

6.1. Introducció

- Abans de l'aparició dels gestors de bases de dades, els llenguatges de programació utilitzaven els fitxers per emmagatzemar les dades de manera permanent.
- PHP disposa de diverses funcions que permeten treballar amb fitxers a diversos nivells, per aquesta raó aquest tema està organitzat en 3 apartats:
 - o Funcions per treballar amb formularis i fitxers, perquè un usuari pugui pujar fitxers al servidor
 - o Funcions per obrir fitxers per fer lectura de les dades del fitxer o escriure-hi noves dades
 - o Funcions per treballar amb el sistema de fitxers del sistema operatiu, navegar pels directoris, copiar, esborrar fitxers...

6.2. Pujar fitxers al servidor (upload)

- Upload = pujar un fitxer a un servidor
- Els formularis d'HTML disposen d'un element (`type="file"`) que permet enviar un fitxer a través d'un formulari.
- PHP ofereix una sèrie de funcions que permeten agafar aquest fitxer i guardar-lo en el servidor.
- Per implementar l'upload de fitxers haurem de seguir 3 passos:
 1. Configurar el fitxer `php.ini` per tal que accepti l'upload de fitxers
 2. Crear el formulari d'enviament del fitxer
 3. Crear l'script que recull el fitxer enviat i el guarda en el servidor

6.2.1. Configuració del fitxer `php.ini`

- Cal comprovar que el fitxer `php.ini` té activada la opció que permet l'upload
- Recordeu que en el fitxer `php.ini` hi ha la configuració de l'interpret de PHP
- S'ha d'obrir aquest fitxer i assegurar-se que hi ha les següents línies:

```
; File Uploads
file_uploads = on
upload_tmp_dir = "c:/wamp/tmp"           ; o qualsevol altre ruta
upload_max_filesize = 2M
```

- Observacions:
 - El punt i coma (;) és el símbol per fer comentaris.
 - El directori `upload_tmp_dir` és el directori on es copia la imatge temporalment, després cal moure-la a la ruta desitjada.
 - `upload_max_size = 2M` estableix que la mida màxima dels fitxers serà de 2Mbytes.

6.2.2. Creació del formulari d'enviament del fitxer

- El formulari que agafarà la imatge a pujar ha de seguir les següents instruccions:
 - o El mètode d'enviament sempre ha de ser POST
 - o El formulari ha d'incloure la propietat:

```
ENCTYPE="multipart/form-data"
```

- o Per crear l'element del formulari que permet seleccionar un arxiu, cal utilitzar:

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

- Exemple:

```
<form action="upload.php" method="POST" enctype="multipart/form-data">
    Tria el fitxer: <input type="file" name="fitxer"> <br>
    <input type="submit">
</form>
```

6.2.3. Recollir el fitxer enviat

- En la plana destí del formulari, a més de l'array associatiu `$_POST`, també existeix l'array associatiu `$_FILES` que conté tota la informació sobre el fitxer enviat.
- Les dades disponibles dins d'aquest array són:
 - o `$_FILES["fitxer"]["name"]` -> nom original del fitxer en la màquina client des d'on s'ha enviat
 - o `$_FILES["fitxer"]["type"]` -> tipus MIME del fitxer enviat. Els més habituals són: "image/gif", "image/jpeg", "image/png"
 - o `$_FILES["fitxer"]["size"]` -> mida en bytes de l'arxiu enviat
 - o `$_FILES["fitxer"]["tmp_name"]` -> nom del fitxer temporal en el que s'emmagatzema el fitxer pujat al servidor
 - o `$_FILES["fitxer"]["error"]` -> codi d'error associat al fitxer pujat
- Observació:
 - o Recorda que "fitxer" és el nom que se li ha posat a l'element de tipus "file" del formulari des d'on s'ha introduït el fitxer `<input type="file" name="fitxer">`
- Els passos que cal fer per poder guardar el fitxer en el servidor són:
 - Si s'ha pujat el fitxer al servidor:
 1. Generar un nom pel fitxer
 2. Moure el fitxer a la carpeta on el volem guardar
 - Si no:
 1. Mostrar un missatge d'error
- Per saber si s'ha pujat el fitxer al servidor, disposem de la següent funció:
 - o `is_uploaded_file("nom_temporal_fitxer")`
 - Aquesta funció rep com a paràmetre el nom temporal del fitxer
 - Retorna `true` si l'script ha rebut un fitxer, i `false` en cas contrari
- Per moure el fitxer a la carpeta on volem guardar el fitxer, disposem de la següent funció:
 - o `move_uploaded_file("nom_temporal_fitxer", "ruta_destí_i_nou_nom")`

- Aquesta funció còpia el fitxer que s'ha enviat des del formulari a la ruta i amb el nom que nosaltres indiquem.
- Rep 2 paràmetres, el primer és el nom temporal del fitxer i el segon és el nou nom que li volem posar al fitxer.
- El nou nom pot incloure una ruta diferent a l'actual.
- Observacions:
 - o Normalment es guardaran tots els fitxers pujats en una mateixa carpeta del servidor.
 - o Es recomana establir un criteri pels noms que han de rebre els fitxers pujats al servidor. De no ser així, si hi ha molts usuaris que pugen fitxers i tots es guarden en una mateixa ruta és molt fàcil que hi hagi coincidència de noms i per tant es sobreescrigui sobre fitxers ja existents.

6.3. Funcions per treballar amb fitxers

6.3.1. Obrir i tancar un fitxer

- La funció **fopen(...)** permet obrir un fitxer.
- Aquesta funció rep com a paràmetre el nom del fitxer i el mode d'obertura i retorna un descriptor (punter) al fitxer.
- Sintaxi:

```
$descriptor_fitxer=fopen("nom_fitxer.txt","mode_obertura");
```

- Quan s'obre el fitxer cal especificar el mode d'obertura, els valors que pot prendre són els següents:

MODE	DESCRIPCIÓ
"r"	Només lectura començant per l'inici
"r+"	Lectura i escriptura pel principi El fitxer ha d'existir
"w"	Només escriptura des de l'inici. Si el fitxer no existeix es crea. Si existeix, esborra tot el que hi havia anteriorment!
"w+"	Esctura i lectura des de l'inici. Si el fitxer no existeix es crea. Si s'escriu, suprimeix les dades anteriors!
"a"	Mode afegir → escriu pel final Si el fitxer no existeix, es crea.
"a+"	Esctura sempre pel final i permet llegir. Si el fitxer no existeix, es crea
"x"	Creació per escriptura des del principi. Si existeix el fitxer, dona error
"x+"	Creació per lectura i escriptura des del principi Si existeix el fitxer, dona error

- Observació:
 - o Es recomana utilitzar únicament els modes d'escriptura "r", "a" i "w".
 - o La funció **fseek(...)** permet desplaçar-se per dins el fitxer

- o La funció **`fclose(...)`** permet tancar un fitxer obert anteriorment amb la funció `fopen(...)`.
- Sintaxi:

```
fclose($descriptor_fitxer);
```

6.3.2. Lectura de fitxer

- Per poder llegir dades d'un fitxer, prèviament cal haver obert el fitxer en algun mode de lectura com ara "r", "r+", "a" o "a+".
- Existeixen diverses funcions que permeten llegir dades d'un fitxer.

a. La funció **`fgets(...)`**

- La funció **`fgets(...)`** llegeix una cadena de text de la longitud que s'especifica com a paràmetre o fins que troba un salt de línia.
- Sintaxi:

```
$longitud=10;  
$dades=fgets($descriptor_fitxer, $longitud);  
//s'han llegit 9 caràcters
```

- Observacions:
 - Com a màxim es llegirà fins al salt de línia o longitud -1 caràcters
 - *Per poder veure els salts de línia d'un fitxer es recomana que utilitzeu el WordPad i no el Bloc de notes.*

b. La funció **`fgetc(...)`**

- La funció **`fgetc(...)`** permet llegir 1 caràcter del fitxer que rep com a paràmetre.
- Sintaxi:

```
$dada=fgetc($descriptor_fitxer);
```

c. La funció `fread(...)`

- La funció `fread(...)` permet llegir una quantitat concreta de bytes del fitxer que rep com a paràmetre.
- Si s'arriba a EOF, final de fitxer, la lectura acaba.
- Sintaxi:

```
$dades=fread($descriptor_fitxer, $longitud);
```

- Observació sobre `fread(...)`:
 - Per saber la quantitat de bytes que hem de llegir podem utilitzar la funció `fsize(..)`
 - La funció `fsize(...)` retorna la longitud en bytes del fitxer que rep com a paràmetre.

Exemple:

```
$longitud=fsize("fitxer.txt");
```

- *Observeu que aquesta funció no rep un descriptor de fitxer, si no que directament rep el nom del fitxer.*

d. Funció final de fitxer

- La funció `feof(...)` retorna `true` si el descriptor del fitxer està a final de fitxer, i `false` en cas contrari.
- Sintaxi:

```
$final=feof($descriptor_fitxer);
```

- Observació:
 - Aquesta funció és útil per anar llegint dades mentre no s'arriba a final de fitxer.
 - Exemple:

```
$f=fopen("dades.txt","r");
do{
    // llegir dades
} while(!feof($f));
fclose($f);
```

- Altres funcions

- `ftell()` → retorna la posició del punter dins el fitxer

6.3.3. Escriptura a fitxer

- La funció **fwrite(...)** permet escriure dades al fitxer que s'indica com a paràmetre.
- Sintaxi:

```
fwrite($descriptor_fitxer, $dades);
```

- Observacions:
 - Per poder escriure dades a un fitxer, prèviament cal haver obert el fitxer en algun mode d'escriptura com ara "w", "w+", "a" o "a+".
 - Per escriure un salt de línia en el fitxer podem utilitzar el caràcter \n
 - Existeix una funció totalment equivalent a **fwrite(...)** anomenada **fputs(...)**.

6.3.4. Altres funcions útils

a. La funció fseek(...)

- La funció **fseek(...)** permet situar el punter del fitxer en qualsevol posició del fitxer, per poder triar on escriure o d'on llegir
- Sintaxi:

```
fseek($descriptor_fitxer, $offset,[relatiu a...])
```

- Observacions:
 - o **[relatiu a...]** és una constant opcional que pot prendre 3 valors:
 - SEEK_SET → relatiu al principi del fitxer (opció per defecte)
 - SEEK_CUR → relatiu a la posició actual del punter
 - SEEK_END → relatiu al final del fitxer
 - o **\$offset** és el desplaçament des de la posició relativa
 - o Si el fitxer està obert en mode "a+" les dades sempre s'escriuran pel final, independentment d'on estigui el punter del fitxer.

b. La funció file(...)

- La funció **file(...)** permet volcar directament tot el contingut d'un fitxer sobre un array.
- Sintaxi:

```
$array=file("fitxer.txt");
```

- Observacions:
 - A cada casella de l'array hi haurà 1 línia del fitxer
 - Per poder mostrar les dades caldrà recórrer l'array com es va veure en els temes anteriors.
 - Fixeu-vos que a la funció **file(...)** no se li passa un descriptor de fitxer si no directament el nom del fitxer!
 - Per tant, quan s'utilitza aquesta funció no s'ha d'obrir el fitxer prèviament

6.4. Funcions per treballar amb sistema de fitxers

- En aquest apartat es veuran les principals funcions que ofereix PHP per treballar amb el sistema de fitxers del sistema operatiu.
- No totes les funcions tenen sentit en tots els sistemes operatius, des d'aquí s'intentarà només comentar les que són multiplataforma (funcionen igual en Linux i en Windows)
- Cal recordar que el caràcter \ és un caràcter de protecció, per tant per prevenir errors es recomana utilitzar sempre el caràcter /

6.4.1. Funcions per treballar amb directoris

- Les principals funcions per treballar amb directoris són:

Funció	Observacions
<code>string getcwd(void)</code>	Retorna el directori de treball actual (<i>getwd=Get Current Working Direcotry</i>)
<code>bool chdir(string directori)</code>	Canvia a un altre directori Retorna <i>true</i> si la operació s'executa correctament Retorna <i>false</i> en cas contrari
<code>array scandir(string dir, [int order])</code>	Retorna un array amb la llista de fitxers i directoris d'una ruta concreta <i>order = 0 → ordre alfabetic ascendent (per defecte)</i> <i>order = 1 → ordre invers</i>
<code>bool mkdir(string \$path)</code>	Permet crear una carpeta <ul style="list-style-type: none"> - Retorna <i>true</i> si OK - Retorna <i>false</i> en cas contrari
<code>bool rmdir(string \$path)</code>	Permet eliminar una carpeta <ul style="list-style-type: none"> - Retorna <i>true</i> si OK - Retorna <i>false</i> en cas contrari

- **Exemple 1:**

Mostra la ruta actual, entra a la carpeta "exemple", torna a mostrar la ruta i llista els fitxers que hi ha

```
<?php
echo getcwd() . "\n";
chdir('exemple'); //Entrem al directori "exemple"
echo getcwd() . "\n";
$a=scandir(getcwd()); //Fem llistat fitxers del directori actual
$num = count($a);
echo "Numero de fitxers: $num <BR>";
echo $a[0]."<BR>"; //Aquí caldria fer un bucle per llistar tots
els fitxers
...
?>
```

6.4.2. Funcions per copiar, esborrar i reanomenar elements

- Les principals funcions que permeten copiar, esborrar i reanomenar fitxers són:

Funció	Observacions
<code>bool copy(string \$source, string \$dest)</code>	Permet copiar un fitxer <ul style="list-style-type: none"> Retorna <i>true</i> si operació OK Retorna <i>false</i> en cas contrari
<code>bool unlink(string \$filename)</code>	Permet eliminar un fitxer El nom del fitxer pot contenir tota la ruta <ul style="list-style-type: none"> Retorna <i>true</i> si operació OK Retorna <i>false</i> en cas contrari
<code>bool rename(string \$oldname, string \$newname)</code>	Permet reanomenar un fitxer <ul style="list-style-type: none"> Retorna <i>true</i> si operació OK Retorna <i>false</i> en cas contrari
<code>bool file_exists(string \$fitxer)</code>	Comprova si el fitxer existeix <ul style="list-style-type: none"> Retorna <i>true</i> si el fitxer existeix Retorna <i>false</i> en cas contrari

- Exemple 2:**

Copia el fitxer "exemple.txt" en la mateixa ruta i anomenant-lo "exemple2.txt".

Després elimina el fitxer antic.

Finalment renombra el nou fitxer amb un altre nom

```
<?php
    $oldfile = 'exemple.txt';
    $newfile = 'exemple2.txt';
    if (!copy($oldfile, $newfile)) {
        echo "Error copiant el fitxer $file...\n";
    }else{
        unlink($oldfile);
    }

    rename($newfile, "nounom.txt");
?>
```

6.4.3. Funcions auxiliars

- Existeix un conjunt de funcions que aporten funcionalitats auxiliars a les funcions ja esmentades:

Funció	Observacions
<code>bool file_exists(string \$path)</code>	<p>Comprova si el fitxer o directori existeix</p> <ul style="list-style-type: none"> Retorna <i>true</i> si operació OK Retorna <i>false</i> en cas contrari <p>Normalment s'utilitza abans de fer qualsevol operació sobre un fitxer, perquè no doni error en cas que el fitxer no existeixi.</p> <p><i>Abans de copiar, esborrar o reanomenar un fitxer cal sempre comprovar que existeixi.</i></p>
<code>string dirname(string \$path)</code>	Retorna el directori pare (absolut) de la ruta actual
<code>float disk_free_space(string \$particio)</code>	<p>Retorna la quantitat de bytes disponibles en la partició actual. Si es passa una ruta qualsevol, retornarà els bytes disponibles de la partició</p> <ul style="list-style-type: none"> Exemple Windows: <code>echo disk_free_space("C:");</code> Exemple Linux: <code>echo disk_free_space("/");</code>
<code>float disk_total_space(string \$path)</code>	Retorna la mida de la partició actual (bytes lliures + bytes ocupats)
<code>string basename(string \$path)</code>	<p>Retorna el nom de la última carpeta o fitxer d'una ruta. Exemples:</p> <ul style="list-style-type: none"> <code>C:\Dades\jv\fitxer.txt</code> → <code>fitxer.txt</code> <code>C:\Dades\jv\</code> → <code>jv</code> <p><i>És el cas contrari que <code>dirname</code></i></p>
<code>string dirname(string \$path)</code>	<p>Retorna tot el path de la ruta passada, excepte el nom del fitxer o el directori actual</p> <p>Exemples:</p> <ul style="list-style-type: none"> <code>C:\Dades\jv\fitxer.txt</code> → <code>c:\Dades\jv</code> <code>C:\Dades\jv\</code> → <code>c:\Dades</code> <p><i>És el cas contrari que <code>basename</code></i></p>
<code>int filesize(string \$fitxer)</code>	Retorna la mida en bytes del fitxer passat
<code>bool is_dir(string \$path)</code>	<p>Retorna <i>true</i> si la ruta passada és un directori</p> <p>Retorna <i>false</i> en cas contrari</p>
<code>bool is_file(string \$path)</code>	<p>Retorna <i>true</i> si la ruta passada és un fitxer</p> <p>Retorna <i>false</i> en cas contrari</p>
<code>Int filemtime(string \$path)</code>	<p>Retorna quan es va fer la última modificació d'un fitxer en format timestamp (segons des d'1 gener 1970)</p> <p><i>Veure exemple 3</i></p>
<code>array pathinfo(string \$path)</code>	<p>Retorna un array amb diversa informació sobre la ruta passada (dirname, basename, extension, filename)</p> <p><i>Veure exemple 4</i></p>

- **Exemple 3:** Mostrem la data de modificació d'un arxiu

```
<?php
$time= filemtime('c:/wamp/www/tema6/f.txt');
echo "Segons: $time <BR>";
echo date ("F d Y H:i:s ", $time); //transformem segons en una data
?>
```

- **Exemple 4:** Mostrem informació de la ruta passada a través de la funció "pathinfo(...)"

```
<?php
$arr= pathinfo('c:/wamp/www/tema6/f.txt');
echo $arr['dirname']."<BR>\n"; // → c:/wamp/www/tema6
echo $arr['basename']."<BR>\n"; // → f.txt
echo $arr['extension']."<BR>\n"; // → txt
echo $arr['filename']."<BR>\n"; // → f
?>
```