



Informatica **U**manistica

# PHP e i files

Laboratorio di Progettazione Web

*AA 2009/2010*

*Chiara Renso*

*ISTI- CNR - c.renso@isti.cnr.it*



UNIVERSITÀ DI PISA

# FILES in PHP

Da PHP e' possibile accedere e manipolare i file che risiedono nel disco del server.

In particolare e' possibile:

- ◆ Creare, scrivere e leggere un file
- ◆ Fare l'upload di un file

# Azioni su files

1. Creazione o apertura - `fopen`
2. Scrittura / lettura - `fwrite`, `fread`, `fgets`
3. Chiusura - `fclose`

# Creare o aprire Files

La funzione

*Filehandler = fopen(filename, modalita')*

permette di aprire un file. Se il file non esiste, lo crea. Come argomenti ha il nome del file e la modalita' di apertura. Restituisce un numero identificativo del file chiamato filehandler.

```
<?php
```

```
$myFile = "testFile.txt";
```

```
$MyFileHandler = fopen($myfile, 'w') ;
```

```
fclose($MyFileHandler );    ?>
```

## fopen: modalita' di apertura file

**Letture: 'r'**. Apre un file solo per la lettura. Il puntatore di inizio lettura e' all'inizio del file. **'r+'** apre in lettura e scrittura.

**Write: 'w'** . Apre un file solo per la scrittura. Attenzione, i dati del file vengono cancellati e la scrittura inizia all'inizio del file, in quanto il puntatore di scrittura viene posizionato all'inizio del file. . **'w+'** apre in lettura e scrittura.

**Append: 'a'** Apre un file in scrittura, ma i dati vengono preservati e la scrittura inizia al termine del file. Il puntatore di scrittura stavolta e' alla fine del file. . **'a+'** apre in lettura e scrittura.

# Chiudere un file

Dopo l'apertura e la scrittura o lettura di un file, il file va sempre chiuso.

La funzione per chiudere i file è'

`fclose(filehandler)`

## Scrivere su un file

La funzione che permette di scrivere su un file e'

`fwrite(filehandler,stringa)`

**Il file va preventivamente aperto in scrittura e il file handler risultante deve essere il primo argomento della funzione.**

## Esempio: scrivere su un file

```
<?php
```

```
$myFile = "testFile.txt";
```

```
$fh = fopen($myFile, 'w');
```

```
$frase1 = "scrivo questa stringa e vado a capo\n";
```

```
fwrite($fh, $frase1);
```

```
$frase2 = "scrivo una seconda frase e vado a capo\n";
```

```
fwrite($fh, $frase2);
```

```
fclose($fh); ?>
```

## Appendere ad un file

```
<?php
```

```
$myFile = "testFile.txt";
```

```
$fh = fopen($myFile, 'a');
```

```
$frase1 = "scrivo questa nuova frase\n";
```

```
fwrite($fh, frase1);
```

```
$frase2 = "altra nuova frase\n";
```

```
fwrite($fh, $frase2);
```

```
fclose($fh); ?>
```

# Leggere un file

La lettura di un file (deve preventivamente essere aperto in lettura ) avviene principalmente con due funzioni:

`fread(filehandler, integer)` oppure `fgets(filehandler)`

L'intero indica quanti caratteri leggere dal puntatore di inizio lettura. In `fgets` e' opzionale in quanto legge l'intera riga.

Per leggere l'intero file occorre indicare l'intera lunghezza del file, che si puo' ottenere con la funzione

`filesize(nomefile)`

## Esempio: Leggere un file con fread

```
<?php
```

```
$myFile = "testFile.txt";
```

```
$fh = fopen($myFile, 'r');
```

```
$leggo = fread($fh, 5);
```

```
fclose($fh);
```

```
echo $leggo;
```

```
?>
```

## Esempio: Leggere un intero file con fread

```
<?php
```

```
$myFile = "testFile.txt";
```

```
$fh = fopen($myFile, 'r');
```

```
$leggo = fread($fh, filesize($myfile));
```

```
fclose($fh);
```

```
echo $leggo;
```

```
?>
```

## Esempio: Leggere un file con fgets

```
<?php  
  
$file = fopen("testFile.txt","r");  
  
while (!feof($file)) {  
    echo fgets($file);  
}  
  
fclose($file);  
  
?>?>
```

# Leggere un file in un array

La funzione `file("nome_file")` restituisce un array con gli elementi uguali ad ogni riga del file di testo.

```
<?php
```

```
$myfilename="miofile.txt";
```

```
$myarray=file($myfilename);
```

```
foreach ($myarray as $item)
```

```
echo $item;
```

```
fclose($fh); ?>
```

# Cancellare un file

Un file si cancella con la funzione

`unlink(nomefile)`

```
<? php
```

```
$myFile = "testFile.txt";
```

```
unlink($myFile);
```

```
?>
```

## Upload di file sul server

**E' di uso comune per la applicazioni web poter "uploadare" file (ad esempio foto, filmati, testo) dal client verso il server.**

**Nel client selezionamo il file da spedire al server tramite una form, il server riceve il file tramite uno script PHP e lo salva nel suo file system.**

**Ovviamente lo script PHP deve essere in grado di reperire il file selezionato dall'utente e di manipolarlo (ad esempio spostarlo in una determinata cartella etc)**

## Upload di file: la form HTML

```
<form enctype="multipart/form-data"  
  action="uploader.php" method="POST">
```

```
<input type="hidden" name="MAX_FILE_SIZE"  
  value="100000" />
```

Scegli un file da caricare:

```
<input name="uploadedfile" type="file" /><br />
```

```
<input type="submit" value="Upload File" />
```

```
</form>
```

# Uploader.php

- ◆ All'esecuzione di uploader.php il file esiste già ed è in una locazione temporanea nel server. Occorre spostare il file se non vogliamo perderlo!
- ◆ I file caricati vengono memorizzati nell'array globale `$_FILES`. In particolare abbiamo:
- ◆ `$_FILES['uploadedfile']['name']` - name contiene il path original del file caricato
- ◆ `$_FILES['uploadedfile']['tmp_name']` - tmp\_name contiene il path del file temporaneo sul server.

# Uploader.php

```
<?php
```

Cartella dove voglio salvare I  
file caricati

```
$target_path = "uploads/";
```

```
$target_path = $target_path.basename  
( $_FILES['uploadedfile']['name']);
```

Costruisco il  
path finale

```
if (move_uploaded_file($_FILES['uploadedfile']  
['tmp_name'], $target_path)) {
```

```
echo "Il file ".basename( $_FILES['uploadedfile']  
['name'])."e stato caricato";    }
```

```
else { echo "errore nel caricamento del file,  
riprova!";}
```

# Esercizi

- ◆ Eseguire gli esempi di questi lucidi sulla scrittura e lettura dei files
- ◆ Creare una form HTML che riceva dall'utente un nome (campo text NOME) e un testo (da un campo textarea TESTO). Lo script PHP dovrà scrivere il testo passato in TESTO in un nuovo file che avrà come nome il nome inserito all'utente nel campo NOME.
- ◆ Creare una nuova form per l'upload di un file da parte dell'utente. Verificare il funzionamento dello script effettuando varie prove.