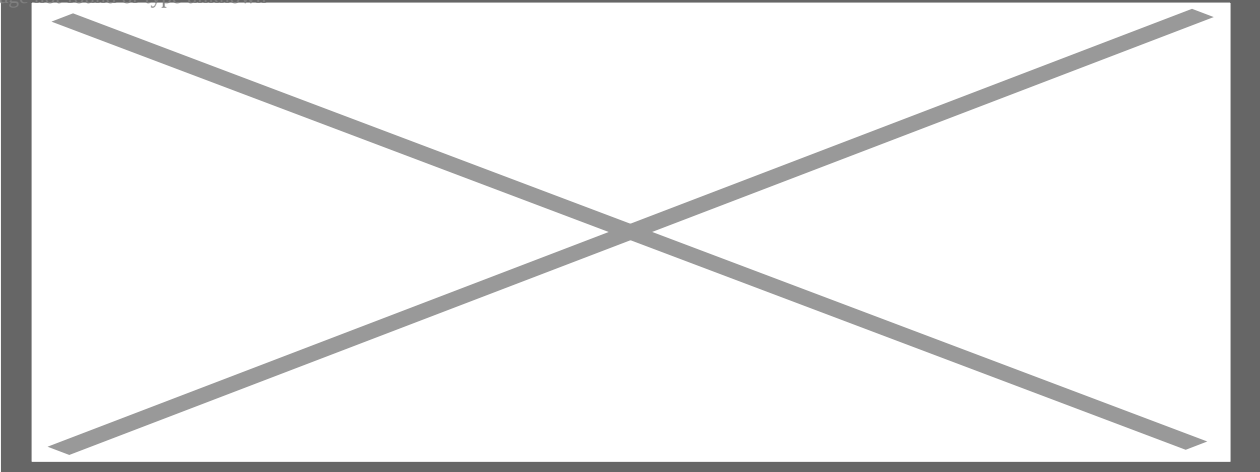
Image not found of type unknown



Тег script с использованием атрибута 'Defer'

В приведенном ниже примере вы видите тег script, используемый в теле документа. В этом теге используется атрибут defer, который откладывает выполнение сценария до окончания рендеринга HTML.

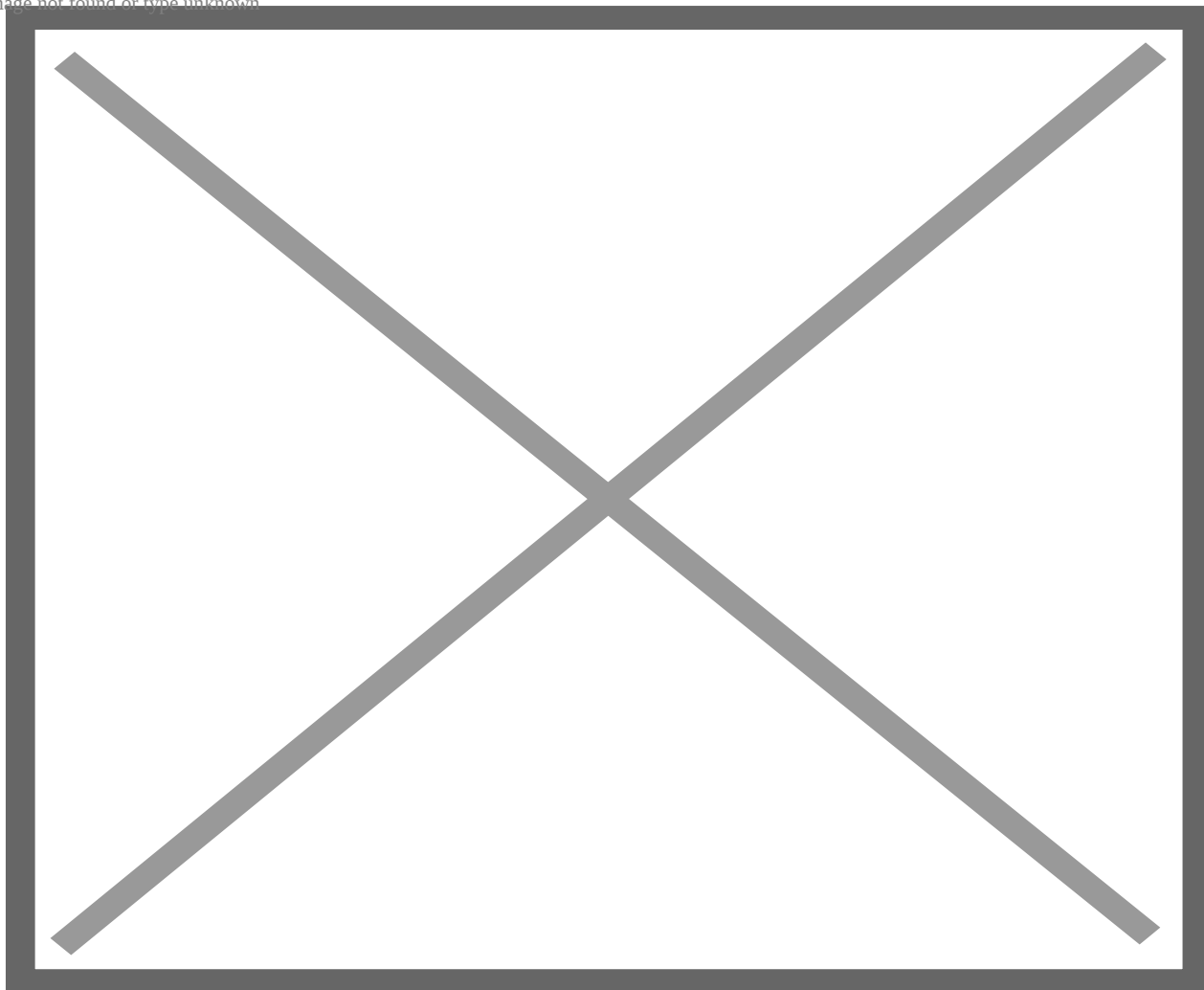
Image not found or type unknown



Тег script с использованием атрибута 'Integrity'

Здесь приведен пример использования атрибута целостности с тегом сценария. Тег сценария включает хэш-значение в атрибут целостности, чтобы убедиться, что сценарий не был подделан. Хэш, указанный в теге скрипта, сравнивается с хэшем загруженного скрипта. Если они совпадают, сценарий будет признан действительным, загружен и выполнен. Атрибут cross-origin является анонимным, что позволяет загружать скрипт из другого домена.

Image not found or type unknown



Начало работы с HTML тегом script

Использование тега script в HTML – это эффективный способ добавления сценариев на ваши веб-страницы. Он обеспечивает безопасный, надежный и последовательный метод загрузки содержимого из внешних источников. При использовании тега script убедитесь, что вы понимаете, что означает каждый атрибут и как он влияет на производительность вашей страницы, чтобы вы могли принимать обоснованные решения о размещении скриптов на ваших веб-страницах. Это обеспечит их оптимальную загрузку и бесперебойную работу вашего сайта.

Дата Создания

01.06.2023