PHP ESSENTIALS #8

By WI400 Team

: files, streaming

files

- leggere da ifs
- scrivere su ifs
- PHP 5 magic

Files: leggere dal file/system

- Accedere ai files significa accedere al filesystem del server nel quale il PHP viene eseguito
- Non è possibile attraverso i linguaggi lato server scrivere file direttamente sul filesystem dei client
- Per poter accedere ad un file del filesystem è necessario ottenere un handle di file attraverso la seguente istruzione:

```
<?php
$handle = fopen('/path/to/file.txt', 'r');
```

- Molte delle funzioni per leggere e scrivere su file utilizzano l'handle di file

Files: leggere dal file/system

- Il primo argomento passato alla funzione fopen() è il file da aprire
- Il secondo argomento passato alla funzione fopen() è la modalità di apertura del file
- Esistono diverse modalità di apertura di un file tra le quali:

  r – read

  w – write

  a - append

Files: leggere dal file/system

- La funzione fread() permette la lettura da un file precedentemente aperto con la funzione fopen()

```php
<?php
$file = 'studenti.txt';
$handle = fopen($file, 'r');
$testo = fread($handle,1024);
fclose($handle);
```

Files: leggere dal file/system

- Il primo argomento passato alla funzione fread() è l'handle di file
- Il secondo argomento passato alla funzione fread() è la quantità di dati in byte da leggere
- E' possibile combinare la lettura in un ciclo while attraverso l'utilizzo della funzione feof():

```
<?php
$handle = fopen('studenti.txt', 'r');
$contenuto = '';
while (!feof($handle))
{
    $contenuto .= fread($handle, 1024);
}
fclose($handle);
```

Files: leggere dal file/system

- La funzione filesize() restituisce la dimensione di un file

- Accetta come unico parametro il nome del file

```
echo filesize('studenti.txt');
```

Esercizio 14:

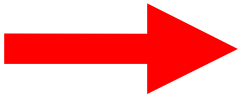
- crea un file di testo students.txt con in ogni riga:
  - nome
  - cognome
  - email

  ogni campo è delimitato da ';'
- crea uno script che legga il file e che faccia l'output di ogni studente così:
  “Chris Shiflett” <chris@brainbulb.com>
- *nota: usa le funzioni php per l'htm per visualizzare correttamente i caratteri speciali* ***<*** *per < e* ***>*** *per >.*

Esercizio 14: solution

- fopen() sul file e lettura completa
- scomposizione della stringa in array
- foreach sull'array per ogni riga e stampa risultato

```php
<?php
$file = 'students.txt';
$handle = fopen ( $file, 'r' );
$text = fread ( $handle, filesize ( $file ) );
fclose ( $handle );
$students = explode ( "\n", $text );
foreach ( $students as $student ) {
    list ( $first, $last, $email ) = explode ( ';', $student );
    echo "<p>\"$first $last\"  &lt;$email&gt;</p>";
}
```



"pico depaperis" <pico@depaperis.com >
"paolino paperino" <paperino@pape.com >
"foo bar" <info@alfa.com>

Files: leggere dal file/system

- La funzione fgets() è un'altra funzione utile per leggere da un file
- Accetta due parametri:
  - ✓ L'handle di file
  - ✓ La quantità di dati da leggere in byte
- La sintassi è la stessa di fread(), ma fgets() termina la lettura quando incontra la fine della riga o la fine del file.
- Molto utile per leggere il file una riga alla volta

Files: leggere dal file/system

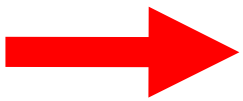
- La funzione file() restituisce un array composto da tanti elementi per quante sono le righe del file
- Ogni elemento dell'array contiene una riga del file
- La funzione accetta come parametro il file da leggere
- Non necessita di un handle al file. Questo significa che non è necessario aprire e chiudere il file da leggere:

```
$studenti = file('studenti.txt');
```

Files: leggere dal file/system

- file() sul file e ottengo direttamente l'array
- foreach sull'array per ogni riga e stampa risultato

```php
<?php
$students = file ( 'students.txt' );
foreach ( $students as $student ) {
    list ( $first, $last, $email ) = explode ( ';', $student );
    echo "<p>\"$first $last\" &lt;$email&gt;</p>";
}
```



"pico depaperis" <pico@depaperis.com >
"paolino paperino" <paperino@pape.com >
"foo bar" <info@alfa.com>

Files: scrivere sul file/system

- La scrittura su file risulta abbastanza semplice una volta acquisita familiarità con la lettura.
- Abbiamo bisogno di un handle al file da scrivere, ma dobbiamo utilizzare come modalità di apertura del file w (write) oppure a (append) al momento in cui apriamo l'handle al file:

```
$handle = fopen('/path/to/file.txt', 'w');
```

Files: scrivere sul file/system

- La funzione fwrite() scrive una stringa sul file precedentemente aperto
- Accetta due parametri:
  - ✓ L'handle al file
  - ✓ Il contenuto da scrivere
- Un esempio di utilizzo della funzione fwrite() è il seguente:

```
fwrite($handle, 'La mia prima scrittura su file');
```

Esercizio 15:

- crea un file fwrite.php che riceve i dati da una form e scriva l'output in un file students.csv
- questo l'HTML relativo

```
<form action="fwrite.php" method="POST">
<p>First name: <input type="text" name="first" /></p>
<p>Last name: <input type="text" name="last" /></p>
<p>Email: <input type="text" name="email" /></p>
<p><input type="submit" value="Add" /></p>
</form>
```

- *nota: il file di testo dovrà avere il ";" come separatore di campo*

Esercizio 15: solution

- reperimento dei dati attraverso il $_POST
- concatenazione dei dati
- fopen() e fwrite()

```php
<?php
$line = $_POST['first'] . ';' .
        $_POST['last']  . ';' .
        $_POST['email'];
$handle = fopen('students.csv', 'a+');
fwrite($handle, $line);
fclose($handle);

echo "il file è stato generato !";
```

Files: PHP 5 Magic

- Il PHP5 aggiunge altre importanti funzioni per la manipolazione dei file
  - file_get_contents()
  - file_put_contents()

- entrambe le funzioni manipolano i files (leggono e scrivono) senza la necessità di dover effettuare l'apertura e chiusura

Files: PHP 5 Magic

- file_get_contents() reperisce tutto il contenuto di un file in una variabile

```
<?php
$contents=file_get_contents('/path/to/file.txt');
```

- file_put_contents() sovrascrive un file con il contenuto di una variabile

```
<?php
file_put_contents('/path/to/file.txt', $contents);
```

streaming

- panoramica di base

Streaming: panoramica di base

- Avete tutte le conoscenze per poter lavorare con i file
- PHP può anche connettersi con altre risorse ed offre la possibilità di manipolare queste risorse come fossero dei file
- Gli stream possono essere così rappresentati
  - http: // www. example.org/
  - https: // www. example.org
  - ftp: // user:pass@ ftp. example.org/miofile. Txt
  - file: // path_to_my_file/file. txt

Streaming: panoramica di base

- Sono molto utili con gli URLs:

```
$html = file_get_contents('http://www.example.org/');
```

- E' possibile fare riferimento ai file del filesystem:

```
$contents = file_get_contents('/path/to/file.txt');
```

Esercizio 16:

- Creare un file chiamato google.php che legge il contenuto dell'URL http://www.google.com/.
- Utilizzando str_replace(), modificare l'action della form in modo da farla puntare al file che creeremo successivamente e che dovrà avere il nome di search.php.
- Utilizzando echo(), visualizzare la pagina, in modo che google.php abbia un aspetto simile ad http://www.google.com/ ma abbia come action della form noi stessi.
- Scrivere la parte di script PHP in modo che visualizzi il risultato della "nostra personale" ricerca.

Esercizio 16: solution

```php
<?php
if ($_POST['btnG']) {
    $search=$_POST['q'];
    echo "non puoi cercare la parola $search";
    //die();
}

$url='http://www.google.com';
$google =file_get_contents($url);
$google = str_replace('/search', $_SERVER['PHP_SELF'], $google);
$google = str_replace("img src=", "img src=$url", $google);
$google = str_replace('name=f', 'method="POST"', $google);
echo $google;
```

Files: upload

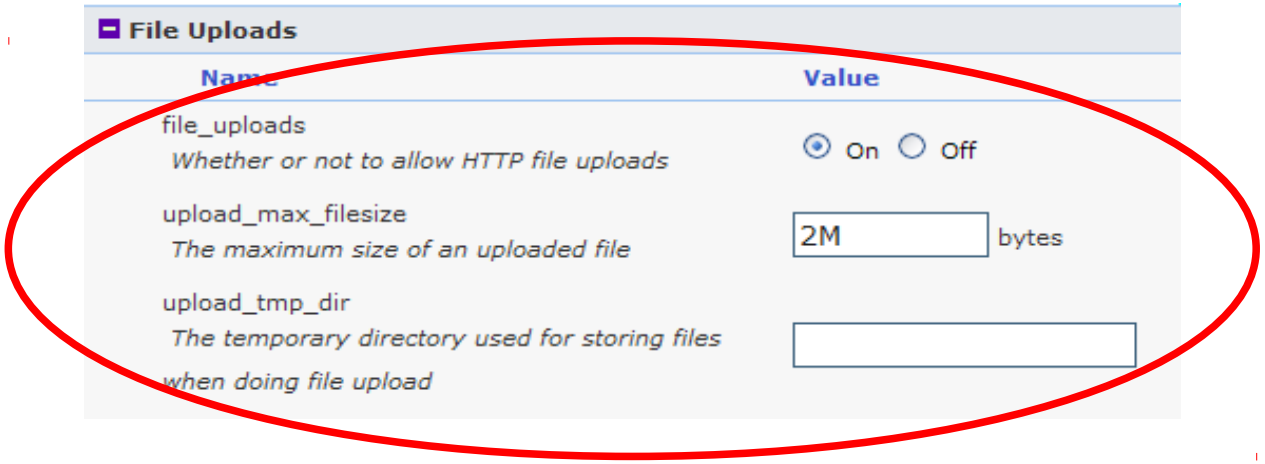
- La trasmissione di un file da “client” a “server” viene ottenuta per mezzo della funzione “upload”

- Tramite il campo di tipo “file” dell'html è possibile far selezionare all'utente un file presente nel file system del client

```
<input type="file" name="nome_file">
```

Files: upload

- In questo caso va opportunamento riportato un attributo della form enctype=”multipart/form-data”
- Inoltre va impostata, come attributo delle impostazioni del php, la directory che conterrà temporanamente il file in corso di upload



Files: upload

- Tutte le informazioni relative al file trasmesso si trovano nella variabile $_FILES
  (di tipo array di array)

- Contiene rispettivamente:
  [name] nome originale sul client
  [type] mime-type del file inviato
  [tmp_name] nome temporaneo sul web server
  [error] codice errore associato all'upload
  [size] dimensione

Files: upload

- E' possibile controllare la corretta operazione di upload tramite:

```
if (is_uploaded_file($_FILES['nome_file']['tmp_name'])) {
    echo "file caricato sul server<br>";
}
```

- Inoltre è possibile spostare il file dalla cartella temporanea ad un percorso diverso:

```
move_uploaded_file(
    $_FILES['nome_file']['tmp_name'],"path/to/file"
);
```

QUESTION TIME ?

Nome_____ Cognome_____ Data_____

ARRIVEDERCI

TITOLO