

Fisiere si foldere in PHP

Folosirea fisierelor in PHP

In orice limbaj de programare, si deci si in PHP, cand se lucreaza cu fisiere trebuiesc efectuate urmatoarele operatiuni:

1. deschiderea fisierului
2. citirea din fisier/scrierea in fisier
3. inchiderea fisierului

Instructiunile PHP corespunzatoare operatiilor de mai sus sunt prezentate in cele ce urmeaza. De remarcat este faptul ca trebuie specificat inca de la deschiderea fisierului ce fel de operatii vor avea loc: citire, scriere sau ambele.

```
<?php
# deschiderea unui fisier si citirea din el
$id_fisier = fopen("c:\\folder\\fisier.txt", 'r'); // deschidere
fread( $id_fisier, 10 ); // citire 10 octeti din fisier
fclose($id_fisier); // inchidere
# deschiderea unui fisier, stergerea continutului si scrierea in el
$id_fisier = fopen("c:\\folder\\fisier.txt", 'w'); // deschidere
fwrite( $id_fisier, 'Text nou in fisier' );
fclose($id_fisier); // inchidere
# deschiderea unui fisier si scrierea la sfarsitul lui
$id_fisier = fopen("c:\\folder\\fisier.txt", 'a'); // deschidere
fwrite( $id_fisier, 'Text adaugat in fisier' );
fclose($id_fisier); // inchidere
?>
```

Asa cum s-a observat, un mod de deschidere pentru citire ('r') si 2 moduri de deschidere pentru scriere: cu stergere a continutului vechi (modul 'w') si cu pastrare a continutului (modul 'a'), caz in care scrierea se face la sfarsitul fisierului, imediat dupa continutul vechi. Depinde de specificul fiecarei situatii ce mod de scriere trebuie ales.

Limbajul PHP dispune si de alte moduri de deschidere a fisierului, care sunt mici variatii ale celor prezentate mai sus.

Instructiunile de mai sus pot fi folosite cu orice tip de fisier, text sau binar si reprezinta baza in lucrul cu fisierele. Exista o suita de alte functii pentru citirea sau scrierea anumitor tipuri de date specifice, ce pot fi folosite cu ambele tipuri de fisiere.

Instructiuni pentru fisiere text

Pentru simplificarea codului de citire/scriere a unui fisier in situatii generale, PHP ofera cateva functii foarte convenabile: `file_get_contents`, `file_put_contents` si `file`. Primele doua, permit preluarea intregului continut al unui fisier si punerea lui intr-o variabila string, printr-un singur apel, respectiv crearea unui fisier care sa contina valoarea unei variabile (ca in exemplele mai jos). Cea de-a treia functie permite crearea unui vector ce are ca elemente liniile fisierului specificat. De exemplu, un fisier cu 3 linii va genera un vector cu 3 elemente, fiecare element reprezentant o linie din fisier.

Este important de mentionat ca odata citit continutul fisierului (cu `file_get_contents` si `file`) variabilele se "deconecteaza" de fisier, adica orice modificare facuta asupra variabilei nu se reflecta si asupra fisierului, cele 2 entitati fiind separate.

```

<?php

// pun tot continutul fisierului intr-o variabila
$continut = file_get_contents( 'fisier.txt' );

// in acest moment tot continutul este stocat in variabila $continut
// pot procesa continutul sau il pot modifica

// adaug ceva la final
// Nota: modificarea se face doar in variabila $continut, nu si in fisier
$continut = $continut . ' -- Text adaugat de PHP';

// la final salvez variabila $continut intr-un al doilea fisier
file_put_contents( 'fisier2.txt', $continut );

// creea un alt fisier care contine 2 cifre doar
file_put_contents( 'fisier3.txt', '11' );

// din continutul unui fisier creez un vector format din liniile fisierului
$linii = file( 'fisier4.txt' );

echo 'Linia a 3a din fisier este: ', $linii[ 2 ];

?>

```

Manipularea fisierelor in PHP

Pe langa functiile pentru preluarea/modificarea continutului fisierelor, PHP ofera si facilitati de gestionare a fisierelor: creare, mutare, copiere, modificare attribute, etc. Cateva din acestea sunt prezentate mai jos.

```

<?php

// verificare daca un fisier exista
echo file_exists( 'fisier.txt' );

// copiere fisier
copy( 'sursa.txt', 'destinatie.txt' );

// stergere fisier
unlink( 'fisier.txt' );

// redenumire sau mutare
rename( 'vechi.txt', 'nou.txt' );

// afisarea numelui unui fisier
echo basename( "c:\\cale\\catre\\fisier.txt" ); // afiseaza fisier.txt

// afisarea folderului unui fisier

```

```

echo dirname( "c:\\cale\\catre\\fisier.txt" ); // afiseaza c:\\cale\\catre

// afisarea dimensiunii unui fisier in octeti
echo filesize( 'fisier.txt' );

// verificare daca numele specificat este un fisier
echo is_file( "c:\\cale\\catre\\fisier.txt" ); // afiseaza true

// verificare daca un fisier poate fi citit
echo is_readable( 'fisier.txt' );

// verificare daca un fisier poate fi scris/modificat
echo is_writable( 'fisier.txt' );

?>

```

Directoare in PHP

Manipularea directoarelor (folderelor) folosind PHP se face la fel de usor ca in cazul fisierelor. Majoritatea functiilor folosite pentru fisiere se pot aplica si la foldere (de exemplu copy, rename, is_file, etc), dar exista o serie de alte instructiuni specifice dosarelor.

```

<?php

// afiseaza directorul curent (current working directory)
// de obicei este folderul in care se afla scriptul ce se executa
echo getcwd(); // ex. c:\\scripturi

// schimba directorul curent
chdir( 'exemple' ); // ex. c:\\scripturi\\exemple

// returneaza un vector ce contine numele fisierelor si directoarelor dintr-un folder
$vector = scandir( getcwd() );
print_r( $vector );

// verifica daca un element este director (folder)
echo is_dir( "c:\\cale\\fisier.txt" ); // afiseaza false

?>

```

Limbajul PHP dispune de o serie de functii ce permit citire continutului unui folder intr-un mod similar cu preluarea continutului unui fisier. Astfel, exista functii pentru deschiderea unui director (opendir), citirea continutului, adica a fisierelor sau folderelor existente in acel director (readdir) si inchiderea lui (closedir). O situatie in care aceste functii pot fi folosite este aceea cand se doreste afisarea unei liste a elementelor continute intr-un folder si se doreste efectuarea unor calcule sau procesari pe baza fiecaruia dintre aceste elemente.

```

<?php

```

```

$folder = getcwd();          // va lista folderul curent
$handle = opendir( $folder ); // deschid folderul

if ( !empty( $handle ) ) {
    echo "Fisiere si directoare:\n";

    $terminat = false;
    while ( $terminat == false ) {
        $file = readdir($handle); // citesc urmatorul fisier
        if( $file === false ) {
            // atentie la operatorul de exactitate ===
            // daca nu mai sunt alte fisiere/foldere trebuie sa ies din bucla
            $terminat = true;
        } else {
            // aici pot face orice procesare, de exemplu sa redenumesc fisierul/subdirectorul
            // doar afisez numele fisierului/subdirectorului
            echo "$file\n";
            echo "<br />";
        }
    }
}

closedir($handle); // inchei citirea din folder
}
?>

```

Tema

1. In fisierul text intrare.in se afla doua cuvinte pe cate o linie fiecare. Afisati in fisierul iesire.out cuvintele cu literele inversate (ex. dana=>anad).
2. In fisierul text intrare.in se afla n numere naturale. Afisati in fisierul iesire.out numerele in ordine crescatoare.
3. În fişierul email.in se află pe câte o linie câte o adresă de email. Validati aceste adrese si scrieti rezultatul în fişierul email.out.
4. În fişierul cnp.in se află pe câte o linie câte un CNP. Validati aceste CNP-uri si scrieti rezultatul în fişierul cnp.out.