

# HTML MANIAX

第10回

## はじめてのクッキー

クッキーとは、ウェブページにアクセスしたクライアントPC上に小さなデータを保管し、再び同じウェブページにアクセスした際にそれを読み込む仕組みだ。前回にアクセスした際の状態の復元やユーザー認証などに利用することで、ユーザーごとに異なったサービスを提供できる。今月と来月の2回に分け、クッキーの基本をじっくりと紹介してみよう。

文:佐藤和人

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



## JavaScriptを使えばクッキーもお手軽

### クッキーを保存するときの書式

名前=値; expires=日付; path=パス; domain=ドメイン名; secure

- ・名前=値は必須。expires、path、domain、secureの各オプションは省略可。
- ・expiresはクッキーを破棄する期限を指定。
- ・pathはクッキーを読み取れるURLのパスを指定。
- ・domainはクッキーを読み取れるドメインの指定。
- ・secureはセキュアな通信(SSL)の場合だけクッキーをやり取りする指定。

例) name=kazuto; expires=Thu, 5 Jun 2003 23:43:31 GMT;  
path=/manix/; domain=.impress.co.jp

### クッキーを読み取るときの書式

名前1=値1; 名前2=値2; 名前3=値3; .....

- ・複数のクッキーが一度に送られる

例) name=kazuto; mail=kazuto@mvi.biglobe.ne.jp; id=12345

参考: Client Side State - HTTP Cookies

[URL http://wp.netscape.com/newsref/std/cookie\\_spec.html](http://wp.netscape.com/newsref/std/cookie_spec.html)

クッキーはブラウザとサーバーの間でやり取りされるHTTPヘッダーの中でデータを送受信するものだが、JavaScriptでも扱える。サーバー側で動的にページを生成できない場合でも、ユーザーがカスタマイズしたページの状態を保存するような仕掛けがお手軽にできるのだ。

クッキーを保存したり読み出したりするには、決まった書式に従う。ややこしいことに、保存するときと読み出すときの書式が違う。保存するときには、「名前=値」のペア1つとオプションをセミコロンで並べた文字列を使い、複数のクッキーを保存するには、この文字列を何度も送る。読み出すときには、複数のクッキーをセミコロンで並べたものが一度に出てくる。

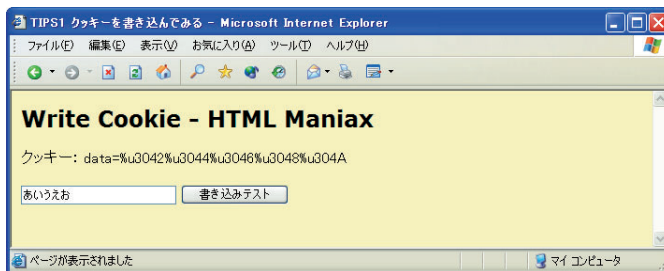
なお、クッキーがブラウザを通じて他のサーバーに漏れてしまうバグが発見されることがときどきあった。重要な個人情報などをクッキーに保存させるのは危険だ。

## [ 基本的なクッキーの読み書き ]

それでは早速JavaScriptを使ってクッキーを読み書きする練習をしてみよう。基本的なスクリプトを出発点にして、複雑な機能を少しずつ加えていけば、面倒そうなクッキーの処理もわりと簡単だと分かるだろう。



クッキーを書き込んでみる

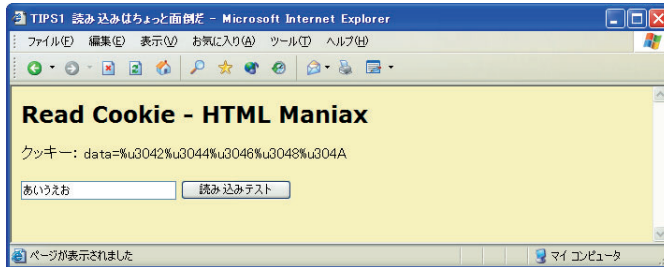


tips1-1.html

```
function writeCookie() {
    document.cookie = "data=" +
        escape(document.forms["f1"].elements["test"].value);
}
```



読み込みはちょっと面倒だ



tips1-2.html

```
function readCookie() {
    var pairs = document.cookie.split("; ");
    for(i = 0; i < pairs.length; i++) {
        var pair = pairs[i].split("=");
        if(pair[0] == "data") {
            document.forms["f1"].elements["test"].value
                = unescape(pair[1]);
            break;
        }
    }
}
```

JavaScriptでクッキーを保存するには、documentオブジェクトのcookieプロパティに「名前=値」の文字列を入れるだけでいい。ソースは、「書き込みテスト」ボタンを押すと呼び出される関数writeCookieで、フォームに入力された文字列を値にして「data」という名前のクッキーを保存するだけのものだ。クッキーではセミコロンや半角スペース、日本語の文字は使えないので、escapeを使ってエンコードすること。

公開中のサンプルでは、ボタンを押したあとでページを読み込み直すと、フォームの上にクッキーの内容が表示される。これは、document.writeを使ってdocument.cookieを出力しているだけだ（ソースは省略）。

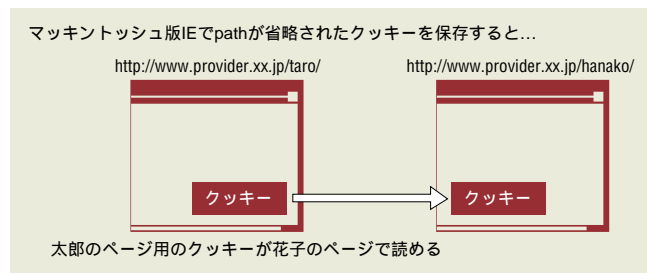
次に、保存されたクッキーから文字列を取り出してみよう。ソースは「読み込みテスト」ボタンを押したときに呼び出される関数readCookieだ。tips1-1.htmlで保存した値が、tips1-2.htmlという別のページで取り出せる。読み込むときもdocument.cookieを使えばいいのだが、書き込むときは「名前=値」のペアが1つだけなのに対して、読み込むときは「名前1=値1; 名前2=値2」のように、保存されていたほかのペアもすべて返される。ペアの間はセミコロンで区切られているので、まずsplitメソッドで分割して配列pairsに取り出す。配列pairsを1つ1つ調べ、やはりsplitメソッドで「=」を境に分割して名前(pair[0])と値(pair[1])を取り出す。名前が目的のもの("data")だったら、値のほうをフォームの入力欄に入れる。書き込むときに文字列はescapeでエンコードされているので、読み込むときはunescapeを使ってデコードすること。

# [ パスと有効期限を設定する ]

前々ページで紹介したように、クッキーを保存するときにはいくつかのオプションを加えて追加情報を一緒に保存できる。ここではクッキーを読み取れるパスの設定とクッキーの有効期限の設定について見てみよう。



パスは必ず設定しよう



tips2-1.html

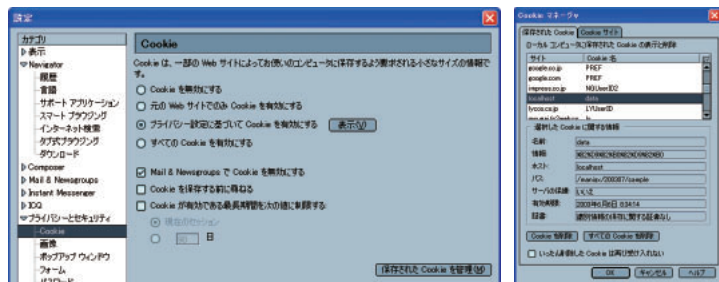
```
function writeCookie() {
    var cookie = "data=" +
        escape(document.forms["f1"].elements["test1"].value);
    cookie += "; path=/manix/";
    document.cookie = cookie;
}
```

クッキーを保存するときには、pathオプションでクッキーを読み取れるページのURLを制限できる。ソースはパスを/manix/以下に制限したものだ。http://internet.impress.co.jp/im/のようなURLではこのクッキーは読めなくなる。pathオプションを省略したときは、HTMLが置かれたパスが自動的にクッキーを読めるパスになる。

ところが、マッキントッシュ版のIEには、pathオプションを省略するとクッキーのパスが「/」になるバグがある。同じプロバイダーのサーバーを使っている他人のページでクッキーが読めてしまうのだ。このバグの対策として、パスは必ず設定するようにしよう。



クッキーの保存期間を決める



ネットスケープ7では、設定画面の「プライバシーとセキュリティ」 「Cookie」で「保存されたCookieを管理」ボタンを押すと、クッキーを確認できる。

tips2-2.html

```
function writeCookie() {
    var cookie = "data=" +
        escape(document.forms["f1"].elements["test1"].value);
    var d = new Date();
    d.setTime(d.getTime() + (1000*60*60*24*30));
    cookie += "; expires=" + d.toGMTString();
    cookie += "; path=/manix/";
    document.cookie = cookie;
}
```

expiresオプションを付けてクッキーを書き込むと、いつまでクッキーのデータを保存しておくかを指定できる。expiresオプションを省略した場合は、クッキーの保存期間はブラウザを閉じるまでとなる。ソースは、クッキーを30日間保存するスクリプトだ。Dateオブジェクトを使って現在の時刻を取り出し、30日(1000×60×60×24×30ミリ秒)を足す。DateオブジェクトのtoGMTStringメソッドを使うと「Thu, 5 Jun 2003 23:43:31 UTC」のような30日後の時刻を表す文字列が得られるので、これをそのままexpiresオプションの値にすればいい。

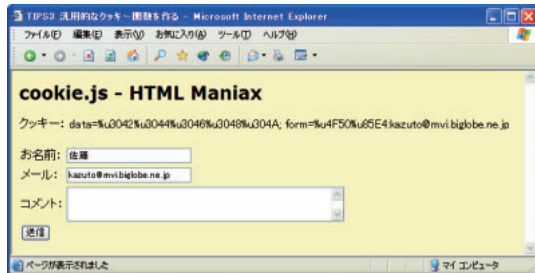
ちなみに、ネットスケープ7やマッキントッシュ版IE 5には、設定画面からパソコンに保存されているクッキーを1つずつ閲覧したり削除したりできる機能がある。この機能を使えば、パスや有効期限が正しく保存されているかどうかを確認できる。

## [ 汎用的なクッキー関数を作る ]

クッキーを使うページを作るたびに、いちいちクッキーのためのスクリプトを書くのは面倒だ。一度書いたらほかのページでも使えるように、クッキー処理の関数を提供する外部jsファイルを作成してみた。



外部jsファイルにして使い回そう



### 関数setCookie

setCookie(データ名, 値の配列, 保存する日数, パス);

### 関数getCookie

値の配列 = getCookie(データ名);

### tips3-1.html

```
<script type="text/javascript" src="cookie.js"></script>
<script type="text/javascript">
function saveForm() {
    var form = document.forms["f1"];
    setCookie("form",
        new Array(form.elements["name"].value,
            form.elements["mail"].value),
        30, "/maniax/");
}
function loadForm() {
    var form = document.forms["f1"];
    var values = getCookie("form");
    if(values[0]) form.elements["name"].value = values[0];
    if(values[1]) form.elements["mail"].value = values[1];
}
</script>
```

### tips3-1.html

```
<body onload="loadForm();">
```

### tips3-1.html

```
<form name="f1" action="tips3-1.html"
onsubmit="saveForm();">
```

今回筆者が作成したのは、HTMLからリンクして使うためのjsファイルcookie.jsだ。クッキーを保存するための関数setCookieと読み出すための関数getCookieを含んでいる。ソースの1行目のようにscript要素のsrc属性で指定すれば、クッキー処理機能をすぐに取り込めるようになる。誌面の都合上、関数setCookieとgetCookieのソースは省略した。インターネットマガジンのホームページからcookie.jsを入手して中身を眺めてほしい。

関数setCookieとgetCookieでは、クッキーの値に配列を指定して、複数のデータを一度に保存できるように工夫している。document.cookieに繰り返し「名前=値」を入れても複数のクッキーを保存できるのだが、訪問者がクッキーに対して警告を出すようにブラウザを設定していると、繰り返し警告が表示されてわずらわしいページになる。そこで、cookie.jsでは複数の値を「名前=値1:値2:値3」のようにコロンでつなげて保存するようにしている。

ソース ~ は、送信フォームに入力した名前とメールアドレスを保存して、再びページを訪れたユーザーの手間を省くスクリプトだ。関数saveFormはフォームを送信するときに呼び出され、setCookieを使って名前とメールアドレスを保存する。setCookieの引数は、順にクッキーの名前、値の配列、保存する日数、クッキーのパスだ。関数loadFormはページを読み込んだときに呼び出され、getCookieを使って保存した名前とメールアドレスを取り出し、フォームに戻すものだ。

次号ではこのjsファイルを再利用して、ユーザーが選んだスタイルを保存する例などを紹介するので楽しみに。



## [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](mailto:im-info@impress.co.jp)