

Chapter 12: Linking Resources

Attribute	Details
<code>charset</code>	Specifies the character encoding of the linked document
<code>crossorigin</code>	Specifies how the element handles cross origin requests
<code>href</code>	Specifies the location of the linked document
<code>hreflang</code>	Specifies the language of the text in the linked document
<code>media</code>	Specifies on what device the linked document will be displayed, often used with selecting stylesheets based on the device in question
<code>rel</code>	Required. Specifies the relationship between the current document and the linked document
<code>rev</code>	Specifies the relationship between the linked document and the current document
<code>sizes</code>	Specifies the size of the linked resource. Only when <code>rel="icon"</code>
<code>target</code>	Specifies where the linked document is to be loaded
<code>type</code>	Specifies the media type of the linked document
<code>integrity</code>	Specifies a base64 encoded hash (sha256, sha384, or sha512) of the linked resource allowing the browser to verify its legitimacy.

While many scripts, icons, and stylesheets can be written straight into HTML markup, it is best practice and more efficient to include these resources in their own file and link them to your document. This topic covers linking external resources such as stylesheets and scripts into an HTML document.

Section 12.1: JavaScript

Synchronous

```
<script src="path/to.js"></script>
```

Standard practice is to place JavaScript `<script>` tags just before the closing `</body>` tag. Loading your scripts last allows your site's visuals to show up more quickly and discourages your JavaScript from trying to interact with elements that haven't loaded yet.

Asynchronous

```
<script src="path/to.js" async></script>
```

Another alternative, when the Javascript code being loaded is not necessary for page initialization, it can be loaded asynchronously, speeding up the page load. Using `async` the browser will load the contents of the script in parallel and, once it is fully downloaded, will interrupt the HTML parsing in order to parse the Javascript file.

Deferred

```
<script src="path/to.js" defer></script>
```

Deferred scripts are like `async` scripts, with the exception that the parsing will only be performed once the HTML is fully parsed. Deferred scripts are guaranteed to be loaded in the order of declaration, same way as synchronous scripts.

<noscript>

```
<noscript>JavaScript disabled</noscript>
```

The `<noscript>` element defines content to be displayed if the user has scripts disabled or if the browser does not support using scripts. The `<noscript>` tag can be placed in either the `<head>` or the `<body>`.

Section 12.2: External CSS Stylesheet

```
<link rel="stylesheet" href="path/to.css" type="text/css">
```

The standard practice is to place CSS `<link>` tags inside the `<head>` tag at the top of your HTML. This way the CSS will be loaded first and will apply to your page as it is loading, rather than showing unstyled HTML until the CSS is loaded. The `type` attribute is not necessary in HTML5, because HTML5 usually supports CSS.

```
<link rel="stylesheet" href="path/to.css" type="text/css">
```

and

```
<link rel="stylesheet" href="path/to.css">
```

... do the same thing in HTML5.

Another, though less common practice, is to use an `@import` statement inside direct CSS. Like this:

```
<style type="text/css">
    @import("path/to.css")
</style>

<style>
    @import("path/to.css")
</style>
```

Section 12.3: Favicon

```
<link rel="icon" type="image/png" href="/favicon.png">
<link rel="shortcut icon" type="image/x-icon" href="/favicon.ico">
```

Use the mime-type `image/png` for PNG files and `image/x-icon` for icon (`*.ico`) files. For the difference, see [this SO question](#).

A file named `favicon.ico` at the root of your website will typically be loaded and applied automatically, without the need for a `<link>` tag. If this file ever changes, browsers can be slow and stubborn about updating their cache.

Section 12.4: Alternative CSS

```
<link rel="alternate stylesheet" href="path/to/style.css" title="yourTitle">
```

Some browsers allow alternate style sheets to apply if they are offered. By default they will not be applied, but usually they can be changed through the browser settings:

Firefox lets the user select the stylesheet using the View > Page Style submenu, Internet Explorer also supports this feature (beginning with IE 8), also accessed from View > Page Style (at least as of IE 11), but Chrome requires an extension to use the feature (as of version 48). The web page can also provide its own user interface to let the user switch styles.

(Source: the [MDN Docs](#))

Section 12.5: Resource Hint: dns-prefetch, prefetch, prerender

Preconnect

The preconnect relationship is similar to dns-prefetch in that it will resolve the DNS. However, it will also make the TCP handshake, and optional TLS negotiation. This is an experimental feature.

```
<link rel="preconnect" href="URL">
```

DNS-Prefetch

Informs browsers to resolve the DNS for a URL, so that all assets from that URL load faster.

```
<link rel="dns-prefetch" href="URL">
```

Prefetch

Informs the browsers that a given resource should be prefetched so it can be loaded more quickly.

```
<link rel="prefetch" href="URL">
```

DNS-Prefetch resolves only the domain name whereas prefetch downloads/stores the specified resources.

Prerender

Informs browsers to fetch and render the URL in the background, so that they can be delivered to the user instantaneously as the user navigates to that URL. This is an experimental feature.

```
<link rel="prerender" href="URL">
```

Section 12.6: Link 'media' attribute

```
<link rel="stylesheet" href="test.css" media="print">
```

Media specifies what style sheet should be used for what type of media. Using the print value would only display that style sheet for print pages.

The value of this attribute can be any of the mediatype values (similar to a CSS media query).

Section 12.7: Prev and Next

When a page is part of a series of articles, for instance, one can use prev and next to point to pages that are coming before and after.

```
<link rel="prev" href="http://stackoverflow.com/documentation/java/topics">
```

```
<link rel="next" href="http://stackoverflow.com/documentation/css/topics">
```

Section 12.8: Web Feed

Use the rel="alternate" attribute to allow discoverability of your Atom/RSS feeds.

```
<link rel="alternate" type="application/atom+xml" href="http://example.com/feed.xml" />
<link rel="alternate" type="application/rss+xml" href="http://example.com/feed.xml" />
```