

Chapter 3

HTML要素の指定と
CSSプロパティの操作

3-1 Webブラウザの種類を知る

jQueryをはじめとするJavaScriptライブラリを用いるのは、さまざまなWebブラウザの種類に対応するためというのが目的の1つです。とはいえ、Webアプリケーションの内容によっては、jQueryだけではカバーできない部分もあります。

そうすると、アプリケーションの側でアクセスされたWebブラウザの種類を知って、それに対応しなければならない場合もあります。そこでjQueryでは、シンプルな方法でWebブラウザの種類を知る方法が提供されています。

jQueryでWebブラウザの種類を知る方法

<code>jQuery.browser.msie</code>	Internet Explorer なら true
<code>jQuery.browser.mozilla</code>	Firefox なら true
<code>jQuery.browser.opera</code>	Opera なら true
<code>jQuery.browser.safari</code>	Safari なら true

※ \$.browser. ~も同じ

■ 使用例 Webブラウザの種類を画面に表示

使用例をリスト3.1に示します。これはtrueだったものの名前を画面に表示するものです。そしてInternet Explorerでの実行結果を図3.1、Firefoxでの実行結果を図3.2に示します。Webブラウザごとに画面表示が異なることを確認できます。

■ リスト 3.1 jQueryでWebブラウザの種類を知る (HTML)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ja" lang="ja">
  <head>
    <title>Web ブラウザの種類を知る (jQuery)</title>
    <script type="text/javascript" src="jquery-1.2.3.min.js"></script>
  </head>
  <body>
    <h3>Web ブラウザの種類を知る </h3>
    <script type="text/javascript">// 
      var browser = 'jQueryでは判別不能';
      if ( jQuery.browser.msie ) {
        browser = 'Internet Explorer';
      } else if ( jQuery.browser.mozilla ) {
        browser = 'Firefox';
      } else if ( jQuery.browser.opera ) {
        browser = 'Opera';
      } else if ( jQuery.browser.safari ) {
        browser = 'Safari';
      }
      document.write( 'このWeb ブラウザは ' + browser + ' です。' );
    // ]]&gt;&lt;/script&gt;
  &lt;/body&gt;
&lt;/html&gt;</pre>
</div>
<div data-bbox="575 463 675 562" data-label="Image">
<img alt="Screenshot of Internet Explorer browser showing the page content: 'このWebブラウザは Internet Explorer です。'"/>
</div>
<div data-bbox="687 463 777 493" data-label="Caption">
<p>■ 図 3.1<br/>Internet Explorer の場合</p>
</div>
<div data-bbox="796 463 896 562" data-label="Image">
<img alt="Screenshot of Firefox browser showing the page content: 'このWebブラウザは Firefox です。'"/>
</div>
<div data-bbox="712 547 801 563" data-label="Caption">
<p>■ 図 3.2 Firefox の場合</p>
</div>
<div data-bbox="564 603 818 622" data-label="Section-Header">■ 応用例 Webブラウザの種類に合わせた対応</div>
<div data-bbox="561 638 918 702" data-label="Text">
<p>この方法の応用例をリスト3.2に示します。これは画面表示直後に、アクセスされたWebブラウザがInternet ExplorerならforIE()を、FirefoxならforFF()を、というようにWebブラウザの種類に合わせた対応をするというものです。</p>
</div>
<div data-bbox="561 721 828 738" data-label="Section-Header">■ リスト 3.2 Webブラウザの種類に対応するプログラミングの応用例 (JavaScript)</div>
<div data-bbox="564 742 908 909" data-label="Text">
<pre>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"&gt;
&lt;html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ja" lang="ja"&gt;
  &lt;head&gt;
    &lt;title&gt;Web ブラウザの種類を知る (jQuery)&lt;/title&gt;
    &lt;script type="text/javascript" src="jquery-1.2.3.min.js"&gt;&lt;/script&gt;
    &lt;script type="text/javascript"&gt;// <![CDATA[
      function forIE() {
        // Internet Explorer の場合
      }

      function forFF() {</pre>
</div>
<div data-bbox="832 895 911 911" data-label="Text">
<p>以降、次ページへ続く →</p>
</div>
<div data-bbox="19 951 44 969" data-label="Page-Footer">036</div>
<div data-bbox="943 951 973 969" data-label="Page-Footer">037</div>
```

→前ページより続く

```

// Firefoxの場合
}

function forOpera() {
// Operaの場合
}

function forSafari() {
// Safariの場合
}

// 画面表示直後に以下を実行
window.onload = function() {
if ( jQuery.browser.msie ) {
forIE(); // Internet Explorerの場合
} else if ( jQuery.browser.mozilla ) {
forFF(); // Firefoxの場合
} else if ( jQuery.browser.opera ) {
forOpera(); // Operaの場合
} else if ( jQuery.browser.safari ) {
forSafari(); // Safariの場合
}
}
// ]]></script>
</head>
<body>
..... 画面に表示される部分 .....
</body>
</html>

```

3-2 要素の指定

jQueryプログラミングで最も多く用いられるのが、これから紹介する要素の指定です。これは、どの要素にCSSプロパティを追加すればよいかを指定する場合などに必要となります。このとき注意しなければならないのは、記述は1つしかなくても、実際には複数の要素が指定されている場合があるということです。

要素を指定する方法

<code>\$('CSS セレクタ ' [, 基準とする要素])</code>	CSSセレクタを用いる
<code>\$(要素オブジェクト [, 基準とする要素])</code>	要素オブジェクトを用いる
<code>\$('要素を表す文字列 ' [, 基準とする要素])</code>	文字列で要素を表す

※jQuery(～)も同じ。[基準とする要素]は省略可能だが、これが記述されたときは、その要素に含まれているものだけが指定される。

3-2-1 CSSセレクタによる指定

●CSSセレクタとは

CSSセレクタは、CSSプロパティを適用する要素を示すものです。たとえば、幅 100ピ

クセル、高さ 100ピクセルのdiv要素は以下のように記述されますが、このときのdivの部分がCSSセレクタです。

```

div { ←divの部分がCSSセレクタ
width:100px;
height:100px;
}

```

CSSセレクタの特徴は、たとえば div と記述されている場合はすべてのdiv要素を適用の対象とするというように、複数の要素の指定をシンプルに表現できることです。これは便利である一方で、Webページの内容が複雑になる場合に、思わぬところでCSSプロパティが適用されてしまったり、それを適用すべきところで適用されていなかったりという間違いを生む原因になりかねませんので、きちんとした指定を心がけなくてはなりません。

●要素の親子関係

要素を指定する場合に知っておくと便利なのは、要素の親子関係という考え方です。これは要素がHTML全体からみてどの位置にあるかを把握するために用いられます。

たとえばリスト3.3のようなHTMLがある場合(表示例は図3.3)、まず図3.4のような図を思い浮かべます。ここからわかるのは、<body>～</body>の範囲に<div id="content">を1つ含んでいて、この要素はその中にさらに2つの<div>を含んでいるということです。他の要素についても、その中に別の要素を含んでいるか、逆に別の要素に含まれているという関係があります。

ここでは、別の要素を含んでいる要素を親要素、親要素に含まれる要素を子要素と呼ぶことにします。図3.4でいえば、<div id="content">を含む<body>が親要素で、<div id="content">のほうの子要素ということになります。いわば各要素の位置関係を家系図に見立てて考えるということです。

そのように図3.4を眺めると、兄弟、先祖、子孫などの関係もみえてきます。たとえばはすべて兄弟です。<body>は他のすべての要素の先祖にあたります。逆に<h3>や<p>は<body>の子孫にあたります。別の書籍などではこれらを親ノード、子ノードなどと呼んでいる場合もあります。

■リスト3.3 要素の位置関係を示すためのサンプル (HTML)

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ja" lang="ja">
<head>
<title>CSS セレクタによる要素の指定 (jQuery)</title>
<style type="text/css"> /*  */
..... CSS プロパティは省略 .....
/*  */</style>
</head>
<body>

```

以降、次ページへ続く →

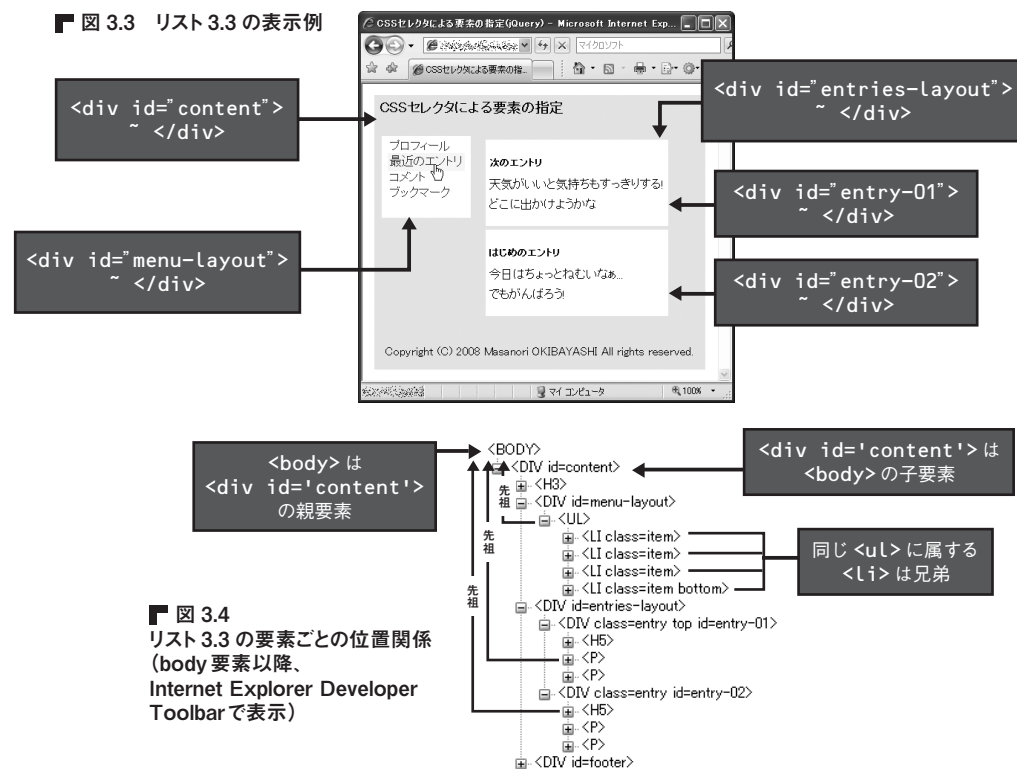
→前ページより続く

```

<div id="content">
  <h3>CSS セレクタによる要素の指定 </h3>
  <div id="menu-layout">
    <ul>
      <li class="item"><a href="profile.html">プロフィール</a></li>
      <li class="item"><a href="recent.html">最近のエントリ</a></li>
      <li class="item"><a href="comments.html">コメント</a></li>
      <li class="item bottom"><a href="bookmark.html">ブックマーク</a></li>
    </ul>
  </div>
  <div id="entries-layout">
    <div id="entry-01" class="entry top">
      <h5>次のエントリ</h5>
      <p>天気がいいと気持ちもすっきりする!</p>
      <p>どこに出かけようかな</p>
    </div>
    <div id="entry-02" class="entry">
      <h5>はじめのエントリ</h5>
      <p>今日はちょっとねむいなあ...</p>
      <p>でもがんばろう!</p>
    </div>
  </div>
  <div id="footer">Copyright (C) 2008 Masanori OKIBAYASHI All rights reserved.</div>
</div>
</body>
</html>

```

■ 図 3.3 リスト 3.3 の表示例



■ 図 3.4
リスト 3.3 の要素ごとの位置関係 (body 要素以降、Internet Explorer Developer Toolbar で表示)

●jQuery の対応

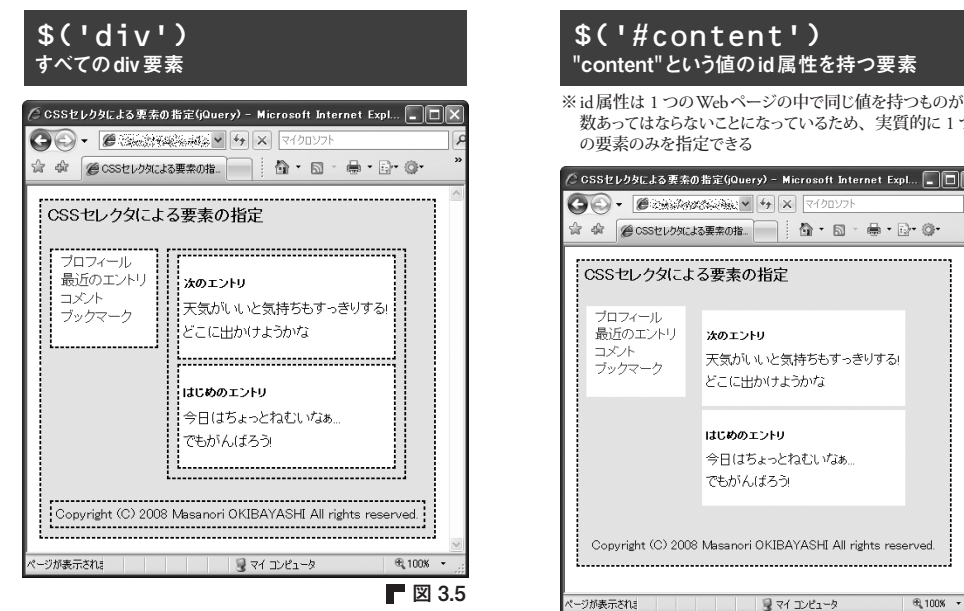
JavaScript だけで CSS セレクタには対応していませんが、jQuery では、Web ブラウザで対応可能な CSS1、CSS2.1 という仕様のセレクタはほとんど網羅しています。それから、仕様が正式決定していないために Web ブラウザではまだほとんど対応していない CSS3 のセレクタも、jQuery ではその多くをサポートしています*4。さらに CSS の仕様に関係なく jQuery で独自に追加されているセレクタもあります。詳細はリファレンス「セレクタ」を参照してください。



なお、CSS の仕様では擬似クラス (pseudo-classes) と定義されているものを jQuery ではフィルタ (filter) と呼んでいますので、jQuery のドキュメントを読む際には注意してください。

■使用例 要素を明示する例

リスト 3.3 を元に、CSS セレクタにより要素を指定する例をいくつか紹介していきましょう。これらは、組み合わせることでより複雑な指定にすることも可能です。指定された要素は画面上の枠線 (dashed による鎖線) で示します。



■ 図 3.5

■ 図 3.6

*4 Web ブラウザの CSS への対応状況は、下記ページに詳しく掲載されています (大きなコンテンツなので注意)
Web browser CSS support URL: <http://www.webdevout.net/browser-support-css>