

HTML <iframe> for Video or Other Inline Content

#iframe# creates an area, possibly with scroll bars, with content loaded from a separate file. #iframe# 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/Breadcrumb](#)
[HTML <iframe> for Video or Other Inline Content](#)

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

The <iframe> Tag in HTML 5

The <iframe> tag defines an inline frame that loads content from another HTML document and displays it as part of the current page. It is one of the [embedded content tags in HTML](#). An **iframe** works almost like an image created with the [img element](#), except that the content in the external resource is HTML code rather than binary image data.

Some possible uses of inline frames are:

- to embed content from another site, such as in the YouTube embed example below
- for an area that is common to multiple web pages, such as a footer, as described below

Using an <iframe> to embed a YouTube video

Note: For the following example to work in HTML 5 without Flash, you must "Join the HTML5 Trial" at <http://www.YouTube.com/html5>.

```
<iframe id="embed-youtube-video-demo" class="youtube-player" width="640" height="480" style="margin-bottom: 15px" src="http://www.YouTube.com/embed/GGT8ZCTBoBA"></iframe>
```

This is an actual working example of the [Embedded YouTube Video example code](#) below.

Using an <iframe> for a common footer

One use of the <iframe> tag is for common elements that appear on multiple pages of a web site. For example, this can be used to include a common footer, which can be accomplished by following these instructions:

1. Create an XSL style sheet with a template for the footer.
2. Include an <iframe> tag in the footer template.
3. Reference the XSL style sheet with an [xml-stylesheet instruction](#) in every web page where the footer is to appear.

Although the footer HTML code could be included inline in the style sheet without the <iframe> tag, loading the footer from a separate document allows the footer to be cached by browsers separately from the site templates. A footer in a separate document can also be used on pages without the template simply by adding the <iframe> tag to those documents.

How to change the web page footer(s) depends on whether the whole footer or just part of it needs to be changed:

- To change which footer is used, change the value of the `src` attribute in the `<iframe>` tag in the template.
- To change the content of the footer, edit the HTML document that the `src` attribute of the `<iframe>` tag references.

More advanced web site templates can be created where different sets of pages use different style sheets, and those style sheets can include another set of higher-level templates that apply to all pages across the entire web site.

[back to top](#)

<iframe> Tag Syntax

Rules for coding HTML iframe elements

```
<a mode="pre" href="../body-tag/index.html#syntax"><body></a>
...
<a mode="pre" href="../html-tag-list.html#flow-content-model">... flow content expected ...</a>
<b mode="pre"><iframe></b>
  <a mode="pre" href="../../definitions/index.html#flow-content">... flow content ...</a>
  <b mode="pre"></iframe></b>
...
</body>
```

Rules for coding HTML `iframe` elements

Make sure you understand the [difference between a tag and element](#) and are familiar with the [definitions](#) of [namespace](#) and other [HTML terms](#).

1. Include a **iframe** element where [flow content](#) is expected.
2. Begin the **iframe** element with a starting `<iframe>` 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. Inside the `<iframe>` starting tag, include a [src attribute](#) with the URL of a source document for the iframe content, a [srcdoc attribute](#) with the HTML for the iframe content or both, in which case the `srcdoc` will be used if supported and the `src` document will be used as fallback content.
4. Inside the **iframe** element, between the starting `<iframe>` tag and the ending `</iframe>` tag, code the inner HTML [flow content](#).
5. End the **iframe** element with a matching `</iframe>` closing tag.

<iframe> Content Model

Contents of the iframe element

Content: *Empty*. All properties are coded using [attributes](#).

The `iframe` element is an [empty element](#), with both a start tag and an end tag, *not* a [void element](#). However, while it can have [text content](#) in documents being parsed as HTML, it must be empty in documents being parsed as xHTML or pure XML. Therefore in [polyglot documents](#) as recommended on this site, [HTML comments](#) should be the only code inside an **iframe** element.

[back to top](#)

<iframe> Tag Attributes

Attributes of the <iframe> tag

<pre>Attributes <script> var cell = document.createElement("th"); document.getElementById("attribute- headings").appendChild(cell); cell.appendChild(document.createTextNode("Supported")); </script></pre>	
global attributes	<p>In addition to the personal attributes of the <iframe> tag below, any of the common HTML attributes can also be coded. <script> var cell = document.getElementById("global-attributes").insertCell(-1); cell.appendChild(document.createTextNode("Yes")); </script></p>
width , height	<p>the width and height of the internal frame, respectively</p> <p>The contents of the iframe as a whole are <i>not</i> resized based on these dimensions like an image would be, although the HTML tags in the referenced source document could resize portions of its contents. <script> var cell = document.getElementById("width-height-attributes").insertCell(-1); elem = document.createElement("iframe"); cell.appendChild(document.createTextNode("width in elem ? "Yes" : "No")); </script></p>
name	<p>the name of the nested browsing context <script> var cell = document.getElementById("name-attribute").insertCell(-1); elem = document.createElement("iframe"); cell.appendChild(document.createTextNode("name in elem ? "Yes" : "No")); </script></p>
sandbox	<p>a list of permissions for the HTML code in the document referenced by the src attribute, separated by spaces.</p> <p>sandbox=" "</p> <p>When the sandbox attribute is coded without any permissions, the referenced source document is restricted from performing any of the restricted actions.</p>

```
Attributes <script> var cell =  
document.createElement("th");  
document.getElementById("attribute-  
headings").appendChild(cell);  
cell.appendChild(document.createTextNode("Supported"));  
</script>
```

sandbox="allow-forms"

If one of the values in the `sandbox` attribute is `"allow-forms"`, then forms are allowed in the referenced source document.

sandbox="allow-scripts"

If one of the values in the `sandbox` attribute is `"allow-scripts"`, then scripts are allowed in the referenced source document. This should *not* be used in conjunction with `allow-same-origin`, since then the script could simply remove the `sandbox` attribute to bypass the restrictions.

sandbox="allow-same-origin"

If one of the values in the `sandbox` attribute is `"allow-same-origin"`, then . This should *not* be used in conjunction with `allow-scripts`, since then the script could simply remove the `sandbox` attribute to bypass the restrictions. When this value is omitted, the content from the referenced source document is forced into a "sandboxed origin browsing context" that prevents it from accessing HTML content or browser cookies from the original document's origin.

`allow-same-origin` can be used to allow the HTML code in the referenced document to:

- access the DOM (HTML elements, etc.) of the original document, while preventing it from running scripts
- post data back to its originating site, while preventing it from doing other things such as opening pop-up windows

<pre>Attributes <script> var cell = document.createElement("th"); document.getElementById("attribute- headings").appendChild(cell); cell.appendChild(document.createTextNode("Supported")); </script></pre>	
	<p>sandbox="allow-top-navigation"</p> <p>If one of the values in the sandbox attribute is "allow-top-navigation", then target="_top" is allowed, which can be used in an HTML <a> tag to replace the current document.</p> <pre><script> var cell = document.getElementById("sandbox- attribute").insertCell(-1); elem = document.createElement("iframe"); cell.appendChild(document.createTextNode("sand in elem ? "Yes" : "No")); </script></pre>
<p>seamless="seamless"</p>	<p>The value "seamless" indicates that the content of the document referenced by the src attribute should appear as if the HTML code was included the document where the <iframe> tag is coded. This is essentially an HTML include statement, which eliminates the need for a separate <include> tag in HTML 5.</p> <p>Omitting the seamless attribute indicates that the content of the internal frame is to be put into a nested browsing context separate from the original document.</p> <pre><script> var cell = document.getElementById("seamless- attribute").insertCell(-1); elem = document.createElement("iframe"); cell.appendChild(document.createTextNode("seam in elem ? "Yes" : "No")); </script></pre>
<p>src</p>	<p>a URI reference that resolves to the URL of a document with the content to be displayed in the internal frame</p> <p>If the srcdoc attribute is also specified, this attribute will be ignored by HTML 5 browsers; it will only be used by older</p>

<pre>Attributes <script> var cell = document.createElement("th"); document.getElementById("attribute- headings").appendChild(cell); cell.appendChild(document.createTextNode("Supported")); </script></pre>	
	<p>browsers that do not support the <code>srcdoc</code> attribute.</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 is 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> scheme 5. absolute or relative path 6. search query, indicated by "?" <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> <pre><script> var cell = document.getElementById("src- attribute").insertCell(-1); elem = document.createElement("iframe"); cell.appendChild(document.createTextNode("src" in elem ? "Yes" : "No")); </script></pre>
<p><code>srcdoc</code></p>	<p>The <code>srcdoc</code> attribute specifies the HTML code for the content to be displayed in the internal frame. Although the code can be a full HTML document starting with a DOCTYPE declaration, some tags that would normally be required in a standalone</p>

<pre>Attributes <script> var cell = document.createElement("th"); document.getElementById("attribute- headings").appendChild(cell); cell.appendChild(document.createTextNode("Supported")); </script></pre>	
	<p>HTML document become optional, such as the <title> tag (and therefore the empty <head> tag, <html> tag and now parentless <body> tag). Therefore, the value of the <code>srcdoc</code> attribute can be minimized to something as simple as a single phrasing content element, such as a paragraph tag. <code><script> var cell = document.getElementById("srcdoc-attribute").insertCell(-1); elem = document.createElement("iframe"); cell.appendChild(document.createTextNode("srcdoc in elem ? "Yes" : "No")); </script></code></p>

[back to top](#)

<iframe> Tag Examples

Examples of the <iframe> tag in HTML 5

It is this easy to embed a YouTube video in HTML 5

```
<iframe class="youtube-player" width="640" height="480"
src="http://www.YouTube.com/embed/GGT8ZCTBoBA" ></iframe>
```

This is the "new" YouTube embed code. (The "old" YouTube embed code used the [<object> tag](#). The 640 x 480 pixel size for the iframe is determined as follows:

- 640 x 45 pixel black bar above video
- 640 x 360 pixel video
- 640 x 45 pixel black bar below video
- 640 x 30 pixel tool bar at bottom

[back to top](#)

Changes in HTML 5 - <iframe> Tag

What's new in HTML 5

The [sandbox](#), [seamless="seamless"](#) and [srcdoc](#) attributes have been added.

Differences between HTML 5 and earlier versions of HTML

The [2000-2010 Recommendations from the W3C HTML Working Group](#) defined the [HTML namespace](#) for the **iframe** 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