

TIPS & TRICKSを超えた“本気で使える”テクニック

HTML MANIAX

第1回

ブラウザ振り分けでCSSを100%活用

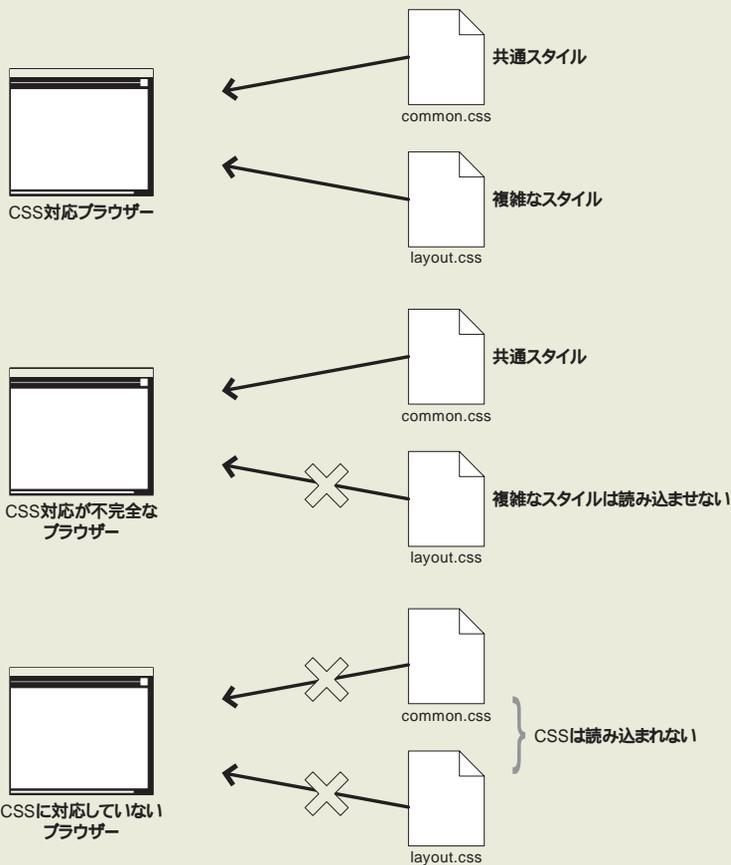
本連載では、HTML 4、XHTML、CSS 2、DOM対応のJavaScriptなど最新のウェブ技術を追求していく。今月は、これからの連載のカギとなるCSSの切り分けだ。記事ではIE 5以降やネットスケープ6以降で使うテクニックを研究するが、古いブラウザへの配慮も忘れてはならない。かといって古いブラウザでの見栄えにこだわるのも考えものだ。そこで重要になるのがCSSの読み込ませ方だ。

文:佐藤和人

今回紹介したテクニックは [Jump internet.impress.co.jp/maniax/](http://internet.impress.co.jp/maniax/) でソースを公開!



ナビゲーター4にはCSSを読ませるな!

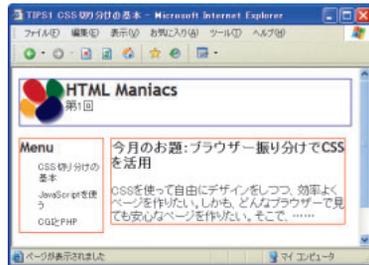


CSSを活用するうえで問題となるのがネットスケープナビゲーター4の存在だ。機能の貧弱なナビゲーター4で「まったく同じように見える」ウェブページを作ろうとしたら、CSSの恩恵を受けることができず、効率の悪いページ作りをしなければならぬ。ただし、古いブラウザを排除して「IEで読めればいいや」という態度を取るのはいただけない。

どんな環境の人にも共通の情報を伝えるというウェブの理念を守りつつテクニックを追求しよう。最新のブラウザには、CSSを使ったデザインを見せる。CSSに対応していないブラウザにもHTMLがきちんと読めるようにする。ナビゲーター4などCSS対応が不完全なブラウザには、CSSを読ませずにHTMLだけ表示させるようにする。ただし、使っても安全なCSSを切り分けて、簡単なデザインだけは見せる。これが今月のテーマだ。

[CSS 切り分けの基本]

せっかくCSSを使ってデザインを工夫したページを作っても、ナビゲーター4で見ると表示がぐちゃぐちゃになってしまうことがある。ナビゲーター4で問題が起きるCSSを切り分けて、読み込ませないようにしよう。



CSSでレイアウトしたページを作ってみたが.....。



ナビゲーター4ではぐちゃぐちゃな表示になってしまう。



ナビゲーター4に外部CSSを読み込ませないようにする。デザインは貧弱だが、中身は正確に読めるページになる。

media属性でナビゲーター4をはじく

```
<link rel="stylesheet" type="text/css" href="common.css">
<link rel="stylesheet" type="text/css" href="layout.css"
      media="screen,print">
```

@importルールを使ったブラウザ振り分け

```
A) @import url(layout.css); /* 除く N4 */
B) @import url("layout.css"); /* 除く N4, IE3, MacIE4.5 */
C) @import "layout.css"; /* 除く N4, IE3, WinIE4 */
```

tips1.html

```
<link rel="stylesheet" type="text/css" href="common.css">
<style type="text/css">
@import "layout.css";
</style>
```



バグを利用して
CSS切り分ける!

style要素やstyle属性を使ってCSSをHTML中に埋め込まずに、外部のCSSファイルをHTMLにリンクさせよう。CSSファイルは2つに分け、ナビゲーター4でも書けないスタイル/ページの背景色や背景画像、文字色、リンクの色など。このサンプルではcommon.css)と複雑なスタイル/ layout.css)を用意する。ソースのようなlink要素でHTMLに結び付けるが、layout.cssのほうは、media属性に「screen,print」という値を指定する。ナビゲーター4にはmedia属性が「screen」以外の値だとCSSを読まないというバグがある。このバグを利用すれば、ナビゲーター4はlayout.cssを読まず、レイアウトが崩れることはない。ナビゲーター4で見ると寂しいデザインになってしまうが、情報を正確に伝えるというウェブページの役目は果たせる。

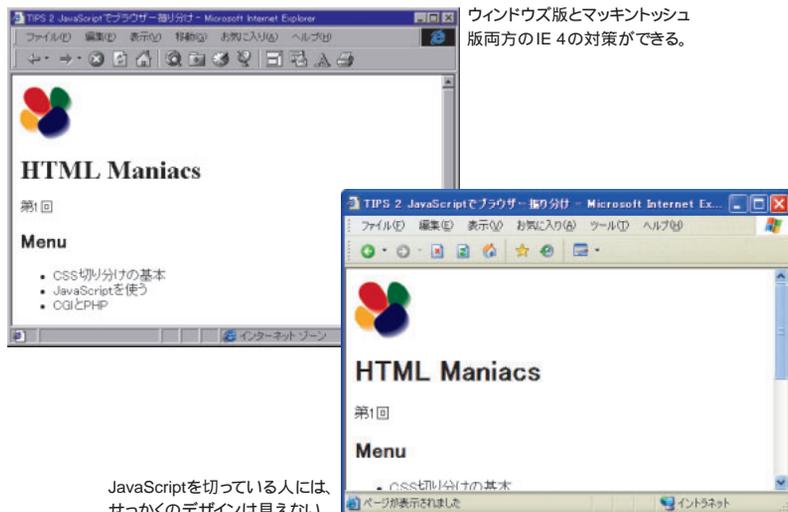


IE 4には完全に
対応できない

CSSの中で@importを使っても外部CSSファイルを取り込める。ソースのA)のように記述すれば、@import機能に対応していないナビゲーター4とIE3をはじくことができる。さらにIE4もはじきたい。今となってはIE4のCSSは貧弱だし、現在IE4のテスト環境を用意するのは難しい。ソースのB)のようにurl(~)の中をダブルクォーテーションで囲めば、マッキントッシュ版のIE4、4.5を除ける。C)のようにurl(~)を使わずにCSSを指定すれば、ウィンドウズ版IE4を除ける。しかし、ウィンドウズとマッキントッシュ両方のIE4を除けるようなブラウザのバグは見つからない。このサンプルでは妥協して、ソースのようにIE3、ウィンドウズ版IE4、ナビゲーター4にlayout.cssを読ませないことにした。

[JavaScriptでブラウザ振り分け]

前ページのようにバグを利用したテクニックでは、CSSに問題のあるブラウザすべてには対応できない。振り分けをもっと正確にするために、JavaScriptを使ってブラウザの種類やバージョンを調べてみよう。



JavaScriptを切っている人には、せっかくのデザインは見えないし。

tips2.html

```
<link rel="stylesheet" type="text/css" href="common.css">
<script type="text/javascript">
<!--
var ua = navigator.userAgent;
var tag = "<link rel='stylesheet' type='text/css' href='
layout.css'>";
if(!(ua.indexOf("Mozilla/4") >= 0 &&
    ua.indexOf("compatible") < 0) &&
    !(ua.indexOf("MSIE 3") >= 0) &&
    !(ua.indexOf("MSIE 4") >= 0))
    document.write(tag);
//-->
</script>
```

ソース中のマークは誌面レイアウトの関係で改行していますが、実際にはここには改行は入りません。

ユーザーエージェントの文字列

ブラウザ	navigator.userAgent
IE 4	Mozilla/4.0 (compatible; MSIE 4.01; Windows NT)
IE 5	Mozilla/4.0 (compatible; MSIE 5.0.1; Windows 98)
IE 5.5	Mozilla/4.0 (compatible; MSIE 5.5; Windows 98; Win 9x 4.90; T312461)
IE 6	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; Q312461)
Navigator 4	Mozilla/4.78 [ja] (Windows NT 5.0; U)
Netscape 6	Mozilla/5.0 (Windows; U; Windows NT 5.1; ja-JP; rv:0.9.4) Gecko/20011128 Netscape6/6.2.1
IE 4.5 (Mac)	Mozilla/4.0 (compatible; MSIE 4.5; Mac_PowerPC)
IE 5.1 (Mac)	Mozilla/4.0 (compatible; MSIE 5.14; Mac_PowerPC)



ユーザーエージェントを見極める!

左下の表のように、ブラウザは自分の種類やバージョンを表すユーザーエージェント文字列を持っている。JavaScriptでこの文字列を調べ、問題のあるブラウザ以外にlink要素を埋め込めばいいわけだ。ソースを見てみよう。common.cssはふつうにlink要素で指定する。layout.cssはJavaScriptで埋め込む。まず変数uaにユーザーエージェント文字列を取り出し、変数tagでlink要素を表す。if文で問題のあるブラウザかどうかを調べてlink要素を書き込む。ここではナビゲーター4とIEの3と4をはじいている。ナビゲーター4かどうかは、文字列に「Mozilla/4」が含まれていて、かつ「compatible」が含まれていないという条件でわかる。IEの3と4はそれぞれ「MSIE 3」と「MSIE 4」が含まれているかどうかでわかる。

JavaScriptを使う方法には、大きな欠点がある。最新のブラウザを使っている、ブラウザの設定でJavaScriptをオフにしていると、CSSが読み込まれないことだ。JavaScriptをオフにするかどうかはユーザーの自由なので、このテクニックはかなり不親切と言える。



特定のブラウザをひいきするな!

このサンプルを見て、「IE 5以上またはネットスケープ6以上」の場合にlink要素を書き出すようにしたほうがいいじゃないか、と思った人もいるかもしれない。しかし、CSSに対応しているブラウザはIEとネットスケープだけではなく、将来ユーザーエージェント文字列がどう変わるかはわからない。特定のブラウザをひいきしてCSSを読ませるより、問題のあるブラウザにCSSを読ませないほうが、柔軟性は高いのだ。

[CGIやPHPを活用する]

バグを利用する方法には限界がある。JavaScriptを使うとオフにしている人にデザインを見てもらえなくなる。CGIやPHPといったウェブサーバー上のプログラムを活用すれば、ブラウザ振り分けが完璧にできるぞ。



ハードディスク上のPHPファイルをIE 6に読み込ませてデザインの確認ができる。

tips3.html

```
<link rel="stylesheet" type="text/css" href="common.css">
<link rel="stylesheet" type="text/css" href="getcss.cgi">
```

getcss.cgi

```
#!/usr/local/bin/perl

print "Content-Type: text/css\n\n";
print "/* dummy text */\n";
$cssfile = "./layout.css";
$ua = $ENV{'HTTP_USER_AGENT'};
if(!($ua =~ /Mozilla¥4/ && $ua !~ /compatible/)
    && $ua !~ /MSIE (3|4)/) {
    if(open(IN, $cssfile)) {
        while(<IN>) { print $_; }
        close(IN);
    }
}
```

tips3.php

```
<link rel="stylesheet" type="text/css" href="common.css">
<?php
$ua = $HTTP_USER_AGENT;
if(!(ereg("Mozilla¥4", $ua) && !ereg("compatible", $ua)
    && !ereg("MSIE (3|4)", $ua))): ?>
<link rel="stylesheet" type="text/css" href="layout.css">
<?php endif; ?>
```



CSSのURLを
CGIにせよ!

CGIが使えるサーバーにページを置いているなら、CGIプログラムを経由してCSSを取り込む方法が使える。ソースのようにlink要素のhref属性をlayout.cssからCGIへのURL(getcss.cgi)に変える。ソースがPerlで書いたCGIプログラムだ。JavaScriptの場合と同じようにユーザーエージェント文字列を調べ、問題のあるブラウザでなければlayout.cssを読み込んでCSSを出力する。

ページを読み込むたびにCGIプログラムが実行されることになるので、サーバーへの負担が気にかかるところだ。しかし、ページを読み込むたびにアクセスカウンターCGIを呼び出しているサイトも多いので、そんなに無茶な方法というわけでもない。

CGIを使う方法の欠点は、パソコン上でHTMLを編集しながらデザインをいじるのが面倒になることだ。CSSを変更するたびにサーバーにアップロードしてテストしなければならない。ページを編集するたびに、一時的に「getcss.cgi」を「layout.css」に変え、アップロードする前に元に戻すというのもややこしい。



PHPならもっとお手軽

PHPは、HTMLの中に埋め込んだスクリプトをサーバー上で実行してからブラウザにHTMLごと渡してくれるプログラムだ。最近はPHPが使えるレンタルサーバーも増えてきた。ソースは、PHPを使ってブラウザを振り分け、link要素を埋め込むサンプルだ。<?php ~ ?>の部分はパソコン上ではブラウザに無視されるのでCSSのデザインをテストできる。そのままサーバーに置けばブラウザ振り分けを実行できる。PHPが使える人はぜひ試してほしい。



[インターネットマガジン バックナンバーアーカイブ] ご利用上の注意

このPDFファイルは、株式会社インプレスR&D(株式会社インプレスから分割)が1994年～2006年まで発行した月刊誌『インターネットマガジン』の誌面をPDF化し、「インターネットマガジン バックナンバーアーカイブ」として以下のウェブサイト「All-in-One INTERNET magazine 2.0」で公開しているものです。

<http://i.impressRD.jp/bn>

このファイルをご利用いただくにあたり、下記の注意事項を必ずお読みください。

- 記載されている内容(技術解説、URL、団体・企業名、商品名、価格、プレゼント募集、アンケートなど)は発行当時のものです。
- 収録されている内容は著作権法上の保護を受けています。著作権はそれぞれの記事の著作者(執筆者、写真の撮影者、イラストの作成者、編集部など)が保持しています。
- 著作者から許諾が得られなかった著作物は収録されていない場合があります。
- このファイルやその内容を改変したり、商用を目的として再利用することはできません。あくまで個人や企業の非商用利用での閲覧、複製、送信に限られます。
- 収録されている内容を何らかの媒体に引用としてご利用する際は、出典として媒体名および月号、該当ページ番号、発行元(株式会社インプレス R&D)、コピーライトなどの情報をご明記ください。
- オリジナルの雑誌の発行時点では、株式会社インプレス R&D(当時は株式会社インプレス)と著作権者は内容が正確なものであるように最大限に努めましたが、すべての情報が完全に正確であることは保証できません。このファイルの内容に起因する直接のおよび間接的な損害に対して、一切の責任を負いません。お客様個人の責任においてご利用ください。

このファイルに関するお問い合わせ先

株式会社インプレスR&D

All-in-One INTERNET magazine 編集部

im-info@impress.co.jp