The **HTML External Resource Link element (`<link>`)** specifies relationships between the current document and an external resource. This element is most commonly used to link to stylesheets, but is also used to establish site icons (both "favicon" style icons and icons for the home screen and apps on mobile devices) among other things.

To link an external stylesheet, you'd include a `<link>` element inside your `<head>` like this:

```
<link href="main.css" rel="stylesheet">
```

This simple example provides the path to the stylesheet inside an `href` attribute, and a `rel` attribute with a value of `stylesheet`. The `rel` stands for "relationship", and is probably one of the key features of the `<link>` element — the value denotes how the item being linked to is related to the containing document. As you'll see from our Link types reference, there are many different kinds of relationship.

There are a number of other common types you'll come across. For example, a link to the site's favicon:

```
<link rel="icon" href="favicon.ico">
```

There are a number of other icon `rel` values, mainly used to indicate special icon types for use on various mobile platforms, e.g.:

```
<link rel="apple-touch-icon-precomposed" sizes="114x114"
      href="apple-icon-114.png" type="image/png">
```

The `sizes` attribute indicates the icon size, while the `type` contains the MIME type of the resource being linked. These provide useful hints to allow the browser to choose the most appropriate icon available.

You can also provide a media type or query inside a `media` attribute; this resource will then only be loaded if the media condition is true. For example:

```
<link href="print.css" rel="stylesheet" media="print">
<link href="mobile.css" rel="stylesheet" media="screen and (max-width: 600px
```

Some interesting new performance and security features have been added to the `<link>` element too. Take this example:

```
<link rel="preload" href="myFont.woff2" as="font"
      type="font/woff2" crossorigin="anonymous">
```

A `rel` value of `preload` indicates that the browser should preload this resource (see Preloading content with rel="preload" for more details), with the `as` attribute indicating the specific class of content being fetched. The `crossorigin` attribute indicates whether the resource should be fetched with a CORS request.

Other usage notes:

- A `<link>` element can occur either in the `<head>` or `<body>` element, depending on whether it has a link type that is **body-ok**. For example, the `stylesheet` link type is body-ok, and therefore `<link rel="stylesheet">` is permitted in the body. However, this isn't a good practice to follow; it makes more sense to separate your `<link>` elements from your body content, putting them in the `<head>`.
- When using `<link>` to establish a favicon for a site, and your site uses a Content Security Policy (CSP) to enhance its security, the policy applies to the favicon. If you encounter problems with the favicon not loading, verify that the `Content-Security-Policy` header's `img-src` directive is not preventing access to it.
- The HTML and XHTML specifications define event handlers for the `<link>` element, but it is unclear how they would be used.
- Under XHTML 1.0, empty elements such as `<link>` require a trailing slash: `<link />`.
- WebTV supports the use of the value `next` for `rel` to preload the next page in a document series.

Attributes

This element includes the global attributes.

`as`

This attribute is only used when `rel="preload"` or `rel="prefetch"` has been set on the `<link>` element. It specifies the type of content being loaded by the `<link>`, which is necessary for request matching, application of correct content security policy, and setting of correct `Accept` request header. Furthermore, `rel="preload"` uses this as a signal for request prioritization. The table below lists the valid values for this attribute and the elements or resources they apply to.

Value	Applies To
audio	`<audio>` elements

Value	Applies To
document	`<iframe>` and `<frame>` elements
embed	`<embed>` elements
fetch	fetch, XHR This value also requires `<link>` to contain the crossorigin attribute.
font	CSS @font-face
image	`<img>` and `<picture>` elements with srcset or imageset attributes, SVG `<image>` elements, CSS `*-image` rules
object	`<object>` elements
script	`<script>` elements, Worker `importScripts`
style	`<link rel=stylesheet>` elements, CSS `@import`
track	`<track>` elements
video	`<video>` elements
worker	Worker, SharedWorker

`crossorigin`

This enumerated attribute indicates whether CORS must be used when fetching the resource. CORS-enabled images can be reused in the `<canvas>` element without being *tainted*. The allowed values are:

`anonymous`

A cross-origin request (i.e. with an `Origin` HTTP header) is performed, but no credential is sent (i.e. no cookie, X.509 certificate, or HTTP Basic authentication). If the server does not give credentials to the origin site (by not setting the `Access-Control-Allow-Origin` HTTP header) the resource will be tainted and its usage restricted.

`use-credentials`

A cross-origin request (i.e. with an `Origin` HTTP header) is performed along with a credential sent (i.e. a cookie, certificate, and/or HTTP Basic authentication is performed). If the server does not give credentials to the origin site (through `Access-Control-Allow-Credentials` HTTP header), the resource will be *tainted* and its usage restricted.

If the attribute is not present, the resource is fetched without a CORS request (i.e. without sending the `Origin` HTTP header), preventing its non-tainted usage. If invalid, it is handled as if the enumerated keyword **anonymous** was used. See CORS settings attributes for additional information.

`disabled`

For `rel="stylesheet"` only, the `disabled` Boolean attribute indicates whether or not the described stylesheet should be loaded and applied to the document. If `disabled` is specified in the HTML when it is loaded, the stylesheet will not be loaded during page load. Instead, the stylesheet will be loaded on-demand, if and when the `disabled` attribute is changed to `false` or removed.

Setting the `disabled` property in the DOM causes the stylesheet to be removed from the document's `document.styleSheets` list.

`href`

This attribute specifies the URL of the linked resource. A URL can be absolute or relative.

`hreflang`

This attribute indicates the language of the linked resource. It is purely advisory. Allowed values are determined by BCP47. Use this attribute only if the `href` attribute is present.

`importance`

Indicates the relative importance of the resource. Priority hints are delegated using the values:

`auto`: Indicates **no preference**. The browser may use its own heuristics to decide the priority of the resource.

`high`: Indicates to the browser that the resource is of **high** priority.

`low`: Indicates to the browser that the resource is of **low** priority.

> **Note:** The `importance` attribute may only be used for the `<link>` element if `rel="preload"` or `rel="prefetch"` is present.

`integrity`

Contains inline metadata — a base64-encoded cryptographic hash of the resource (file) you're telling the browser to fetch. The browser can use this to verify that the fetched resource has been delivered free of unexpected manipulation. See Subresource Integrity.

`media`

This attribute specifies the media that the linked resource applies to. Its value must be a media type / media query. This attribute is mainly useful when linking to external stylesheets — it allows the user agent to pick the best adapted one for the device it runs on.

> **Notes:**
>
> - In HTML 4, this can only be a simple white-space-separated list of media description literals, i.e., media types and groups, where defined and allowed as values for this attribute, such as `print`, `screen`, `aural`, `braille`. HTML5 extended this to any kind of media queries, which are a superset of the allowed values of HTML 4.

> - Browsers not supporting CSS3 Media Queries won't necessarily recognize the adequate link; do not forget to set fallback links, the restricted set of media queries defined in HTML 4.

`prefetch` Secure context

Identifies a resource that might be required by the next navigation and that the user agent should retrieve it. This allows the user agent to respond faster when the resource is requested in the future.

`referrerpolicy`

A string indicating which referrer to use when fetching the resource:

- `no-referrer` means that the `Referer` header will not be sent.
- `no-referrer-when-downgrade` means that no `Referer` header will be sent when navigating to an origin without TLS (HTTPS). This is a user agent's default behavior, if no policy is otherwise specified.
- `origin` means that the referrer will be the origin of the page, which is roughly the scheme, the host, and the port.
- `origin-when-cross-origin` means that navigating to other origins will be limited to the scheme, the host, and the port, while navigating on the same origin will include the referrer's path.
- `unsafe-url` means that the referrer will include the origin and the path (but not the fragment, password, or username). This case is unsafe because it can leak origins and paths from TLS-protected resources to insecure origins.

`rel`

This attribute names a relationship of the linked document to the current document. The attribute must be a space-separated list of link type values.

`sizes`

This attribute defines the sizes of the icons for visual media contained in the resource. It must be present only if the `rel` contains a value of `icon` or a non-standard type such as Apple's `apple-touch-icon`. It may have the following values:

- `any`, meaning that the icon can be scaled to any size as it is in a vector format, like `image/svg+xml`.
- a white-space separated list of sizes, each in the format *`<width in pixels>x<height in pixels>`* or *`<width in pixels>X<height in pixels>`*. Each of these sizes must be contained in the resource.

> **Note:** Most icon formats are only able to store one single icon; therefore most of the time the `sizes` attribute contains only one entry. MS's ICO format does, as well as Apple's ICNS. ICO is more ubiquitous, so you should use this format if cross-browser support is a concern (especially for old IE versions).

`title`

The `title` attribute has special semantics on the `<link>` element. When used on a `<link rel="stylesheet">` it defines a preferred or an alternate stylesheet. Incorrectly using it may cause the stylesheet to be ignored.

`type`

This attribute is used to define the type of the content linked to. The value of the attribute should be a MIME type such as **text/html**, **text/css**, and so on. The common use of this attribute is to define the type of stylesheet being referenced (such as **text/css**), but given that CSS is the only stylesheet language used on the web, not only is it possible to omit the `type` attribute, but is actually now recommended practice. It is also used on `rel="preload"` link types, to make sure the browser only downloads file types that it supports.

Non-standard attributes

`methods`

The value of this attribute provides information about the functions that might be performed on an object. The values generally are given by the HTTP protocol when it is used, but it might (for similar reasons as for the **title** attribute) be useful to include advisory information in advance in the link. For example, the browser might choose a different rendering of a link as a function of the methods specified; something that is searchable might get a different icon, or an outside link might render with an indication of leaving the current site. This attribute is not well understood nor supported, even by the defining browser, Internet Explorer 4.

`target`

Defines the frame or window name that has the defined linking relationship or that will show the rendering of any linked resource.

Obsolete attributes

`charset`

This attribute defines the character encoding of the linked resource. The value is a space- and/or comma-delimited list of character sets as defined in RFC 2045. The default value is `iso-8859-1`.

> **Usage note:** To produce the same effect as this obsolete attribute, use the `Content-Type` HTTP header on the linked resource.

`rev`

The value of this attribute shows the relationship of the current document to the linked document, as defined by the `href` attribute. The attribute thus defines the reverse relationship compared to the value of the `rel` attribute. Link type values for the attribute are similar to the possible values for `rel`.

> **Note:** This attribute is considered obsolete by the WHATWG HTML living standard (which is the specification MDN treats as canonical). However, it's worth noting

> that rev is *not* considered obsolete in the W3C specification. That said, given the uncertainty, relying on rev is unwise.
>
> Instead, you should use the rel attribute with the opposite link type value. For example, to establish the reverse link for made, specify author. Also this attribute doesn't stand for "revision" and must not be used with a version number, even though many sites misuse it in this way.

Examples

Including a stylesheet

To include a stylesheet in a page, use the following syntax:

```
<link href="style.css" rel="stylesheet">
```

Providing alternative stylesheets

You can also specify alternative style sheets.

The user can choose which style sheet to use by choosing it from the View > Page Style menu. This provides a way for users to see multiple versions of a page.

```
<link href="default.css" rel="stylesheet" title="Default Style">
<link href="fancy.css" rel="alternate stylesheet" title="Fancy">
<link href="basic.css" rel="alternate stylesheet" title="Basic">
```

Providing icons for different usage contexts

You can include links to several different icons on the same page, and the browser will choose which one works best for its particular context using the rel and sizes values as hints.

```
<!-- third-generation iPad with high-resolution Retina display: -->
<link rel="apple-touch-icon-precomposed" sizes="144x144" href="favicon144.pn
```

```
<link rel="apple-touch-icon-precomposed" sizes="144x144" href="favicon144.pn
<!-- iPhone with high-resolution Retina display: -->

<link rel="apple-touch-icon-precomposed" sizes="114x114" href="favicon114.pn
<!-- first- and second-generation iPad: -->
<link rel="apple-touch-icon-precomposed" sizes="72x72" href="favicon72.png">
<!-- non-Retina iPhone, iPod Touch, and Android 2.1+ devices: -->
<link rel="apple-touch-icon-precomposed" href="favicon57.png">
<!-- basic favicon -->
<link rel="icon" href="favicon32.png">
```

Conditionally loading resources with media queries

You can provide a media type or query inside a `media` attribute; this resource will then only be loaded if the media condition is true. For example:

```
<link href="print.css" rel="stylesheet" media="print">
<link href="mobile.css" rel="stylesheet" media="all">
<link href="desktop.css" rel="stylesheet" media="screen and (min-width: 600p
<link href="highres.css" rel="stylesheet" media="screen and (min-resolution:
```

Stylesheet load events

You can determine when a style sheet has been loaded by watching for a `load` event to fire on it; similarly, you can detect if an error has occurred while processing a style sheet by watching for an `error` event:

```
    <script>
    var myStylesheet = document.querySelector('#my-stylesheet');

    myStylesheet.onload = function() {
      // Do something interesting; the sheet has been loaded
    }

    myStylesheet.onerror = function() {
      console.log("An error occurred loading the stylesheet!");
    }
    </script>

    <link rel="stylesheet" href="mystylesheet.css" id="my-stylesheet">
```

Note: The `load` event fires once the stylesheet and all of its imported content has been loaded and parsed, and immediately before the styles start being applied to the content.

Preload examples

You can find a number of `<link rel="preload">` examples in Preloading content with `rel="preload"`.

Technical summary

Content categories	Metadata content. If `itemprop` is present: Flow content and phrasing content.
Permitted content	None, it is an empty element.
Tag omission	As it is a void element, the start tag must be present and the end tag must not be present
Permitted parents	Any element that accepts metadata elements. If itemprop is present: any element that accepts phrasing content.
Implicit ARIA role	`link` with `href` attribute
Permitted ARIA roles	No `role` permitted
DOM interface	`HTMLLinkElement`