

Selectors API :
Windows® Internet Explorer® 8
Beta 1 for Developers



Web 作業の操作性を向上

2008 年 3 月

詳細の問い合わせ先 (報道関係者専用):
Rapid Response Team
Waggener Edstrom Worldwide
(503) 443-7070
rrt@waggeneredstrom.com

このドキュメントに記載されている情報は、このドキュメントの発行時点におけるマイクロソフトの見解を反映したものです。マイクロソフトは市場の変化に対応する必要があるため、このドキュメントの内容に関する責任をマイクロソフトは問われないものとします。また、発行日以降に発表される情報の正確性を保証できません。

このドキュメントに記載された内容は情報の提供のみを目的としています。明示、黙示または法律の規定にかかわらず、これらの情報についてマイクロソフトはいかなる責任も負わないものとします。

お客様ご自身の責任において、適用されるすべての著作権関連法規に従ったご使用を願います。このドキュメントのいかなる部分も、米国 Microsoft Corporation の書面による許諾を受けることなく、その目的を問わず、どのような形態であっても、複製または譲渡することは禁じられています。ここでいう形態とは、複写や記録など、電子的な、または物理的なすべての手段を含みます。ただしこれは、著作権法上のお客様の権利を制限するものではありません。

マイクロソフトは、このドキュメントに記載されている内容に関し、特許、特許申請、商標、著作権、またはその他の無体財産権を有する場合があります。別途マイクロソフトのライセンス契約上に明示の規定のない限り、このドキュメントはこれらの特許、商標、著作権、またはその他の無体財産権に関する権利をお客様に許諾するものではありません。

別途記載されていない場合、このソフトウェアおよび関連するドキュメントで使用している会社、組織、製品、ドメイン名、電子メール アドレス、ロゴ、人物、場所、出来事などの名称は架空のものです。実在する商品名、団体名、個人名などとは一切関係ありません。

© 2008 Microsoft Corp. All rights reserved.

Microsoft、Windows、Windows Vista、Windows Server、ActiveX、Active Directory、Internet Explorer、Internet Explorer のロゴ、MSN のロゴは、米国 Microsoft Corp. の米国およびその他の国における登録商標または商標です。

記載されている会社名、製品名には、各社の商標のものもあります。

概要

セレクタとは、Web ページ内のタグ (要素) を検索し、選択するためのクエリ言語のことです。セレクタは、プロパティ適用先の要素のグループを選択する場合に CSS で一般的に使用されています。次に例を示します。

```
Selector
{
    プロパティ: 値;
    プロパティ 2: 値;
}
```

Windows Internet Explorer 7 には、CSS とは独立してセレクタを実行する方法はありません。Selectors API を使用することによって、Web 開発者が CSS セレクタの要素を迅速に参照する機能と柔軟性が得られます。

```
var oneElement = document.querySelector('Selector');
var elementCollection = document.querySelectorAll('Selector');
```

Selectors API では、非線形要素トラバーサルとフィルタのパフォーマンスが向上し、独立した要素参照と DOM の手動スクリプト走査の代替方法を検討する場合に最適です。Selectors API では、コードのサイズも削減できるため、ダウンロードの時間、開発のコストと期間、および長期的な保守コストも削減されます。

たとえば、多くの Web サイトが同時に複数のテキスト入力コントロールに対してフォーム検証を実行するとします。Selectors API では、処理する要素の検索と収集に必要なコードを大幅に簡便化できます。例として、各種フォーム コントロールを取得し、コントロールが空でないことを検証するコードは次のようになります。

Selectors API の実装前	Selectors API の実装後
<pre>function doValidation() { // Retrieve all the required elements by ID var name = document.getElementById('fld_name'); var empnum = document.getElementById('fld_emp'); var ssn = document.getElementById('fld_ssn'); var ccnum = document.getElementById('fld_ccNum'); var bribe = document.getElementById('fld_cost'); // Set the flag to false by default var missingRequiredField = false; // Validate each of the elements if (name.childNodes[1].value == "") missingRequiredField = true; if (empnum.childNodes[1].value == "") missingRequiredField = true; if (ssn.childNodes[1].value == "") missingRequiredField = true; if (ccnum.childNodes[1].value == "") missingRequiredField = true; if (bribe.childNodes[1].value == "") missingRequiredField = true; }</pre>	<pre>function doValidation () { // Retrieve the required elements by using Selectors // Selects all the form fields with 'required' classes var reqs = document.querySelectorAll(".required"); // Set the flag to false by default var missingRequiredField = false; // Validate that the form data is not empty for (var i = 0; i < reqs.length; i++) { if (reqs[i].value == "") missingRequiredField = true; } }</pre>

(この例は、少し不自然ですが、要点をわかりやすく示しています。)

Internet Explorer 8 Beta 1 for Developers における Selectors API の実装は、『W3C ワーキング ドラフト (英語)』(<http://www.w3.org/TR/selectors-api/>) に準拠しています。

互換性 : Internet Explorer 7 から変更された動作

既定では、Internet Explorer 8 Beta 1 for Developers ですべての Web ページに対して Selectors API を使用できません。Selectors API は、標準に準拠した解析に依存して CSS セレクタを処理するため、IE8 Standards モードからオプトアウトしたドキュメントに対して使用できません (Quirks モードまたは IE7 Standards モードでは動きません)。IE8 Standards モードを使用しないようにするには、次の META 要素を Web ページの HEAD 要素の最上位に配置します。

```
<meta http-equiv="X-UA-Compatible" content="IE=7">
```

Internet Explorer 8 では、DOM L3 XPath クエリ API 仕様がサポートされません。ただし、Selectors API は DOM L3 XPath クエリ API 仕様と類似しています。Selectors API の利点のほとんどは、XPath をサポートするブラウザを実装することにより実現できます。ただし、ブラウザ内には両方のクエリ言語用の場所が存在しています。XPath に対する Selectors API をサポートする主な理由は、多くのクライアント側開発者が CSS を数年間使用してセレクタの構文の知識を持ち、理解しているものの、XPath の知識は持っていないことにあります。

機能の詳細

W3C では、一致するものをすべて返すフォームと最初に一致するもののみを返すフォームという Selectors API の 2 つのフォームが定義されています。

- **querySelector** : ドキュメント内で最初に見つかった一致する要素を返します (深さ優先、前順走査)。指定されたセレクタと一致する要素がない場合、null 値が返されます。
- **querySelectorAll** : ドキュメント順で一致するすべての要素のコレクションを返します。指定されたセレクタと一致する要素がない場合、空のコレクションが返されます。

querySelectorAll によって返されるコレクションは静的です。つまり、要素は DOM から削除されるか DOM に追加されるため、コレクションは更新されず、そのサイズは変化しません。コレクションに含まれる要素のポインタは、対応する要素が DOM から削除されていても存在します。

範囲を指定した要素の検索

Selectors API には、ドキュメント全体の検索と要素の範囲を指定した検索の両方を実行する機能が用意されています。ドキュメント全体の検索を実行するには、ドキュメント オブジェクトで次のように Selectors API を呼び出します。

```
<html>
<body>
  <p class="para">Paragraph 1</p>
  <div class="para">Block 2</div>
  <div><p class="para">Some text</p></div>
  <code class="para">Code sample here</code>
  <script>
    var allMyParagraphs = document.querySelectorAll('.para');
    // allMyParagraphs には、段落の場所に関係なく、ドキュメント内の
    // すべての段落が含まれる。
  </script>
</body>
</html>
```

要素の下位要素のみを検索するように制限するには、該当する要素で Selectors API を次のように呼び出します。

```
<html>
<body>
<div>
  <span id="example">
    <a href="somewhere.htm">Go get it!</a>
  </span>
</div>
<script>
  var span = document.querySelector("#example");
  var a = span.querySelector("div a");
  // 'a' will select the anchor.
  var div = span.querySelector("div");
  // 'div' will be null because the div is not a descendent of 'span'
</script>
</body>
</html>
```

要素に対して Selectors API のメソッドを呼び出す場合、返される要素の範囲がその要素の下位要素に限定されることに注意してください (指定されたセレクタの範囲ではありません)。サンプルの例では、指定されたセレクタが DIV 要素を参照していても変数 'a' はアンカーを返します。これは、指定された要素の範囲外です。また、範囲を指定した検索では、Selectors API のメソッドを呼び出した要素は返されません。

複数のセレクタの一括処理

CSS セレクタと同じように、Selectors API ではコンマ区切り一覧として渡されている複数のセレクタを処理できます。セレクタのグループ内では、各セレクタが結果を絞り込む論理的な AND の意味ではなく、検索範囲を広げる論理的な OR の意味で結合されています。

```
<html>
<body>
<div>
  <span class="one">One span</span>
  <span class="two">Two span</span>
</div>
<script>
  var spans = document.querySelectorAll('div span.one, div span.two');
  // The spans collection contains both spans.
</body>
</html>
```

名前空間のサポート

Internet Explorer 8 における Selectors API の実装では、W3C によって定義された NSResolver パラメータはサポートされません。このパラメータ (関数ポインタ) は、指定されたセクタ内に見つかったセクタ名前空間プレフィックスを解決する場合に XHTML ドキュメントでを使用することを目的としています (W3C 仕様の例を参照)。

:visited および :link 擬似クラス セクタのセキュリティ

W3C 仕様からの引用 : "履歴の窃盗は潜在的なセキュリティの問題です。[Selectors] 内に :visited 擬似クラスがあるためです。:visited を使用すると、[CSS21] を使用して参照したリンクをクエリできます。この仕様では問題は発生しておらず、:visited は他の方法によって悪用される可能性があるため、ユーザー エージェントは対策を講じる必要はありません。"

この文章とは矛盾しますが、Internet Explorer 8 では Selectors API で履歴の窃盗の問題を軽減する対策を講じています。Internet Explorer 8 では、セクタに **:visited** 擬似クラスまたは **:link** 擬似クラスが含まれている場合、クエリの処理中にブラウザはそのセクタを無視します。セクタのグループが指定されている場合、:visited 擬似クラスまたは :link 擬似クラスを含むセクタのみが無視されます。グループ内の他のセクタは、次のように照合の候補となります。

```
// The second selector in this group of selectors will be rejected, the first and
// the third selector will continue to apply.
document.querySelectorAll("two, a:link, .myClass");
```

Selectors API では、CSS で使用されるすべてのセクタの組み合わせを使用できます。このことでスクリプト コードが減少し、要素の参照が迅速化するため、開発者は従来の DOM ナビゲーションやフレームワーク コードよりも効率的な Web アプリケーションを開発できます。