



Selecting Objects with JavaScript

This article describes the [W3C Selectors API](http://www.w3.org/tr/selectors-api.aspx) [<http://www.w3.org/tr/selectors-api.aspx>] and how it enables you to use Cascading Style Sheets (CSS) selectors to select objects in JavaScript applications.

- [Introduction](#)
- [Limiting the Scope of Selector Queries](#)
- [Specifying Multiple Selector Rules](#)
- [Namespace Support](#)
- [Related Topics](#)

Introduction

When used with CSS, selectors provide a convenient way to select groups of objects, and, as of Windows Internet Explorer 8, to do this with JavaScript. This article describes how the browser supports **W3C Selectors API** and how the API is used to select objects in JavaScript applications.

Note: **W3C Selectors API** is only available to documents that opt into Internet Explorer 8 Mode; for more information, see [Defining Document Compatibility](http://msdn.microsoft.com/en-us/library/cc288325(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/cc288325\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/cc288325(VS.85).aspx)] .

W3C Selectors API provides two methods for using selectors to select objects: [querySelector](http://msdn.microsoft.com/en-us/library/cc288169(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/cc288169\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/cc288169(VS.85).aspx)] , which selects the first object that matches your criteria; and [querySelectorAll](http://msdn.microsoft.com/en-us/library/cc304115(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/cc304115\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/cc304115(VS.85).aspx)] , which returns all objects that match your criteria. You can select among the document objects or those that descend from a common container object.

Limiting the Scope of Selector Queries

By default, **W3C Selectors API** queries include all objects in a document. The following code example returns a list of [DIV elements](http://msdn.microsoft.com/en-us/library/ms535240(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms535240\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms535240(VS.85).aspx)] that share the same class value.

[Copy Code](#)

```
<html>
<head>
  <!-- Set document compatibility mode to IE8Mode. -->
  <meta http-equiv="X-UA-Compatible" content="IE=8">
</head>

<body>

<script language="Javascript">
function doit()
{
  var oAllPs = document.querySelectorAll("div.detail");
  var sFound = ( oAllPs == null ) ?
    "No matches found" : allPs.length;
  alert( "Results: " + sFound );
}
</script>

<button onclick="doit();">Simple Selector Example</button>

<ul>
  <li><div class="header" name="first">Alpha</div></li>
  <ol>
    <li><div class="detail">One</div></li>
```

```
<li><div class="detail">Two</div></li>
<li><div class="detail">Three</div></li>
</ol>
<li><div class="header" name="second">Beta</div></li>
<ol>
  <li><div class="detail">Four</div></li>
  <li><div class="detail">Five</div></li>
  <li><div class="detail">Six</div></li>
</ol>
<li><div class="header" name="third" >Gamma</div></li>
<ol>
  <li><div class="detail">Seven</div></li>
  <li><div class="detail">Eight</div></li>
  <li><div class="detail">Nine</div></li>
</ol>
</ul>

</body>
</html>
```

You can limit the scope of a **W3C Selectors API** query to the child object of a specific container object; the query will only return objects contained by the limiting object, not the limiting object itself. The following example is similar to the preceding one, but it limits the query to the second header **div** element.

[Copy Code](#)

```
<html>
<head>
  <!-- Set document compatibility mode to IE8Mode. -->
  <meta http-equiv="X-UA-Compatible" content="IE=8">
</head>

<body>

<script language="Javascript">
function doit()
{
  var oLimit = document.querySelector("#second");
  var oAllPs = oLimit.querySelectorAll("div");
  var sFound = ( oAllPs == null ) ?
    "No matches found" : oAllPs.length;
  alert( "Results: " + sFound );
}
</script>

<button onclick="doit();">Simple Selector Example</button>

<ul>
  <li><div class="header" id="first">Alpha</div></li>
  <ol>
    <li><div class="detail">One</div></li>
    <li><div class="detail">Two</div></li>
    <li><div class="detail">Three</div></li>
  </ol>
  <li><div class="header" id="second">Beta</div></li>
  <ol>
    <li><div class="detail">Four</div></li>
    <li><div class="detail">Five</div></li>
    <li><div class="detail">Six</div></li>
  </ol>
  <li><div class="header" id="third" >Gamma</div></li>
  <ol>
    <li><div class="detail">Seven</div></li>
    <li><div class="detail">Eight</div></li>
    <li><div class="detail">Nine</div></li>
  </ol>
```

```
</ul>
</body>
</html>
```

Limiting a query to specific elements only limits the query's results. Query criteria can refer to objects outside the container of the limiting element. Consider the following example:

[Copy Code](#)

```
<html>
<head>
  <meta http-equiv="X-UA-Compatible" content="IE=8">
  <title>Limit Example 1</title>
</head>
<body>

  <p>
    <div name="div1"><span name="span1">Found</span></div>
    <div name="div2"><span name="span2">Not found</span></div>
  </p>

  <script>
  function Sample()
  {
    var oDiv = document.getElementById( "div1" );
    var oSpan = oDiv.querySelector( "p span" );
    var sResult = ( oSpan == null ) ?
      "No match found; check your query" : "The query returned " + oSpan.name;
    alert( sResult );
  }
  </script>
  <button onclick="Sample();">Run Test</button>
</body>
</html>
```

This sample includes a [P element](http://msdn.microsoft.com/en-us/library/ms535878(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms535878\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms535878(VS.85).aspx)] that contains two **DIV elements**, each containing a single [SPAN element](http://msdn.microsoft.com/en-us/library/ms535895(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms535895\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms535895(VS.85).aspx)]. Running this sample in Internet Explorer 8 will return the first **SPAN element**. Because the query is limited to the first **DIV element**, the results include only the first **SPAN element**, even though the criteria refer to the **P element** containing both **DIV elements**.

Specifying Multiple Selector Rules

To specify multiple selector rules in a selector query, separate each rule with commas. The following example specifies a query that selects **div** and **span** elements.

[Copy Code](#)

```
<html>
<head>
  <!-- Set document compatibility mode to IE8Mode. -->
  <meta http-equiv="X-UA-Compatible" content="IE=8">
</head>

<body>

<script language="Javascript">
function doIt()
{
  var oAllPs = document.querySelectorAll("div, span");
  var sFound = ( oAllPs == null ) ?
    "No matches found" : oAllPs.length;
  alert( "Results: " + sFound );
}
</script>
```

```
</script>

<button onclick="doit();">Simple Selector Example</button>

<ul>
  <li><span class="header" id="first">Alpha</span></li>
  <ol>
    <li><div class="detail">One</div></li>
    <li><div class="detail">Two</div></li>
    <li><div class="detail">Three</div></li>
  </ol>
  <li><span class="header" id="second">Beta</span></li>
  <ol>
    <li><div class="detail">Four</div></li>
    <li><div class="detail">Five</div></li>
    <li><div class="detail">Six</div></li>
  </ol>
  <li><span class="header" id="third">Gamma</span></li>
  <ol>
    <li><div class="detail">Seven</div></li>
    <li><div class="detail">Eight</div></li>
    <li><div class="detail">Nine</div></li>
  </ol>
</ul>

</body>
</html>
```

Important For security reasons, Internet Explorer 8 ignores the [:visited](http://msdn.microsoft.com/en-us/library/ms531181(VS.85).aspx) [http://msdn.microsoft.com/en-us/library/ms531181(VS.85).aspx] pseudo class selector in selector query criteria. If a selector query combines the **:visited** pseudoselected with other rules, the other rules are processed as usual. For example, if a selector query requests the **:visited** and [:link](http://msdn.microsoft.com/en-us/library/ms530787(VS.85).aspx) [http://msdn.microsoft.com/en-us/library/ms530787(VS.85).aspx] pseudo selectors, the query ignores **:visited** and returns a collection of all links, whether or not they have been visited.

Namespace Support

Because Internet Explorer 8 does not formally support XHTML documents, it does not support the namespace features (such as the **NSResolver** parameter) of the **W3C Selectors API** specification.

Related Topics

- [Defining Document Compatibility](http://msdn.microsoft.com/en-us/library/cc288325(VS.85).aspx) [http://msdn.microsoft.com/en-us/library/cc288325(VS.85).aspx]

Tags:



Community Content

Nice feature!

Last Edit 5:07 PM by netsi1964

I am happy to see that you will support a the <mtps:InstrumentedLink NavigateUrl="http://www.w3.org/tr/selectors-api" runat="server" xmlns:mtps="http://msdn2.microsoft.com/mtps">W3C Selectors API</mtps:InstrumentedLink>! Though for a while I will go for JQuery in my webpages. That is until the other big browsers will start supporting that API also - but I am happy for this choice of yours! :-)

Tags: