

HTML <script> Tag for JavaScript Code

#script# for dynamic JavaScript code. #script# description, syntax, usage, attributes and examples. HTML-5.com is a great guide for web developers. [TV Series & Actors and Actresses](#). Follow [TV Series](#) and [HTML 5](#) on Google+.

[HTML-5.com](#) > [itemscopehttp://data-vocabulary.org/BreadcrumbHTML 5](#) > [itemscopehttp://data-vocabulary.org/BreadcrumbHTML Tags](#) > [itemscopehttp://data-vocabulary.org/BreadcrumbHTML <script> Tag for JavaScript Code](#)

> [itemscopehttp://data-vocabulary.org/BreadcrumbExamples](#)

The <script> Tag in HTML 5

```
<script type="text/javascript"> window.onload = function() { var elem = document.getElementById('noscript1'); elem.textContent = ""; var button = document.createElement('button'); button.textContent = "Click Me!"; var output = document.createElement('span'); button.onclick = function() { output.textContent = 'Button got clicked!'; return false; } elem.appendChild(button); elem.appendChild(document.createElement('br')); elem.appendChild(output); } </script>
```

Scripts failed or are disabled. Please make sure scripting is enabled.

This is an actual working example of the [<script> tag example code](#) below.

[back to top](#)

<script> Tag Syntax

Rules for coding HTML `script` elements

```
<a mode="pre" href=" ../head-tag/index.html#syntax"><head></a>
  <a mode="pre" href=" ../html-tag-list.html#metadata-content-model">... metadata content ...<
  <a mode="pre" href=" ../title-tag/index.html#syntax"><title></a><a mode="pre" href=" ../de
  ...
  <b mode="pre"><script type="text/javascript" src="..."></script></b>
  <b mode="pre"><script type="text/javascript"></b>
    <a mode="pre" href=" ../.. /javascript/">... JavaScript code ...</a>
  <b mode="pre"></script></b>
  ...
</head>
```

<script> Content Model

Contents of the <script> Tag

How to code a script element with an external script resource

```
<script <a mode="pre" href="#type-attribute">type="text/javascript"</a> <a mode="pre" href="
```

1. Inside an element where [metadata content](#) or [phrasing content](#) is allowed, code one or more optional **script** elements.
2. Begin the **script** element with a starting `<script>` tag. The element name uses lower case letters and should be in the [HTML namespace](#), which it will pick up automatically from the [xmlns attribute](#) on the `<html>` tag.
3. Specify the URL of the external JavaScript code in the [src attribute](#).
4. Include any other [attributes of the <script> tag](#) as appropriate.

5. End the **script** element with a matching `</script>` closing tag.

How to code a script element with inline script code

```
<script <a mode="pre" href="#type-attribute">type="text/javascript"</a> >
  <a mode="pre" href="../../../javascript/">... JavaScript code ...</a>
</script>
```

1. Inside an element where [metadata content](#) or [phrasing content](#) is allowed, code one or more optional **script** elements.
2. Begin the **script** element with `<script>`. The element name uses lower case letters and should be in the [HTML namespace](#), which it will pick up automatically from the `xmlns` attribute on the `<html>` tag.
3. Include any other [attributes of the <script> tag](#) as appropriate.
4. Include the JavaScript code between the starting and ending `<script>` tags.
5. End the **script** element with a matching `</script>` closing tag.

[back to top](#)

<script> Tag Attributes

Attributes of the <script> tag

global attributes	In addition to the personal attributes of the <code><script></code> tag below, any of the common HTML attributes can also be coded.
<code>async="async"</code>	Sets the value of the <code><script async></code> boolean attribute to <code>true</code> . Omitting it sets to <code>false</code> .
<code>charset</code>	
<code>defer="defer"</code>	Sets the value of the <code><script defer></code> boolean attribute to <code>true</code> . Omitting it sets to <code>false</code> .
<code>language</code>	deprecated - use the type attribute instead
<code>src</code>	<p>a URI reference that resolves to the URL of a resource with JavaScript code</p> <p>Use percent escape codes as explained in the URL-encoding tutorial for any special characters in the URI reference.</p> <p>If the value of the <code>src</code> attribute resolves to an HTTP URL, it may contain any of the following components:</p> <ol style="list-style-type: none"> 1. protocol scheme, typically <code>http:</code> or else <code>https:</code> 2. username, followed by an "@" 3. host name or IP address 4. port number, which defaults to 80 for the <code>http</code> scheme and 443 for the <code>https</code>

	<p>5. absolute or relative path 6. search query, indicated by "?"</p> <p>If the protocol scheme, username, host name/IP address and port number are omitted the default is the current host - the same server as the base of the current document. If the path starts with a slash <code>/...</code>, it is an absolute path from the document root directory (AKA "<dfn>web root</dfn>") on the server. A relative path will be resolved relative to the base of the current document.</p>
<p><code>type</code></p>	<p><code>type="text/javascript "</code> (default) <code>type="application/ecmascript "</code> <code>type="application/javascript "</code> <code>type="application/x-ecmascript "</code> <code>type="application/x-javascript "</code> <code>type="text/ecmascript "</code> <code>type="text/javascript1.0 "</code> <code>type="text/javascript1.1 "</code> <code>type="text/javascript1.2 "</code> <code>type="text/javascript1.3 "</code> <code>type="text/javascript1.4 "</code> <code>type="text/javascript1.5 "</code> <code>type="text/jscript "</code> <code>type="text/livescript "</code> <code>type="text/x-ecmascript "</code> <code>type="text/x-javascript "</code> <code>type="text/javascript;e4x=1 "</code></p>

[back to top](#)

<script> Tag Examples

Examples of the `script` tag in HTML 5

```
<script type="text/javascript" src="external-script-file.js"/>
```

```
<script type="text/javascript">
<!--
    window.onload=alert("Hello JavaScript World!")
// -->
</script>
```

The [HTML comments](#) in this example do *not* comment out the JavaScript code. They will only hide the code from browsers without JavaScript support or with scripts disabled. In some cases, browsers give the user an option to disable scripts. This HTML code pattern was designed to allow hiding content inside a **script** element from any browsers that might otherwise treat that content as [phrasing content](#) rather than as JavaScript code.

Script with fallback

Since there are various mechanisms that can disable scripting without the browser being aware of it, the recommended way to provide fallback content for a script is to change from noscript content to scripted content in the script itself, like this:

```
<head>
...
<script type="text/javascript">
window.onload = function() {
  var elem = document.getElementById('noscript1');
  elem.textContent = "";
  var button = document.createElement('button');
  button.textContent = "Click Me!";
  var output = document.createElement('span');
  button.onclick = function() { output.textContent = 'Button got clicked!';
  return false; }
  elem.appendChild(button);
  elem.appendChild(document.createElement('br'));
  elem.appendChild(output);
}
</script>
...
</head>
<body>
...
<div id="script-tag-demo">
<p id="noscript1">Scripts failed or are disabled. Please make sure scripting
is enabled.</p>
</div>
...
</body>
```

[back to top](#)

Changes in HTML 5 - <script> Tag

What's new in HTML 5

The `async="async"` attribute has been added.

Differences between HTML 5 and earlier versions of HTML

The following attributes should *not* be coded on the <script> tag because they either have been deprecated or were never officially supported:

- `language` - use the `type` attribute instead

The [2000-2010 Recommendations from the W3C HTML Working Group](#) defined the [HTML namespace](#) for the **script** element type name along with the names of all [HTML element types](#). In older (pre-2000) [versions of HTML](#), element type names were not associated with a namespace.

[back to top](#)

THE END