



GESTIONE FILE

PROF. ASSUNTA SANTANIELLO

INCLUSIONE DI CODICE DA FILE ESTERNO

Le funzioni possono essere definite indifferentemente nell' HEAD o nel BODY, oppure ancora possono essere incluse da un file esterno con l'istruzione **include()**.

Nel caso dell'inclusione è buon uso che avvenga nell'intestazione, per evitare problemi di visibilità e perché essa verrà caricata subito in memoria.

Nel seguente esempio vedremo come caricare una funzione presente in un file esterno, il quale può avere una qualunque estensione, anche se per uniformità i programmatori php usano **.inc**

INCLUSIONE DI CODICE DA FILE ESTERNO

La funzione `disegnaTabelle()` è presente in un file esterno chiamato `disegna_tab.inc`, che viene richiamato attraverso il comando `include`

```
<!DOCTYPE html>
<head>
</head>
<body>
    <?php
        echo "Il disegno della tabella avviene includendo una funzione esterna<br>";
        //inclusione del file contenente la funzione disegnaTabella
        include("C:/xampp/htdocs/CORSO_PHP/disegna_tab.inc");
        //chiamata della funzione
        disegnaTabella(10,20);
    ?>
</body>
</html>
```

Esercizio.php

INCLUSIONE DI CODICE DA FILE ESTERNO

disegna_tab.inc

```
<?php
function disegnaTabella($righe,$colonne)
{
    echo "<TABLE WIDTH=30% BORDER=1>";
    for($r=0;$r<$righe;$r++)
    {
        echo "<TR>";
        for($c=0;$c<$colonne;$c++)
        {
            echo"<TD>&nbsp;</TD>";
        }
        echo "</TR>";
    }
    echo "</TABLE>";
}
?>
```

I FILE IN PHP

I file in php vengono gestiti con istruzioni analoghe a quelle del linguaggio. L'apertura del file avviene utilizzando la funzione **fopen()**

Nome file = percorso del file da aprire

```
$idmiofile = fopen(«nome del file», «modalità di apertura»);
```

La chiusura del file avviene utilizzando la funzione **fclose()**, indicando come parametro il puntatore relativo al file

MODALITA' DI APERTURA

Modalità di apertura, invece, può essere una delle seguenti:

MODALITA' DI APERTURA	SIGNIFICATO
r	SOLO LETTURA
r+	LETTURA E SCRITTURA
w	SOLA SCRITTURA: IL CONTENUTO VIENE PERSO (*)
w+	LETTURA E SCRITTURA, MA PERDONO IL CONTENUTO (*)
a	SOLO PER AGGIUNTA (modalità append) (*)
a+	PER LETTURA E AGGIUNTA(*)
b	DA AGGIUNGERE AI PRECEDENTI PER TRATTARE IL FILE COME FILE BINARIO

(*) se il file non esiste tenta di crearlo

PRINCIPALI FUNZIONI PER LA GESTIONE DEI FILES

FUNZIONE	SIGNIFICATO
Fopen(\$filedaaprire,\$mode)	Apre il file
Fread(\$filedaleggere)	Legge da un file
Unlink(\$filedaeliminare)	Elimina un file
File_exists(\$filedacontrollare)	Controlla se un file esiste
is_writable(\$filedacontrollare)	Controlla se un file è riscrivibile
is_readable(\$filedacontrollare)	Controlla se un file è leggibile
Fwrite(\$testodascrivere, \$fileincuiscrivere)	Scrive un file eliminandone il contenuto precedente
Fsize(\$filedacontrollare)	Restituisce la dimensione del file in byte
Fclose(\$filedachiusure)	Chiude un file

ESEMPIO PRATICO

Esempio pratico con l'utilizzo delle funzioni di apertura, scrittura e chiusura del file

```
<!DOCTYPE html>
<head>

</head>
<body>
    <?php
        $idfile=fopen("accessi.txt","r+");
        //se l'apertura del file fallisce la funzione die() visualizza il messaggio e chiude lo script
        if(!$idfile)
        {
            die("Si è verificato un errore durante l'apertura dello script!");
        }
        $scriviamo = fwrite($idfile,"ciaone");
        // Verifica se la scrittura ha avuto successo
        if ($scriviamo === false)
        {
            die("Errore durante la scrittura nel file.");
        }
        else
        {
            echo "Scrittura nel file riuscita.";
        }
        // Chiudi il file
        fclose($idfile);
    ?>
</body>
</html>
```


PROVA TU!

Realizza un programma in cui viene verificata l'esistenza del file `esercizio.txt`. Se il file non esiste, crealo memorizzando 0 al suo interno. Quando il file esiste, lo script deve provvedere ad aprirlo e dopo aver verificato se si riesce ad aprirlo, legge il valore in esso contenuto.

A questo punto, la variabile che contiene il totale di visite viene incrementata. Quindi dopo aver riaperto il file in modalità scrittura, viene scritto il nuovo valore. Infine, viene visualizzato il totale browser.

I FILE CSV

Capita spesso però di dover gestire tipologie di file differenti da un file .txt, tra cui: file search CSV, file JSON, file XML.

In particolar modo ci concentreremo come gestire i file CSV.

I file CSV sta per Comma-Separated Values e, come dice il nome stesso è un formato di file dove i dati sono un **elenco di valori separati da una virgola**. Viene utilizzato per importare ed esportare tabelle di dati.

I FILE CSV

In generale, in un file search CSV si dovrebbero rispettare le seguenti regole:

1. Separare i campi tramite punto e virgola.
2. Se i campi contengono virgole o punto e virgole racchiuderli tra doppi apici.
3. Separare le righe con un a capo (\n);
4. Non lasciare spazi prima e dopo i campi.
5. Se il campo presenta un carattere di doppio apice racchiuderlo tra doppi apici:
" diventa "".

CREARE UN FILE CSV

```
<!DOCTYPE html>
<head>
</head>
<body>
  <?php
    $filename = 'users.csv';
    $users = [
      ['Cristiano', 'Ronaldo', 'Portogallo'],
      ['Gonzalo', 'Higuain', 'Argentina'],
      ['Giorgio', 'Chiellini', 'Italia']
    ];

    $handler = fopen($filename, 'w');

    foreach($users as $user) {
      fputcsv($handler, $user, ';');
    }
    fclose($handler);
  ?>
</body>
</html>
```

Per **creare o scrivere un file** search CSV in PHP è possibile utilizzare la funzione **fputcsv()** che prende come argomenti il puntatore al file e un array di stringhe da scrivere nel file.

Tramite un **foreach** scriviamo nel file tutti gli elementi dell'array \$users, separando i valori tramite il punto e virgola. È possibile specificare un terzo parametro (opzionale) della funzione, ossia il carattere delimitatore. Di default è la virgola, mentre nell'esempio abbiamo utilizzato il punto e virgola.

LEGGERE UN FILE CSV

```
<!DOCTYPE html>
<head>
</head>
<body>
  <?php
    $filename = 'users.csv';
    $handler = fopen($filename, 'r');

    while($data = fgetcsv($handler))
    {
        $users[] = implode(' ', $data);
    }
    fclose($handler);
    print_r($users);
  ?>
</body>
</html>
```

Leggere un file search **CSV** è altrettanto semplice. PHP mette a disposizione la funzione **fgetcsv()** che preso un puntatore al file restituisce una riga letta dal file in questione oppure false.

LEGGERE UNA STRINGA CSV

```
<?php
$values = '10,89,100,34,15';
$csv = str_getcsv($values);
print_r($csv);
?>
```

PHP consente di leggere anche una singola stringa CSV tramite la funzione `str_getcsv()`