

HTML MANIAX

第4回

マウスオーバー自由自在

リンクにマウスを載せたときに文字の色を変えたり、背景色を付けたりするワザはすっかりお馴染みだ。CSSの機能であるhover疑似クラスを利用したものだが、色を変える以上の使い方はなかなか思い付かないかもしれない。CSSのさまざまなプロパティを組み合わせれば、これまで想像もできなかったような効果も出せる。今回は、この「マウスオーバー」のテクニックをことごとく追究してみよう。

文:佐藤和人

HTML MANIAXで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!



JavaScriptなしでも「動き」は出せる!

マウスをリンクに載せたときに したい

HTML MANIAX

HTML MANIAX



JavaScriptはめんどくさい

```
<a href="maniax.html"
  onmouseover="....."
  onmouseout=".....">
```

CSSですっきり
a:hover { }

リンクにマウスを載せたときに動きを出すには、2つの方法がある。1つはa要素にイベント属性を付けてJavaScriptを記述する方法で、もう1つはCSSのhover疑似クラスを使う方法だ。a要素にhover疑似クラスを付けたセレクター「a:hover」を使えば、マウスを載せたときのリンクのスタイルを変更できる。面倒なスクリプトなしでもCSSの1行でページ全体のリンクの動きを変えられるのだ。

a:hoverを使うときの注意点は、CSSの仕様に忠実なネットスケープ6以降ではIEと動作が異なることだ。ネットスケープ6以降では、name属性を指定してリンクのターゲットにしたa要素にもa:hoverが効いてしまう。左の のようにリンク元になるa要素にだけhoverを付ける手もあるが、IEでうまくいかないことがある。 のようにname属性を持つa要素のスタイルを別に指定するのがおすすめだ。

注意点:リンク先になるa要素があるときは

リンク元になるa要素にだけスタイルを付ける

```
a:link:hover, a:visited:hover {
  color: white;
  background-color: black; }
```

name属性を持つa要素のスタイルを元に戻す

```
a[name]:hover, a[name]:active {
  color: inherit;
  background-color: transparent; }
```

[マウスオーバーのいろいろなワザ]

まずは、hover 疑似クラスを使った簡単なテクニックをいくつか紹介しよう。マウスを載せたときの色を変えたり、下線を付けたり消したりするページはよく見かけるが、hover 疑似クラスの力はそれだけではない。



layout1-1.css

```
a { text-decoration: none; }
a:hover { text-decoration: underline overline;
background-color: #FFFFA0; }
```



tips1-2.html

```
a:hover { position: relative; left: 6px; top: 6px;
background-color: #FFFFA0; }
```



tips1-3.html

```
a { text-decoration: none; border: 4px dotted #B0FFB0; }
a:hover { border-color: red; }
```



tips1-4.html

```
a { text-decoration: none; padding-left: 16px; }
a:hover { background: #FFFFA0 url(heart.gif)
no-repeat; }
```



マウスオーバーの秘技
怒濤の4連発!

上線と下線を付ける

ソース は、リンクにマウスを載せたときに、下線に加えて文字の上にも線が引かれるようにするCSSだ。a要素に text-decoration: none を指定して、リンクの下線を消す。a:hover の text-decoration プロパティには、underline に加えて overline も指定することで、マウスを載せたときにリンクの上下に線が引かれるようになる。

リンクを揺らす

ソース のCSSを使うと、リンクにマウスを載せたときに文字がへこむように右下に動く。a:hover で position プロパティを relative とし、left と top のプロパティに動かす長さを指定すればいい。ここでは6ピクセルにしたが、自分のページで使うときは、1~2ピクセル程度に抑えたほうがいい。

周囲に枠線を付ける

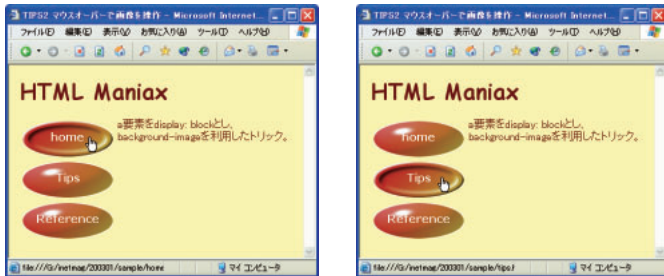
a のようなインライン要素にも、CSS の枠線が指定できる。ソース のサンプルでは、マウスを載せたときに周囲に赤い点線が付く。a要素にページの背景色と同じ色の枠線を指定し、a:hover で枠線の色を赤くしているだけだ。このワザに対応しているのはWindows版IE 5.5以降、マッキントッシュ版IE 5、およびネットスケープ6以降。

頭に画像を付ける

ソース は、リンクにマウスを載せるとハートの画像が文字の前に現れるというものだ。a:hover には、background プロパティで背景画像を指定する。no-repeat を加えれば、背景画像は1つだけ表示される。画像と文字がかぶらないように、a要素には padding-left プロパティで左に余白を空けておく。対応ブラウザはソース と同じだ。

[マウスオーバーで画像を操作]

a要素の中にimg要素を置いて画像リンクを作れば、マウスを載せたときにJavaScriptで画像を切り替えられる。ありふれたワザだ。ところが a:hover を使えば、それと同じようなことがCSSだけでもできるのだ。



マウスを載せたときに画像を切り替えるのはJavaScriptでおなじみの効果。実はCSSだけでもできる。

tips2-1.html(CSS部分)

```
#menu a { background-image: url(off.gif);
display: block; width: 140px; height: 60px;
text-align: center; text-decoration: none;
font: bold 16px/54px Verdana; }
#menu a:link { color: white; }
#menu a:visited { color: #FFC0C0; }
#menu a:hover { background-image: url(on.gif); }
```

tips2-1.html(HTML部分)

```
<p id="menu">
<a href="home.html">home</a>.....</p>
```



画像を横に並べるのはちょっと難しい.....

tips2-2.html(CSS部分)

```
#menu a img { border-style: none; }
#menu a img#home { background: url(off1.gif); }
#menu a:hover img#home { background: url(on1.gif); }
```

tips2-2.html(HTML部分)

```
<p id="menu">
<a href="home.html"></a>.....</p>
```

「a要素」を ブロックにする

ソース と は、ID属性が「menu」の「p要素」の中に、マウスオーバーで画像が変化するリンクを置いたものだ。

ボタン型の画像を2枚用意して、CSSを使って切り替えるには、aとa:hoverに別々の背景画像を指定すればいい。ただし、それだけではボタンの画像がリンクの大きさに収まらない。widthとheightのプロパティーで幅と高さを画像に合わせる必要がある。しかし、aのようなインライン要素には、widthとheightの効き目はない。

そこで、ソース のようにdisplayプロパティーにblockを指定して、a要素の見た目を無理矢理divのようなブロック要素型にする。するとwidthとheightが効くようになり、リンクを画像の大きさにぴったり合わせられる。

さらにソース では、リンクの文字が画像の中央に表示されるように、text-alignプロパティーにcenterを指定している。行の高さを54ピクセル(fontプロパティーの値の「54px」の部分)として、文字の上下位置も調整している。

横に並べるのは 少し難しいぞ!

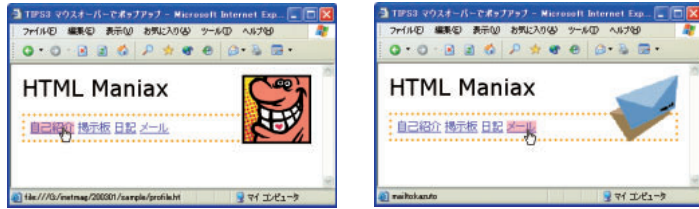
ソース と の方法では、aがブロック要素になるので、必然的にリンクが上下方向に並ぶことになる。aをインライン要素のままにして、画像リンクを左右に並べる方法はないだろうか。

ソース と は、透明なGIF画像を利用して、背景画像を切り替える裏ワザだ。ソース のようにa要素の中にimg要素で透明な画像を置く。ソースのCSSではa要素の中のimg要素に背景画像を指定している。透明な画像に背景画像を付けているわけだ。

なお、筆者の環境では、IEでこのワザを使うと、ときどきマウスを載せたときの画像が元に戻らなくなるという現象が見られた。

[マウスオーバーでポップアップ]

ここで紹介する2つのサンプルは、マウスを載せたときにリンクとは別の場所に画像や文字をポップアップさせるものだ。どう見てもJavaScriptを使っていると思えない。これがCSSだけでできるのだから驚きだ。



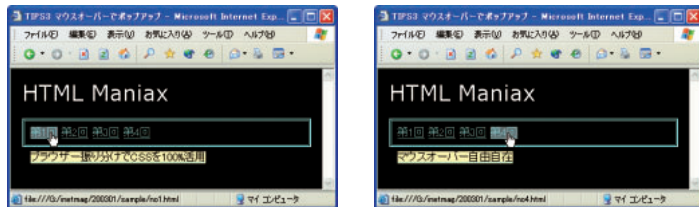
マウスを載せると右側に画像が出現する。これもCSSを使ったマニアックなワザだ。

tips3-1.html(CSS部分)

```
#menu a img { display: inline; width: 0; height: 0;
border-style: none; }
#menu a: hover { background-color: #FFD0D0; }
#menu a: hover img { position: absolute;
right: 1em; top: 1em; width: 100px; height: 100px; }
```

tips3-1.html(HTML部分)

```
<p id="menu">
<a href="profile.html">自己紹介
</a>.....</p>
```



マウスを載せると下側にリンクの説明文が出る。これもCSSだとはなんと驚きだ。

tips3-2.html(CSS部分)

```
#menu { position: relative; }
#menu a span { display: none; }
#menu a: hover { background-color: gray; }
#menu a: hover span { display: block;
position: absolute; top: 3em; left: 0em;
background-color: #FFFFC0; color: black; }
```

tips3-2.html(HTML部分)

```
<p id="menu">
<a href="no1.html">第1回
<span>ブラウザー振り分けでCSSを100%活用</span></a>
.....</p>
```



離れた場所に 画像を出現させる!

ソース と のサンプルは、リンクにマウスを載せると、リンク先をイメージした画像を画面右上に表示するものだ。どうやってリンクとは別の場所に画像を表示させているのだろう。

ソース を見てみよう。a要素の中にはリンクのテキストに加えてimg要素で画像を置いている。この画像はソースの「#menu a img」でwidthとheightのプロパティに0を指定しているため、見えなくなっている。「#menu a: hover img」でwidthとheightに幅と高さを指定することで、マウスを載せたときに画像が表示されるようになる。

このテクニックのミソは、マウスを載せたときのスタイルにposition: absoluteを加え、座標を指定して画像を配置していることだ。これで、a要素の中にあるimg要素を好きな場所にワープさせられるようになる。



リンクの説明表示を 自由自在に操れ!

ソース と では、マウスを載せるとリンクの説明が四角い枠の外に表示される。ソース のimg要素の代わりに、ソース ではa要素の中にspan要素を置く。ソース では、「#menu a span」でa要素の中のspan要素を非表示にする。「#menu a: hover span」ではリンクにマウスを載せたときにspan要素をdisplay: blockとして表示させ、座標を指定して動かしている。ID属性が「menu」のp要素にはposition: relativeを指定しているの、座標の原点はp要素の左上になることに注意しよう。

このページで使ったテクニックは、次のサイトを参考にしたものだ。CSSを使った高度なデザインが紹介されているので、ぜひ見てほしい。

CSS/edge

www.meyerweb.com/eric/css/edge/



[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