

|Cookie-uri

[Sabin Corneliu Buraga](#)

Prezentare generala

Cookie-urile reprezinta un mecanism standard care permite ca un server (site) Web sa trimita anumite informatii pe calculatorul unui client (utilizator), prin intermediul browserului, sa ceara clientului sa stocheze aceste informatii pentru ca, ulterior, in diferite circumstante, navigatorul sa returneze informatiile spre serverul WWW. *Cookie*-urile pot fi privite asadar ca un mijloc persistent de stocare a datelor pe masina clientului Web cu scopul de a fi accesate ulterior de pe server, fiind utilizate la memorarea preferintelor utilizatorilor, la diverse tranzactii in comertul electronic, la completarea automata a formularelor, la stocarea informatiilor de autentificare etc.

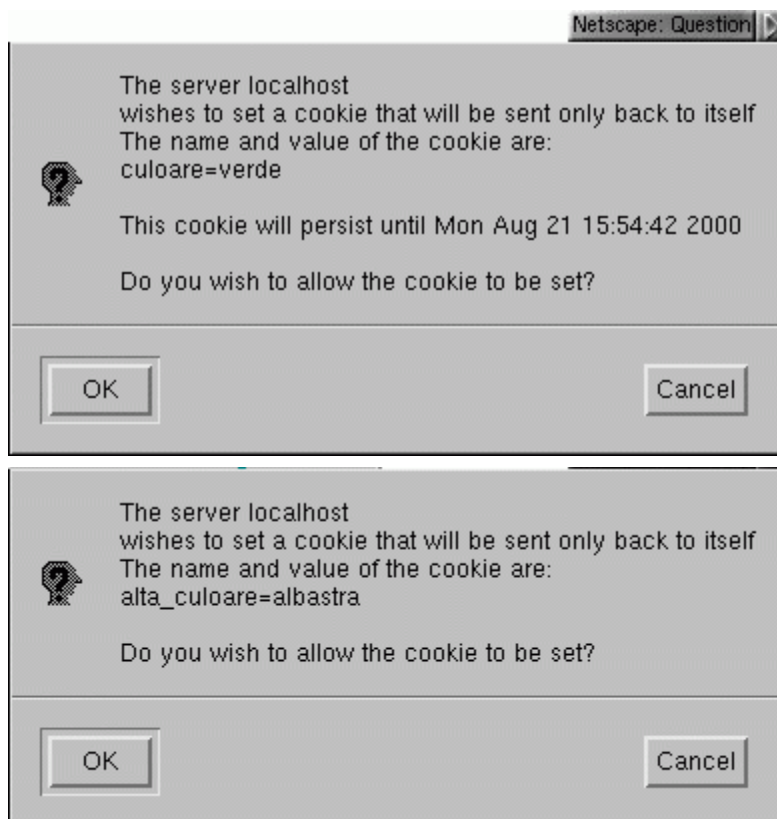
In cadrul unei tranzactii HTTP, serverul Web nu memoreaza nici o informatie despre aceasta, datorita modului de concepere a protocolului de transfer a datelor hipertext. In anumite cazuri, este insa util ca informatiile de stare ale unei conexiuni HTTP sa fie stocate pentru a fi folosite in cadrul altei tranzactii de date. Astfel, a aparut necesitatea implementarii *cookie*-urilor in cadrul navigatoarelor Web ca entitati purtatoare de date intre servere si clienti.

Termenul de *cookie* a fost introdus de *Lou Montulli*, unul dintre autorii specificatiei tehnice a browserului Netscape, care in versiunea 1.1 avea implementat in premiera mecanismul de *cookie*-uri.

Contrar temerilor celor mai multi utilizatori ai Internetului, *cookie*-urile nu sint periculoase pentru calculatoarele gazda, neavind drept continut decit citeva linii de text, neputind accesa programe de pe disc sau extrage informatii confidentiale despre utilizator pentru a fi trimise inapoi la un server Web. Datele detinute de un *cookie* sint datele pe care le ofera, din proprie initiativa, utilizatorul sau cele returnate in cadrul antetului HTTP la o tranzactie de date intre client si server. Un alt server diferit de serverul care a creat un *cookie* nu-l poate accesa.

De altfel, fiecare navigator Web ofera posibilitatea rejectarii *cookie*-urilor ori interogarii utilizatorilor daca sa fie acceptate sau nu. Programatorul poate seta un *cookie* sa fie persistent (nu va fi distrus la inchiderea navigatorului, ci va fi memorat intr-un fisier, dupa cum vom vedea mai jos, perioada lui de viata fiind stabilita de creator) sau nu.

O varianta extinsa e disponibila in cartea
 'Tehnologii Web', Matrix Rom, Bucuresti, 2001
<http://www.infoiasi.ro/~busaco/books/web.html>



Interogarea utilizatorului in legatura cu acceptarea de catre navigatorul Netscape a unui *cookie* persistent (salvat pe disc) si a unui *cookie* nepersistent

Atributele unui *cookie*

Un *cookie* consta in principal dintr-o pereche *nume=valoare* asemanatoare parametrilor vehiculati de scripturile CGI. Valoarea este un sir de caractere care trebuie codificat in maniera URL-urilor. Datele referitoare la un *cookie* vor fi receptionate de browser care mentine o lista de *cookie*-uri apartinand serverului care le-a trimis. Perechile (nume, valoare) ale *cookie*-urilor prezente in memoria navigatorului vor fi returnate spre procesare serverului care le-a creat, la cererea acestuia.

Mecanismul de setare a *cookie*-urilor pe calculatorul client si de retrimiteri a lor spre server se bazeaza pe cimpurile din antetul unei tranzactii de date HTTP.

Astfel, un *cookie* este trimis unui client incluzind un parametru-antet *Set-Cookie* intr-un raspuns HTTP, forma lui generala fiind: *Set-*

O varianta extinsa e disponibila in cartea
'Tehnologii Web', Matrix Rom, Bucuresti, 2001
<http://www.infoiasi.ro/~busaco/books/web.html>

```
Cookie: nume=valoare; expires=data; path=cale;  
domain=domeniu; secure.
```

Observam ca pot fi specificate attributele optionale:

- *expires*

Reprezinta data si timpul (in formatul `Wdy, DD-Mon-YYYY HH:MM:SS GMT`) cind *cookie*-ul va expira si va fi sters de pe disc. Daca nu este dat nici un timp de expirare, atunci *cookie*-ul se considera a fi nepersistent si va dispere la inchiderea navigatorului.

- *domain*

Atunci cind se cauta un *cookie* in lista de *cookie*-uri se efectueaza si o comparatie intre valoarea acestui atribut si adresa domeniului din care s-a receptionat antetul HTTP. Comparatia se realizeaza pornind de la sufixul valorilor comparate, in sensul ca daca avem `domain=uaic.ro` atunci aceasta valoare se va potrivi cu un nume de gazda precum `info.uaic.ro` sau `fenrir.info.uaic.ro`. Dupa efectuarea acesti comparatii, in cazul in care *cookie*-ul este validat se va verifica si valoarea atributului `path` (vezi mai jos). Valoarea implicita a atributului `domain` este numele simbolic al serverului care a generat acel *cookie*.

- *path*

Se utilizeaza pentru a specifica un subset de URL-uri din domeniul corespunzator unui *cookie* valid. Daca deja *cookie*-ul a fost validat in urma procedurii de comparare a domeniului, atunci componenta cale a URL-ului este comparata cu valoarea atributului `path`, in urma careia *cookie*-ul este considerat acceptat sau nu. Calea `/doc` se va potrivi, de exemplu, cu `/documentatii` si cu `/doc/index.html`, dar nu cu `/dom`. Cea mai generala cale de directoare este calea radacina `/`.

- *secure*

Daca un *cookie* este marcat ca `secure`, prin intermediul acestui atribut, va fi transmis numai daca tranzactia HTTP este una sigura (folosindu-se protocolul HTTPS).

Un *cookie* este transmis numai daca intruneste toate conditiile de validitate (se potrivesc domeniul, calea de directoare, timpul de expirare si securitatea canalului de comunicatie).

O varianta extinsa e disponibila in cartea
 'Tehnologii Web', Matrix Rom, Bucuresti, 2001
<http://www.infoiasi.ro/~busaco/books/web.html>

Serverul va primi de la client, in antetul HTTP, o linie de forma:
Cookie: nume1=valoare1; nume2=valoare2....

De retinut urmatoarele aspecte:

- pot fi trimise mai multe antete *Set-Cookie* intr-o tranzactie de date HTTP;
- desi un *cookie* nu a atins timpul de expirare, el poate fi sters fie de navigator, fie de utilizator;
- lungimea antetelor *Set-Cookie* si *Cookie* nu poate fi mai mare de 4 kilobytes;
- anumite navigatoare limiteaza numarul total de *cookie-uri* ce pot fi setate de un server/domeniu (la Netscape, numarul total este de 20);
- daca un script CGI vrea sa stearga un *cookie* de pe masina client, atunci va trimite un *cookie* avind acelasi nume ca si *cookie-ul* dorit a fi eliminat, cu valoare nula si cu timpul de expirare setat in trecut (i.e. Thu, 01-Jan-1970 00:00:00 GMT). Calea va fi de obicei stabilita la "/";
- nu se pot genera *cookie-uri* pentru domenii care nu apartin URL-ului transmis de serverul Web (un script din domeniul *infoiasi.ro* nu poate genera *cookie-uri* avind *domain=ibm.com*);
- antetul *Set-Cookie* nu va fi niciodata memorat in *cache-ul* unui *proxy* Web, fiind propagat spre client, indiferent daca se returneaza cod de stare 200 (*OK*) sau 304 (*Not Modified*);
- serverul IIS incluzind ASP va seta un *cookie* implicit denumit *ASPSESSIONID* pentru fiecare sesiune a unui utilizator. Acest *cookie* include informatii despre sesiunea curenta a unei aplicatii ASP si poate fi consultat asemeni unui *cookie* obisnuit.

Iata un exemplu de tranzactie HTTP in care se vehiculeaza diferite *cookie-uri* intre un server si un client Web:

1. Clientul cere serverului un document si primeste raspunsul:

```
Set-Cookie: culoare=verde; path=/; expires=Mon, 21-Aug-2000 15:54:
```

2. Cind clientul cere un URL al unui document stocat pe server in directorul radacina "/", va trimite:

O varianta extinsa e disponibila in cartea
 'Tehnologii Web', Matrix Rom, Bucuresti, 2001
<http://www.infoiasi.ro/~busaco/books/web.html>

```
Cookie: culoare=verde
```

3. Clientul primeste documentul, insotit de:

```
Set-Cookie: alta_culoare=albastra;path=/
```

4. Atunci cind clientul doreste un document memorat in directorul "/" va expedia antetul:

```
Cookie: culoare=verde;alta_culoare=albastra
```

5. Clientul va primi:

```
Set-Cookie: count=3;path=/doc
```

6. Cind clientul va cere un document stocat pe server in directorul /doc va trimite antetul:

```
Cookie: culoare=verde;alta_culoare=albastra;count=3
```

7. Atunci cind clientul cere serverului un document din directorul "/" va expedia:

```
Cookie: culoare=verde;alta_culoare=albastra
```

Stocarea *cookie*-urilor la client

Un *cookie* este limitat la 4 kilobytes lungime, fiecare *cookie* fiind salvat intr-un fisier separat ori toate *cookie*-urile fiind memorate intr-un fisier comun, in functie de tipul si de versiunea browserului folosit. La fel, numarul total de *cookie*-uri ce pot popula hard-disk-ul este dependent de navigator (la Netscape, maxim 300). *Cookie*-urile pot fi sterse, fie de catre serverul care le-a produs, fie prin intermediul unei optiuni a navigatorului, fie de catre utilizator (cautindu-le pe disc). Localizarea fisierelor continind *cookie*-uri este, de asemeni, dependenta de navigator. La Netscape 4, exista un fisier denumit *cookies.txt* (varianta Windows sau Macintosh) sau *cookies* (varianta UNIX/Linux) aflat in directorul personal al utilizatorului, la Internet Explorer 3 si 5 pentru fiecare utilizator se creeaza in directorul *Cookies* al directorului unde s-a instalat Windows-ul un fisier *nume@adresa.txt*, in care *nume* este numele utilizatorului pe masina Windows, iar *adresa* reprezinta adresa serverului Web care

O varianta extinsa e disponibila in cartea
'Tehnologii Web', Matrix Rom, Bucuresti, 2001
<http://www.infoiasi.ro/~busaco/books/web.html>

a setat acel *cookie*. La anumite variante de IE 4, se stocheaza in directorul `Temporary Internet Files`. Exemple de nume de fisiere valide stocind *cookie*-uri generat de Internet Explorer sint `sabin@yahoo.txt` sau `sabin@C__TEMP__COOKIES.txt`. Pentru Lynx, daca este configurat sa salveze *cookie*-urile pe disc, atunci ele vor fi stocate in directorul home al utilizatorului in fisierul `.lynx_cookies`. Acest fisier (al carui nume se poate modifica in `lynx.cfg`) are acelasi format ca si fisierul `cookies` generat de Netscape.

Un exemplu de continut al unui fisier `cookies` Netscape este cel de mai jos:

```
localhost FALSE / FALSE 966862979 culoare verde
localhost FALSE / FALSE 966782841 alta_culoare albastra
.uaic.ro FALSE / TRUE 966862979 secure Secure%20communications
.uaic.ro FALSE /doc FALSE 969367226 count 3
.yahoo.com FALSE FALSE 966862979 path http%3A//www.yahoo.
```

Cimpurile din acest fisier au urmatoarele semnificatii:

Cimp	Descriere
domeniu	Domeniul care a creat si care poate consulta <i>cookie</i> -ul (atributul <code>domain</code> dintr-un antet <code>Set-Cookie</code>)
indicator de acces	Este <code>TRUE</code> daca toate masinile din domeniul dat pot accesa <i>cookie</i> -ul. Aceasta valoare este setata automat de browser in urma comparatiei de sufixe de domenii
cale	Valoarea atributului <code>path</code>
indicator de securitate	Indica existenta atributului <code>secure</code>
timpul de expirare	Valoarea atributului <code>expires</code> masurata in numarul de secunde scurse de la data de 01 ianuarie 1970, ora 00:00:00 GMT (timpul UNIX)
nume	Numele <i>cookie</i> -ului
valoare	Valoarea asociata <i>cookie</i> -ului, codificata

In cazul lui Internet Explorer, fisierul de *cookie*-uri denumit `sabin@www.elsevier.txt` generat in urma acceptarii unui *cookie* de la `www.elsevier.com` are urmatoarea configuratie criptica:

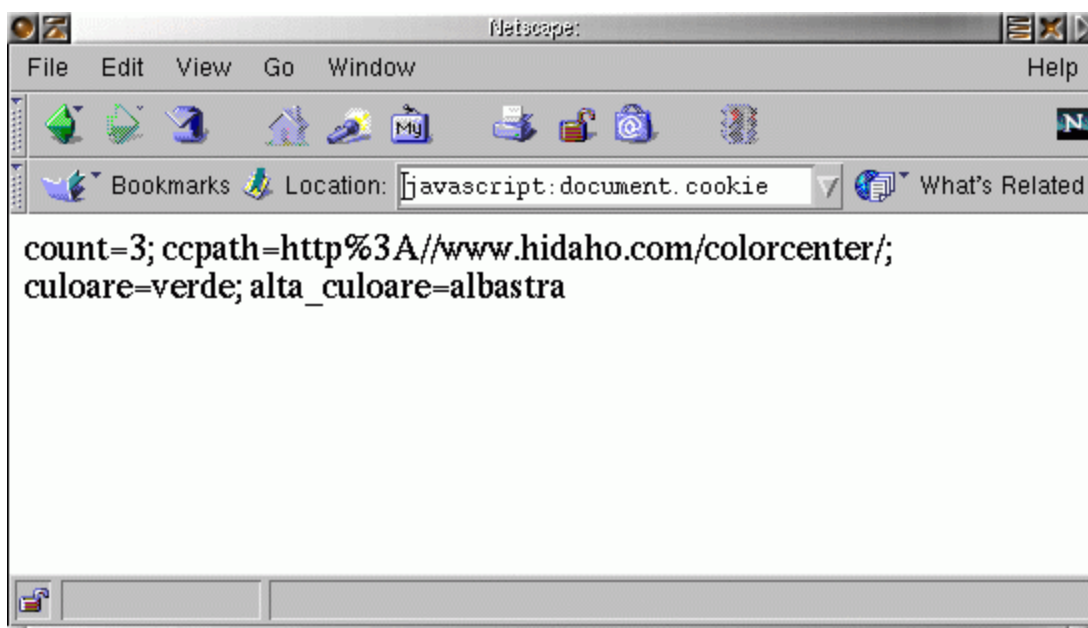
```
AnEFromrel_client
```

O varianta extinsa e disponibila in cartea
'Tehnologii Web', Matrix Rom, Bucuresti, 2001
<http://www.infoiasi.ro/~busaco/books/web.html>

```
www.elsevier.com/  
0  
3415826816  
29364629  
4113724112  
29364628  
*
```

Vizualizarea informatiilor continute de *cookie*-uri

Pentru a vedea toate *cookie*-urile (atit cele persistente cit si cele nepersistente) prezente in memoria navigatorului, ne putem sluji de constructia JavaScript `document.cookie` pe care putem s-o introducem in locul unei adrese (URL), dupa cum observam in figurile urmatoare:



Consultarea informatiilor referitoare la *cookie*-uri

In Lynx, acelasi lucru se realizeaza vizualizind asa-numitul *Cookie Jar* (activat cu combinatia de taste **CTRL+K**):

O varianta extinsa e disponibila in cartea
 'Tehnologii Web', Matrix Rom, Bucuresti, 2001
<http://www.infoiasi.ro/~busaco/books/web.html>

```

File Edit Settings Help
Cookie Jar

Lynx (2.8.3dev.18), help on Cookie Jar

NOTE: Activate links to gobble up cookies or entire domains, or to
change a domain's 'allow' setting.

localhost
  Domain=localhost (Cookies always allowed.)
  alta_culoare=albastra
  Path=/
  Port: 80 Secure: NO Discard: NO
  Maximum Gobble Date: Sun Aug 20 19:52:10 2000

Commands: Use arrow keys to move, '?' for help, 'q' to quit, '<-' to go back.
Arrow keys: Up and Down to move. Right to follow a link; Left to go back.
H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list
  
```

Managementul cookie-urilor in Lynx

Aceasta facilitate ofera si posibilitatea de a organiza cookie-urile.

Programarea cookie-urilor

Vom enumera in continuare citeva abordari pentru crearea si accesarea cookie-urilor atat pe partea server, cit si pe partea client, in cadrul scripturilor CGI, in PHP, ASP si JavaScript.

Crearea cookie-urilor

Dupa cum am vazut, un cookie este creat prin intermediul unui antet `Set-Cookie` intr-o tranzactie HTTP. Astfel, intr-un script CGI putem scrie:

```
#!/bin/bash

echo "Set-Cookie: culoare=verde; path=/; expires=Mon, 21-Aug-2000 15:54
..."
```

O alta cale este sa utilizam obiectul predefinit `document.cookie` intr-o functie JavaScript. Astfel, urmatoarea functie definita de *Bill Dortch* seteaza un cookie:

O varianta extinsa e disponibila in cartea
'Tehnologii Web', Matrix Rom, Bucuresti, 2001
<http://www.infoiasi.ro/~busaco/books/web.html>

```
function SetCookie(name, value, expires, path, domain, secure) {  
    document.cookie = name + "=" + escape(value) +  
        ((expires) ? "; expires=" + expires.toGMTString() : "") +  
        ((path) ? "; path=" + path : "") +  
        ((domain) ? "; domain=" + domain : "") +  
        ((secure) ? "; secure" : "");  
}
```

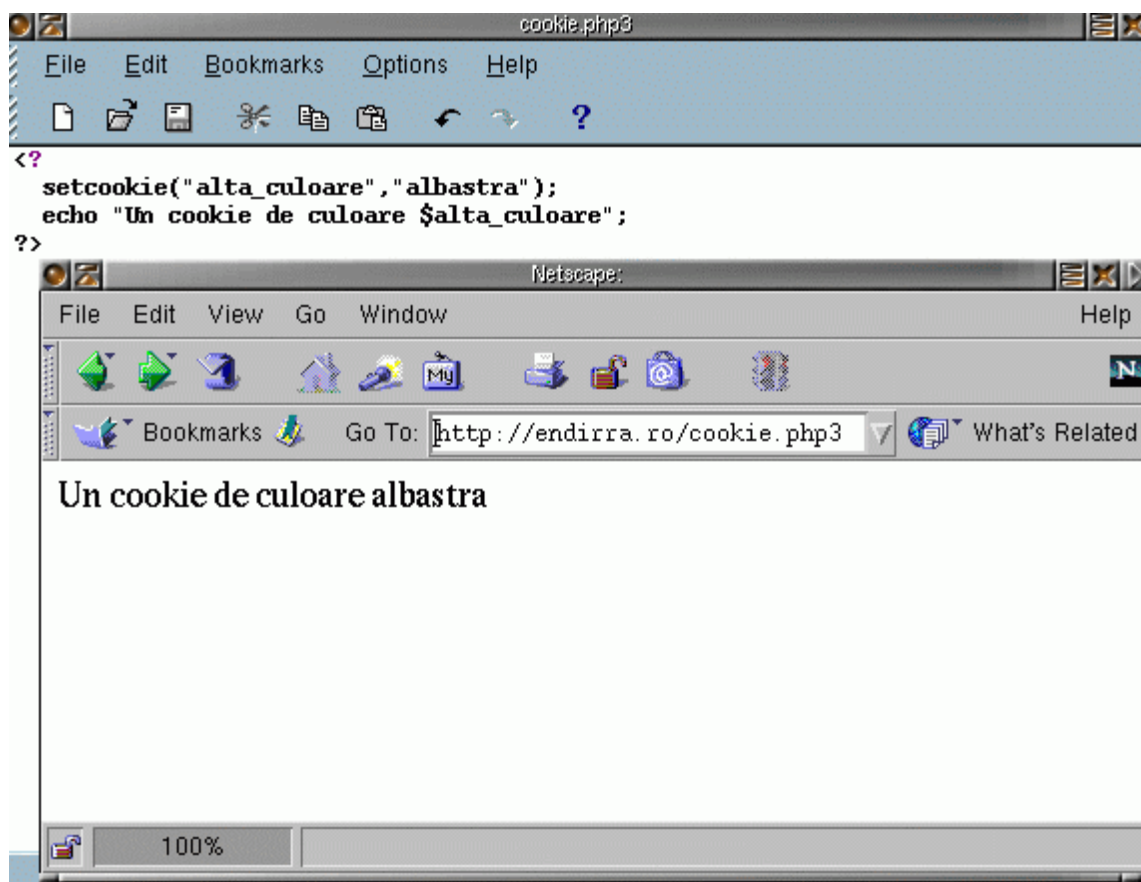
Funcția predefinită `escape()` codifică un șir de caractere în format similar codificării URL-urilor.

Un exemplu de utilizare:

```
var expdate = new Date();  
expdate.setTime(expdate.getTime() + (24*60*60*1000));  
    // setam sa expire dupa o zi de la data curenta  
SetCookie("culoare", "verde", expdate);  
SetCookie("alta_culoare", "albastra");  
SetCookie("sec", "Necesita SSL", expdate, "/", null, true);
```

În PHP se poate folosi funcția predefinită `setcookie()` după cum se poate observa din figura următoare (sintaxa acestei funcții este similară funcției JavaScript de mai sus):

O varianta extinsa e disponibila in cartea
 'Tehnologii Web', Matrix Rom, Bucuresti, 2001
<http://www.infoiasi.ro/~busaco/books/web.html>



Crearea si accesarea unui cookie in PHP

Funcția `setcookie()` trebuie scrisă înaintea oricărui marcator HTML (înainte de `<html>`) pentru ca antetul `Set-Cookie` să fie primul trimis clientului.

În ASP vom utiliza obiectul `Response.Cookies`. Iată un exemplu:

```
<%
Response.Cookies("culoare") = "verde"
Response.Cookies("culoare").Expires = "January 01, 2001"
Response.Cookies("culoare").Domain = "infoiasi.ro"
Response.Cookies("culoare").Path = "/"
Response.Cookies("culoare").Secure = false
%>
```

Accesarea cookie-urilor

Pentru a vedea ce valoare are un *cookie*, putem consulta antetul HTTP `Cookie`. Cel mai ușor este să preluăm valoarea variabilei de

O varianta extinsa e disponibila in cartea
 'Tehnologii Web', Matrix Rom, Bucuresti, 2001
<http://www.infoiasi.ro/~busaco/books/web.html>

mediu `HTTP_COOKIE` setata de serverul Web in cadrul unui script CGI scris, de exemplu, in Perl:

```
@envpairs = split(/;/, $ENV{'HTTP_COOKIE'});
foreach $pair (@envpairs) {
    ($name, $value) = split(/=/, $pair);
    $cookie{$name} = $value;
}
```

Am creat un tablou asociativ `cookie` detinind informatiile despre toate *cookie*-urile disponibile. Putem asigura unei variabile valoarea *cookie*-ului numit `alta_culoare` astfel:

```
$fundal = $cookie{'alta_culoare'};
```

In JavaScript vom utiliza tot `document.cookie`, apelind la functia definita de acelasi Bill Dortch in modul urmat:

```
function GetCookie (name) {
    var arg = name + "=";
    var alen = arg.length;
    var clen = document.cookie.length;
    var i = 0;
    while (i < clen) {
        // parcurgem intreg sirul de perechi "cookie=valoare"
        var j = i + alen;
        if (document.cookie.substring(i, j) == arg)
            // am gasit
            return getCookieVal(j);
        i = document.cookie.indexOf(" ", i) + 1;
        if (i == 0)
            break;
    }
    // n-am gasit...
    return null;
}
```

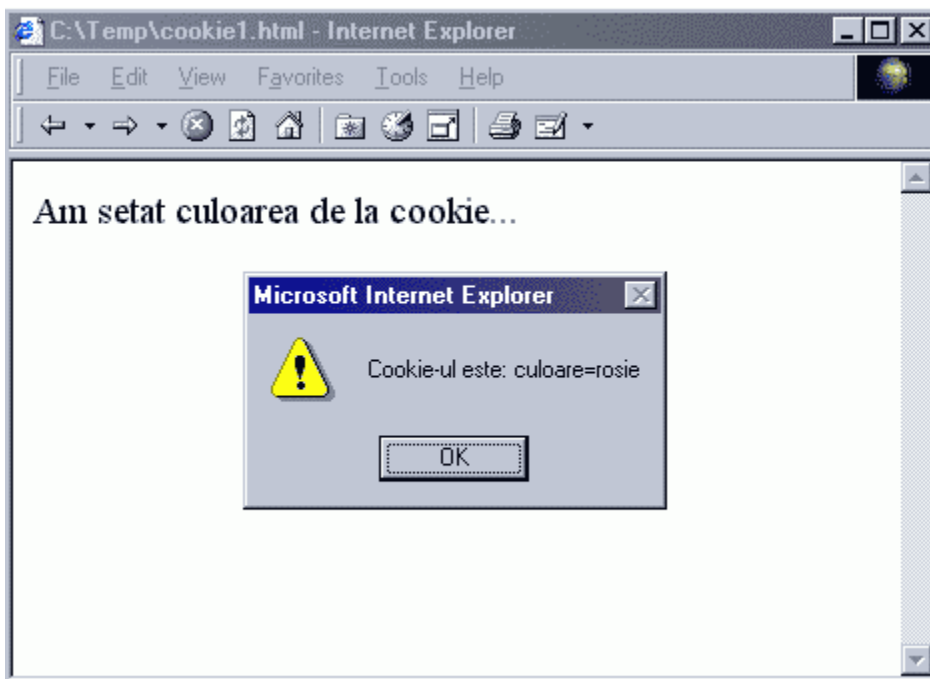
Functia `getCookieVal()` este:

```
function getCookieVal (offset) {
    var endstr = document.cookie.indexOf(";", offset);
    if (endstr == -1)
        endstr = document.cookie.length;
    return unescape(document.cookie.substring(offset, endstr));
}
```

Functia `unescape()` este definita de specificatia limbajului, realizind transformarea din forma codata URL in forma normala de

O varianta extinsa e disponibila in cartea
'Tehnologii Web', Matrix Rom, Bucuresti, 2001
<http://www.infoiasi.ro/~busaco/books/web.html>

reprezentare a unui sir de caractere.



Vizualizarea valorii unui *cookie* in JavaScript

In PHP, fiecare *cookie* este disponibil ca variabila predefinita, valoarea unui *cookie* fiind accesibila prin constructia `$nume_cookie`.

Pentru ASP se poate folosi `Request.Cookies` in maniera de mai jos:

```
<p>Culoarea preferata este <%= Request.Cookies("culoare") %>.</p>
```

In loc de final...

Ce culoare au *cookie*-urile de pe calculatorul dumneavoastra?

Domnul Sabin Corneliu Buraga este doctorand in Computer Science la Facultatea de Informatica a Universitatii "A.I.Cuza" din Iasi si poate fi contactat la adresa busaco@infoiasi.ro.