

Utilizare variabile cookie

1. Accesul la variabilele cookie

Valorile majoritatii variabilelor dispar atunci când scriptul PHP care le contine își încheie executia. Spre deosebire de acestea, valorile variabilelor cookie se pot pastra un timp indefinit. Pentru ca valorile lor sa se poata pastra, browserul utilizatorului stocheaza variabilele cookie în unitatea de hard-disc a utilizatorului.

Majoritatea navigatoarelor de Web au posibilitatea sa activeze/dezactiveze cookie-urile. Trebuie sa tineti cont de acest lucru cand doriti sa folositi cookie in scripturile dv., totusi, conform statisticilor, in jur de 1% dintre utilizatori au optiunea cookie dezactivata.

PHP pune la dispozitia utilizatorilor functii pentru transmiterea cookie-urilor de la server la browser, si modalitati de citire a cookie-urilor.

Pentru a citi cooki-urile puteti utiliza variabila PHP de tip tablou `$_COOKIE` sau `$_HTTP_COOKIE_VARS`, acestea contin numele si valoarea fiecărei variabile cookie curenta. Daca doriti sa obtineti acces la acest tablou, puteti folosi un program ca urmatorul:

- **Foreach(`$_COOKIE` as `$nume` => `$valoare`)**
**echo "
 `$nume` => `$valoare`";**

- `$_COOKIE['numeCookie']` este o variabila super-globala, putand fi accesata si in functii fara a mai specifica "global".

2. Crearea unei variabile cookie

Un lucru important de retinut este faptul ca cookie-urile trebuie sa fie transmise la browser inaintea oricarei alte informatii.

Pentru a evita problemele, asigurati-va ca un script PHP care stabileste o valoare a unei variabile cookie este plasat în partea superioara a fisierului, fara caractere de tip spatiu alb care sa-l preceada. De asemenea, stabiliti valoarea variabilei cookie înainte de a executa o instructiune 'echo' sau o alta instructiune PHP care trimite browserului date de iesire. Pentru a crea o variabila cookie, folositi functia `setcookie()`, care are urmatoarea forma:

- `setcookie(numeCookie, valoare, expirare)`

- Argumentul 'numeCookie' specifica numele variabilei cookie, iar argumentul 'valoare' specifica valoarea variabilei.

- Valoarea variabilei cookie este automat transmisa "urlencode" si cand este primita e automat decodata.-

Argumentul 'expirare' indica momentul expirarii variabilei cookie; dupa ora specificata, variabila cookie nu mai este accesibila.

În general, este convenabil sa se specifice momentul expirarii folosind functia `time()`, care returneaza intervalul de timp (exprimat în secunde) scurs de la 1 ianuarie 1970. Puteti adauga o valoare de tip decalaj (offset), care specifica intervalul de timp pe durata caruia variabila cookie trebuie sa fie accesibila.

Folosind aceasta functie se pot transmite mai multe cookie-uri succesiv; protocoalele Web limiteaza insa la maximum 20 numarul cookie-urilor ce pot fi trimise aceluiasi utilizator. De exemplu, sa luam în considerare urmatoarea instructiune:

- `setcookie("fruct", "banana", time()+3600);`

Aceasta instructiune creeaza o variabila cookie denumita 'fruct', care are valoarea 'banana'. Variabila cookie va fi disponibila timp de o ora (3600 secunde) de la crearea sa.

Daca preferati, puteti specifica momentul expirarii folosind functia `mktime()`. Aceasta functie are urmatoarea forma:

- `mktime(ore, minute, secunde, luna, zi, an)`

De exemplu, urmatoarea instructiune creeaza o variabila cookie care expira la o secunda dupa miezul noptii primei zile a anului 2009:

- `Setcookie("fruct", "banana", mktime(0, 0, 1, 1, 1, 2009));`

3. Stergerea unei variabile cookie

Deoarece o variabila cookie are o data de expirare, aceasta va fi stearsa automat la un oarecare interval de timp dupa crearea sa. Totusi, puteti sterge o variabila cookie imediat. Pentru aceasta, fixati momentul expirarii variabilei cookie la un moment de timp in trecut.

De exemplu, pentru a sterge o variabila cookie denumita 'fruct', puteti folosi urmatoarea instructiune:

- `setcookie("fruct", "", time()-3600);`

Aceasta instructiune stabileste timpul de expirare cu o ora (3600 de secunde) în urma. Remarcati ca valoarea variabilei cookie este exprimata sub forma unui sir vid (""); din moment ce variabila cookie nu va mai fi disponibila, valoarea sa nu mai are importanta.

4. Specificarea accesului la o variabila cookie si alte optiuni

Functia `setcookie()` poate prelua maximum sase argumente.

Iata formatul complet al functiei `setcookie()`:

- `setcookie(nume, valoare, expirare, cale, domeniu, secure)`

Argumentele 'nume', 'valoare' si 'expirare' au fost descrise în sectiunea precedenta.

- Argumentul 'cale' va permite sa specificati calea URL asociata variabilei cookie. În mod

prestabilit, variabila cookie este disponibila pentru scripturile din catalogul care contine scriptul în care a fost configurata variabila respectiva, precum si pentru scripturile din sub-cataloagele aferente catalogului respectiv. În particular, scripturilor din cataloagele parinte ale catalogului care contine scriptul nu li se permite accesul prestabilit la variabila cookie.

Pentru a pune variabila cookie la dispozitia scripturilor dintr-un anumit catalog si cataloagele sale, specificati o valoare a argumentului 'cale'. De exemplu, pentru a pune variabila cookie la dispozitia întregului arbore de cataloage, specificati "/" ca valoare a argumentului cale; pentru a face variabila cookie disponibila în catalogul "/test" si în sub-cataloagele sale, specificati "/test/" ca valoare a argumentului 'cale'.

- Daca nu este specificat nici un argument 'domeniu', o variabila cookie este disponibila numai pentru scripturile din domeniul Web care a creat variabila respectiva. Argumentul 'domeniu' va permite sa specificati numele de domeniu asociat unei variabile cookie. În consecinta, variabila cookie va fi disponibila numai pentru siturile Web din cadrul domeniului specificat. De exemplu, sa presupunem ca un script din serverul Web <http://www.subdomeniu.domeniu.com> creeaza o variabila cookie. În mod prestabilit, variabila cookie este disponibila numai pentru gazda respectiva. Cu toate acestea, puteti face variabila cookie disponibila pe întreg domeniul 'subdomeniu.domeniu.com', specificând "subdomeniu.domeniu.com" ca valoare a argumentului 'domeniu'.

- Argumentul 'secure' este o valoare întreaga, care specifica daca variabila cookie trebuie trimisa prin intermediul unei conexiuni sigure (HTTPS). Specificati valoarea 1 pentru a împiedica transmiterea variabilei cookie în cazul în care conexiunea nu este sigura; pentru a permite transmiterea variabilei cookie prin conexiuni HTTP obisnuite, specificati valoarea 0.

Lucrul cu fisiere

1. Obținerea atributelor unui fisier

Privilegiile asociate unui fisier sau unui catalog determina operatiile pe care le poate executa un utilizator cu fisierul sau catalogul respectiv.

Pentru a adauga un fisier intr-un director, folosind PHP, trebuie sa aveti permisiunea de scriere in acel director, de asemenea, pentru a adauga date intr-un fisier, trebuie sa aveti permisiunea de scriere in fisierul respectiv.

Aceste permisiuni sunt date (configurate) de administratorul sistemului respectiv. PHP furnizeaza numeroase functii care va permit sa obtineti informatii care descriu un fisier. Tabelul urmator rezuma cele mai cunoscute dintre aceste functii.

Functii PHP pentru obținerea atributelor unui fisier
--

Funcție	Descriere
file_exists()	- Returneaza TRUE daca fisierul specificat exista, respectiv FALSE în caz contrar
fileatime()	- Returneaza timpul de acces la fisier sub forma de amprenta de timp UNIX.
filectime()	- Returneaza timpul de modificare al i-nodului (structura de date care contine informatii despre fisiere UNIX - N. T.) sub forma de amprenta de timp UNIX.
filegroup()	- Returneaza identificatorul numeric al grupului care detine fisierul
filemtime()	- Returneaza momentul de timp al modificarii fisierului sub forma de amprenta de timp UNIX
fileowner()	- Returneaza identificatorul numeric de utilizator al fisierului
fileperms()	- Returneaza permisiunile fisierului
filesize()	- Returneaza dimensiunea fisierului, în octeti
filetype()	- Returneaza tipul fisierului, în speta "fifo", "char", "dir", "block", "link", "file" sau "unknown"
is_dir()	- Returneaza TRUE daca directorul specificat exista; în caz contrar, returneaza FALSE
is_file()	- Returneaza TRUE daca fisierul specificat exista; în caz contrar, returneaza FALSE
is_readable()	- Returneaza TRUE daca fisierul specificat exista si poate fi citit; în caz contrar, returneaza FALSE
is_writable()	- Returneaza TRUE daca fisierul specificat exista si se poate scrie în acel fisier; în caz contrar, returneaza FALSE

Funcțiile fileowner() si filegroup() returneaza fiecare un identificator numeric; puteti converti identificatorul numeric într-un sir prin invocarea functiei posix_getpwuid() cu un identificator de utilizator, respectiv a functiei posix_getgrgid() cu un identificat de grup. Acestea returneaza informatii despre identificatorul precizat.

Iata un exemplu in care puteti vedea, si intelege mai bine, cum sunt folosite functiile rezentate in tabelul anterior (fisierul 'test.txt' care va fi folosit de functii se afla in acelasi director cu scriptul php):

```
<?php
$filename = "test.txt";
$result = file_exists($filename);
echo "<br /> file_exists(): $result";
$result = fileowner($filename);
echo "<br /> fileowner(): $result";
$result = filegroup($filename);
echo "<br /> filegroup(): $result";
$result = filetype($filename);
echo "<br /> filetype(): $result";
$result = filesize($filename);
echo "<br /> filesize(): $result";
```

```

$result = fileatime($filename);
$result = date("m/d/Y H:i", $result);
echo "<br /> fileatime(): $result";
$result = filectime($filename);
$result = date("m/d/Y H:i", $result);
echo "<br /> filectime(): $result";
$result = filemtime($filename);
$result = date("m/d/Y H:i", $result);
echo "<br /> filemtime(): $result";
$result = fileperms($filename);
$result = decoct($result);
echo "<br /> fileperms(): $result";
$result = is_file($filename);
echo "<br /> is_file(): $result";
$result = is_dir($filename);
echo "<br /> is_dir(): $result";
$result = is_readable($filename);
echo "<br /> is_readable(): $result";
$result = is_writable($filename);
echo "<br /> is_writable(): $result";
?>

```

2. Privilegii de fisier

Privilegiile asociate unui fisier (sau director) determina operatiile pe care utilizatorii le pot efectua cu fisierul (sau directorul) respectiv.

Privilegiile sunt date sub forma a trei grupuri alcatuite din câte trei caractere fiecare; cu alte cuvinte, trei triade. Prima triada indica privilegiile acordate posesorului fisierului. Cea de-a doua triada indica privilegiile acordate membrilor grupului posesor al fisierului. Cea de-a treia triada indica privilegiile acordate altor utilizatori.

Cele 3 caractere care alcatuiesc un grup sunt reprezentate, in sistemele UNIX, prin urmatoarele trei litere: 'r', 'w' si 'x'; acestea reprezinta:

- r - fisierul poate fi citit
- w - se poate scrie în fisier
- x - continutul fisierului poate fi executat

Caracterele unei triade apar întotdeauna în secventa 'rwx'. Daca o anumita litera este înlocuita de o cratima (-), privilegiul asociat nu este accesibil.

De exemplu, sa examinam privilegiile specificate in urmatorul exemplu:

- rwxr-xr--

Aceste caractere au urmatoarea semnificatie:

- "rwx", posesorul fisierului poate citi, scrie sau executa fisierul
- "r-x", membrii grupului posesor al fisierului pot citi sau executa fisierul, dar nu pot scrie în fisier
- "r--", alti utilizatori pot citi fisierul, dar nu pot scrie în fisier sau executa continutul fisierului

Privilegiul 'x' este semnificativ numai pentru fisierele care includ un continut executabil, cum sunt fisierele binare executabile (.exe) sau anumite categorii de scripturi.

Posesorul unui fisier poate modifica privilegiile asociate fisierului emitând comanda chmod (aceasta functioneaza în sistemele de operare UNIX).

Aceasta comanda are doua forme. O forma va permite sa specificati privilegiile folosind cifre scrise în octal; cealalta va permite sa le specificati folosind litere.

Pentru a specifica privilegiile folosind cifre în octal, calculati valoarea numerica a fiecărei triade. Pentru aceasta, însumati numerele corespunzatoare fiecarui privilegiu disponibil din cadrul triadei. Numerele asociate privilegiilor sunt urmatoarele:

- r - 4
- w - 2
- x - 1

De exemplu, privilegiul "rwx" are valoarea $4 + 2 + 1 = 7$. Similar, privilegiul "r-x" are valoarea $4 + 1 = 5$, iar privilegiul "r--" are valoarea 4.

Dupa ce ati calculat valoarea numerica a fiecărei triade, formati un numar din trei cifre scris în octal, care este alcatuit din valoarea numerica a privilegiilor utilizatorilor, valoarea numerica a privilegiilor membrilor grupului, si valoarea numerica a privilegiilor altor utilizatori. Astfel, privilegiile "rwxr-xr--" corespund valorii în octal 754.

3. Modificarea privilegiilor unui fisier

Pentru a modifica privilegiile unui fisier, folositi functia chmod(), care are urmatoarea forma:

- chmod(nume_fisier, mod)

Argumentul 'nume_fisier' specifica numele si calea de acces a fisierului ale carui privilegii urmeaza a fi modificate, iar argumentul 'mod' specifica privilegiile dorite, în general, se prefera exprimarea privilegiilor sub forma unui numar scris în octal. Pentru aceasta, prefixati valoarea folosind cifra 0.

De exemplu, puteti atribui unui fisier "text.txt" privilegiile "rwxr-xr--" specificând pentru 'mod' valoarea 0754, functia "chmod" va fi chmod("text.txt", 0754)

Pentru ca functia chmod() sa se execute cu succes, PHP trebuie sa ruleze sub contul utilizatorului posesor al fisierului. Functia returneaza TRUE în caz de reusita, respectiv FALSE în caz contrar.

Pentru a modifica grupul posesor al unui fisier, invocati functia chgrp(), care are forma:

- `chgrp("nume_fisier", grup)`

- Unde "grup" este un nume sau numar care desemneaza grupul respectiv.

Funcțiile `chmod()` si `chgrp()` nu functioneaza sub Microsoft Windows.

4. Deschiderea unui fisier

Înainte de a putea citi sau scrie într-un fisier, trebuie sa deschideti fisierul folosind functia `fopen()`.

- `fopen("nume_fisier", "mod")`

- unde 'nume_fisier' specifica numele si calea spre fisierul care urmeaza a fi deschis, iar 'mod' indica tipul de acces dorit. Modurile de deschidere a unui fisier sunt urmatoarele:

- `r` - Permite doar citirea fisierului
- `r+` - Citire sau scriere de la inceputul fisierului
- `w` - Creaza fisierul daca nu exista si suprascrie datele existente
- `w+` - Citire sau scriere; la scriere , creeaza fisierul daca nu exista si suprascrie datele existente
- `a` - Adaugare; Creaza fisierul daca nu exista si adauga datele noi la sfarsitul fisierului existent
- `a+` - Citire sau scriere; la scriere, creeaza fisierul daca nu exista si adauga datele noi la sfarsitul fisierului existent
- `x` - Doar scriere; Creaza fisierul daca nu exista si genereaza un avertisment daca acesta exista
- `x+` - Citire sau scriere; Creaza fisierul daca nu exista si genereaza un avertisment daca acesta exista
- `b` - Deschide fisierul in mod binar

Funcția `fopen()` returneaza `FALSE` daca PHP nu a reusit sa deschida fisierul, în caz contrar, returneaza o valoare care contine un întreg, numit "identificator de fisier", care se foloseste pentru identificarea unui fisier de catre functiile care executa operatii cu fisiere.

De exemplu, instructiunea urmatoare deschide fisierul 'carte.txt' pentru citire:

- `$fh = fopen("carte.txt", "r");`

Sub Microsoft Windows, fisierele ASCII si fisierele binare sunt tratate în mod diferit. Când deschideti un fisier binar sub Windows, specificati "b" ca al doilea caracter al modului; de exemplu, "rb" pentru citire, "wb" pentru scriere, "a+b" pentru citire si adaugare. Daca nu procedati astfel, citirile din fisier si alte operatii se vor încheia prematur sau vor esua.

5. Verificarea finalizarii unei operatii cu un fisier

Operatiile cu fisiere (inclusiv cele legate de deschiderea, citirea unui fisier si scrierea într-un fisier) pot esua dintr-o varietate de motive. Deci, este important sa verificati daca fiecare operatie s-a încheiat cu succes. Iata o modalitate în care puteti proceda:

```
<?php
$fh = fopen("carte.txt", "rb");
if (!$fh) {
    echo "Nu a fost deschis fisierul carte.txt.";
}
?>
```

Functia fopen() returneaza FALSE daca nu reuseste sa deschida fisierul. In acest caz, scriptul executa instructiunea echo care afisaza un mesaj.

O forma alternativa mai simpla foloseste operatorul OR (SAU):

```
<?php
($fh = fopen("carte.txt", "rb"))
    OR die("Nu a fost deschis fisierul carte.txt.");
?>
```

6. Inchiderea unui fisier

Un fisier deschis consuma resursele sistemului. Când un script a terminat de utilizat un fisier, scriptul trebuie sa închida fisierul, eliberând aceste resurse.

La sfârșitul unui script, PHP închide în mod automat fisierele deschise. Totusi, la programare se recomanda sa închideti fisierele mai rapid, ori de câte ori este posibil. Pentru a închide un fisier, folositi functia fclose():

- fclose(identificator_fisier)

- 'identificator_fisier' este identificatorul fisierului, returnat la deschiderea acestuia cu functia fopen()

Iata un exemplu caracteristic de deschidere, utilizare si închidere a unui fisier:

```
<?php
$fh = fopen("carte.txt", "rb");
if (!$fh) {
    echo "Nu a fost deschis fisierul carte.txt.";
}
// Aici se insereaza instructiunile care folosesc fisierul deschis
fclose($fh);
?>
```

Functia fclose() returneaza valoarea TRUE daca fisierul a fost închis cu succes. Testarea acestei valori este rareori necesara, deoarece nu se mai pot face prea multe dupa ce s-a ratat o încercare de a închide un fisier.