Adding content and example to the link types “modulepreload” document.

Link : <https://developer.mozilla.org/en-US/docs/Web/HTML/Link_types/modulepreload>

**Original Source Document.**

<p><span class="seoSummary">The <strong><code>modulepreload</code></strong> keyword for the {{HTMLAttrxRef("rel", "link")}} attribute of the {{HTMLElement("link")}} element provides a declarative way to preemptively fetch a <a href="/en-US/docs/Web/JavaScript/Guide/Modules">module script</a> and its dependencies, and store them in the document's module map for later evaluation.</span></p>

<h2 id="Specifications" name="Specifications">Specifications</h2>

<table class="standard-table">

<thead>

<tr>

<th scope="col">Specification</th>

<th scope="col">Status</th>

<th scope="col">Comment</th>

</tr>

</thead>

<tbody>

<tr>

<td>{{SpecName("HTML WHATWG", "#link-type-modulepreload", "modulepreload")}}</td>

<td>{{Spec2("HTML WHATWG")}}</td>

<td></td>

</tr>

</tbody>

</table>

<h2 id="Browser\_compatibility">Browser compatibility</h2>

<div class="hidden">The compatibility table in this page is generated from structured data. If you'd like to contribute to the data, please check out <a href="https://github.com/mdn/browser-compat-data">https://github.com/mdn/browser-compat-data</a> and send us a pull request.</div>

<p>{{Compat("html.elements.link.rel.modulepreload")}}</p>

**Content and examples I will be adding**

The modulepreload keyword can be used with link elements. Using the ‘modulepreload’ will create an external resource link.

This keyword is also an alternative to the preload keyword. It makes use of the fetch behaviour for module scripts and will then place the result into an appropriate module map for evaluating later.

In contrast, a similar external resource link using the preload keyword will place the result in the preload cache, doing this will still not affect the module map of the document.

The appropriate times to fetch and process the linked resource for such a link are:

* When the external resource link is created on a link element that is already browsing-context connected.
* When the external resource link's link element becomes browsing context connected.
* When the href attribute of the link element of an external resource link that is already browsing-context connected is changed.

Example

<!DOCTYPE html>

<html lang="en">

<title>IRCFog</title>

<link rel="modulepreload" href="app.mjs">

<link rel="modulepreload" href="helpers.mjs">

<link rel="modulepreload" href="irc.mjs">

<link rel="modulepreload" href="fog-machine.mjs">

<script type="module" src="app.mjs"> ...

In the above example we can that the application developer has used modulepreload to declare all of the modules in their module graph,which would then ensure that the user agent initiates the fetch for them all. Without such preloading, the user agent might need to go through multiple network roundtrips before discovering helpers.mjs.

**After adding example to Source document.**

|  |
| --- |
| <p><span class="seoSummary">The <strong><code>modulepreload</code></strong> keyword for the {{HTMLAttrxRef("rel", "link")}} attribute of the {{HTMLElement("link")}} element provides a declarative way to preemptively fetch a <a href="/en-US/docs/Web/JavaScript/Guide/Modules">module script</a> and its dependencies, and store them in the document's module map for later evaluation.</span></p>  <p>The modulepreload keyword can be used with link elements. Using the ‘modulepreload’ will create an external resource link.</p>  <p>This keyword is also an alternative to the preload keyword. It makes use of the fetch behaviour for module scripts and will then place the result into an&nbsp; appropriate module map for evaluating later.</p>  <p>In contrast, a similar&nbsp;external resource link using the preload keyword will place the result in the preload cache, doing this will still not affect the module map of the document.</p>  <p>The appropriate times to&nbsp;fetch and process the linked resource&nbsp;for such a link are:</p>  <ul>  <li>When the&nbsp;external resource link&nbsp;is created on a&nbsp;link&nbsp;element that is already&nbsp;browsing-context connected.</li>  <li>When the&nbsp;external resource link's&nbsp;link&nbsp;element&nbsp;becomes browsing context connected.</li>  <li>When the&nbsp;href&nbsp;attribute of the&nbsp;link&nbsp;element of an&nbsp;external resource link&nbsp;that is already&nbsp;browsing-context connected&nbsp;is changed.</li>  </ul>  <p>Example</p>  <pre class="brush: html">  &lt;!DOCTYPE html&gt;  &lt;html lang="en"&gt;  &nbsp;&lt;title&gt;IRCFog&lt;/title&gt;  &nbsp;&lt;link rel="modulepreload" href="app.mjs"&gt;  &nbsp;&lt;link rel="modulepreload" href="helpers.mjs"&gt;  &nbsp;&lt;link rel="modulepreload" href="irc.mjs"&gt;  &nbsp;&lt;link rel="modulepreload" href="fog-machine.mjs"&gt;  &lt;script type="module" src="app.mjs"&gt; ...</pre>  <p>In the above example we can that the application developer has used&nbsp;modulepreload&nbsp;to declare all of the modules in their module graph,which would then ensure that the user agent initiates the fetch for them all. Without such preloading, the user agent might need to go through multiple network roundtrips before discovering&nbsp;helpers.mjs.</p>  <h2 name="Specifications"><br />  <br />  <br />  Specifications</h2>  <table class="standard-table">  <thead>  <tr>  <th scope="col">Specification</th>  <th scope="col">Status</th>  <th scope="col">Comment</th>  </tr>  </thead>  <tbody>  <tr>  <td>{{SpecName("HTML WHATWG", "#link-type-modulepreload", "modulepreload")}}</td>  <td>{{Spec2("HTML WHATWG")}}</td>  <td></td>  </tr>  </tbody>  </table>  <h2 id="Browser\_compatibility">Browser compatibility</h2>  <div class="hidden">The compatibility table in this page is generated from structured data. If you'd like to contribute to the data, please check out <a href="https://github.com/mdn/browser-compat-data">https://github.com/mdn/browser-compat-data</a> and send us a pull request.</div>  <p>{{Compat("html.elements.link.rel.modulepreload")}}</p> |

**Preview after making changes to the Document.**

Text, letter

Description automatically generated