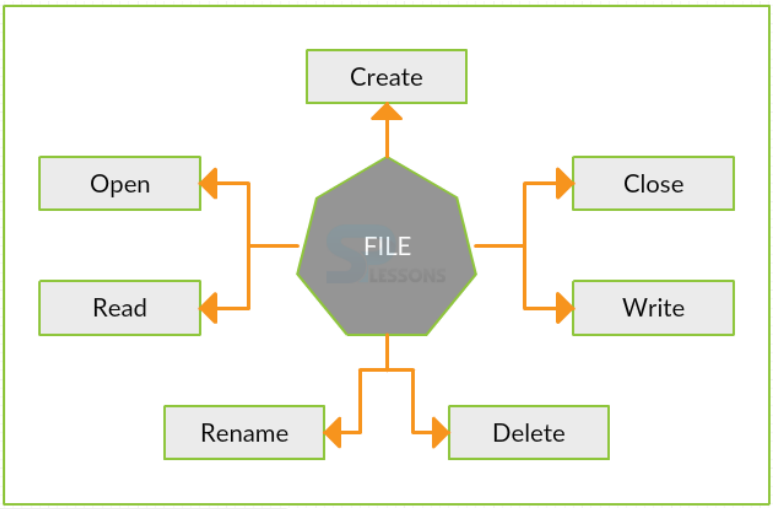
CAP. 7
MANIPULAREA FIȘIERELOR ÎN PHP

Când scriem cod PHP, acesta este salvat într-un fișier cu extensia ".php". De asemenea, pentru informații ce nu au neapărat o structură bine definită, adesea se folosesc tot fișiere (de cele mai multe ori de tip text). Spre exemplu, atunci când accesăm o pagină, erorile ce pot apărea sunt adesea salvate într-un fișier. Când avem diverse configurații, informația este citită dintr-un fișier, drept urmare fișierele pentru lumea PHP și sistemele de operare, în general, sunt foarte importante.

Principalele operații ce pot fi realizate asupra fișierelor sunt reprezentate în figura alăturată.



7.1 Utilizarea mai multor fișiere PHP

Putem include conținutul unui fișier PHP într-un alt fișier PHP pentru utiliza funcționalitățile implementate în acesta. În PHP există 2 funcții care pot fi folosite pentru a include un fișier în alt fișier PHP:

- include()
- require()

Observație: Dacă este necesară o modificare a unei funcționalități, atunci această modificare se va realiza doar în fișierul în care este implementată, rareori fiind necesară modificarea locațiilor în care este utilizată.

Funcția include()

Funcția **include** integrează conținutul din fișierul primit ca parametru în cadrul fișierului în care se apelează funcția. Dacă apare orice problemă în încărcarea fișierului, atunci funcția **include()** va genera o avertizare, dar scriptul va continua execuția.

Să presupunem că dorim să creăm un meniu comun pentru website. Atunci creăm un fișier **menu.php** cu următorul conținut:

```
<a href="https://www.atelierelеilbah.ro/cursuri/curs-programare-it-full-stack-developer-front-end-back-end/"></a>
<a href="https://www.atelierelеilbah.ro/cursuri/curs-administrator-baze-de-date/"></a>
<a href="https://www.atelierelеilbah.ro/cursuri/curs-administrator-retea/"></a>
<a href="https://www.atelierelеilbah.ro/cursuri/curs-utilizarea-calculatorului/"></a>
```

După, creăm câte pagini dorim și includem fișierul anterior în antet. De exemplu, fișierul **index.php** poate avea următorul conținut:

```
<html lang="en">
<body>
    <?php include("menu.php"); ?>
    <h1>Prima Pagina</h1>
</body>
</html>
```

Observație: În momentul în care includem un fișier trebuie să ne asigurăm că menționăm corect referința către locația în care acesta se găsește, fie folosind o cale relativă la fișierul în care este inclus, fie folosind o cale absolută (mai rar întâlnit).

Funcția require()

Funcția **require** funcționează relativ similar cu funcția include, în sensul că integrează conținutul din fișierul primit ca parametru în cadrul fișierului în care se apelează funcția. Diferența constă în faptul că dacă apar excepții la execuția sau încărcarea fișierului, atunci funcția require va genera o eroare fatală și va opri execuția scriptului.

Este recomandată folosirea funcției require în locul funcției include deoarece scripturile nu trebuie să continue execuția dacă acestea generează o eroare ce poate fi mai târziu fatală aplicației.

Pentru testarea funcției require putem folosi exemplul anterior de la funcția include și va genera același rezultat dacă fișierul există și calea către acesta este menționată corect.

Dacă, în schimb fișierul nu există, atunci rezultatul va fi diferit. Rezultatul obținut în cazul în care folosim funcția **include**, iar fișierul **menu.php** nu există:

```
Warning: include(menu.php): failed to open stream: No such file or
directory in C:\laragon\www\test.php on line 6

Warning: include(): Failed opening 'menu.php' for inclusion
(include_path='.;C:/laragon/etc/php/pear')
in C:\laragon\www\test.php on line 6
```

Prima Pagina

Rezultatul obținut în cazul în care folosim funcția require, iar fișierul menu.php nu există, este următorul:

```
Warning: require(menu.php): failed to open stream: No such file or
directory in C:\laragon\www\test.php on line 6

Fatal error: require(): Failed opening required 'menu.php'
(include_path='.;C:/laragon/etc/php/pear')
in C:\laragon\www\test.php on line 6
```

După cum se poate observa, în cazul funcției require, informația conținută de index.php nu mai este procesată și mesajul “Prima Pagina” nu mai apare.

7.2 Operații cu fișiere

Principalele operații ce pot fi realizate asupra unui fișier sunt următoarele:

- Deschiderea unui fișier;
- Citirea unui fișier;
- Scrierea într-un fișier;
- Închiderea unui fișier.

Observație: Aveți grijă când manipulați fișierele! Puteți face multe daune dacă faceți ceva greșit. Erorile obișnuite sunt: editarea fișierului greșit, umplerea unui hard-disk cu date eronate sau ștergerea accidentală a conținutului unui fișier.

7.2.1 Funcții pentru deschiderea, citirea și închiderea fișierelor

Citirea dintr-un fișier folosind PHP necesită o serie de pași:

- Deschiderea fișierului folosind funcția **fopen()**;
- Aflarea lungimii fișierului folosind funcția **filesize()**;
- Citirea conținutul fișierului folosind funcția **fread()**;
- Închiderea fișierul folosind funcția **fclose()**.

Funcția **fopen** este folosită pentru a deschide un fișier. La apelarea acesteia sunt necesare două argumente: numele fișierului și modul în care va fi manipulat fișierul. Există 8 moduri de manipulare a fișierelor cu ajutorul funcției fopen, prezentate în tabelul următor.

Mod	Descriere
r	Deschide un fișier numai pentru citire. Indicatorul de citire a fișierului este poziționat la începutul fișierului.
w	Deschide un fișier numai pentru scriere. Șterge conținutul fișierului sau creează un fișier nou dacă acesta nu există. Indicatorul fișierului este poziționat la începutul fișierului.
a	Deschide un fișier numai pentru scriere. Datele existente în fișier sunt păstrate. Indicatorul fișierului este poziționat la sfârșitul fișierului. Creează un fișier nou dacă fișierul nu există.
x	Creează un fișier nou numai pentru scriere. Returnează FALSE și o eroare dacă fișierul există deja.
r+	Deschide un fișier pentru citire/scriere. Indicatorul este poziționat la începutul fișierului.
w+	Deschide un fișier numai pentru citire/scriere. Șterge conținutul fișierului sau creează un fișier nou dacă acesta nu există. Indicatorul este poziționat la începutul fișierului.
a+	Deschide un fișier numai pentru citire/scriere. Datele existente în fișier sunt păstrate. Indicatorul este poziționat la sfârșitul fișierului. Creează un fișier nou dacă fișierul nu există.
x+	Creează un fișier nou numai pentru citire/scriere. Returnează FALSE și o eroare dacă fișierul există deja.

Structura funcției **fopen** este următoarea:

```
fopen(
    string $filename,
    string $mode,
    bool $use_include_path = false,
    ?resource $context = null
): resource|false
```

Exemplu de utilizare a funcției fopen pentru a citi conținutul unui fișier:

```
<?php
$myfile = fopen("fisier.txt", "r") or die("Fisierul nu poate fi
deschis!");
echo fread($myfile,filesize("fisier.txt"));
fclose($myfile);
```

Dacă încercam să deschidem un fișier iar acesta nu există, funcția va returna **false**, altfel va returna o referință către fișier ce poate fi folosită pentru a citi sau scrie în acel fișier.

Funcția **fread** citește dintr-un fișier deschis. Structura acesteia este următoarea:

```
fread(resource $stream, int $length): string|false
```

Primul parametru al **fread** conține numele fișierului din care se citește, iar al doilea parametru specifică numărul maxim de octeți de citit. Lungimea fișierelor poate fi găsită folosind funcția **filesize** care primește numele fișierului ca și argument.

După ce terminăm operațiile de citire/scriere din/în fișier este important să închidem fișierul folosind funcția **fclose**. Structura funcției fclose este următoarea:

```
fclose(resource $stream): bool
```

Observație: Dacă avem erori la execuția funcțiilor de citire/scriere, trebuie să ne asigurăm că la nivel de server avem permisiunea să citim/scriem informații din/în acel fișier pe hard-disk.

7.2.2 Funcții pentru crearea și scrierea în fișiere

Funcția **fopen** este folosită și pentru a crea un fișier. Poate un pic confuz, dar în PHP, un fișier este creat folosind aceeași funcție folosită pentru deschiderea fișierelor. Dacă utilizați fopen pe un fișier care nu există, aceasta îl va crea, dar cu condiția ca fișierul să fie deschis pentru scriere (w) sau adăugare (a).

Exemplul de mai jos creează un fișier nou numit „testfile.txt”. Fișierul va fi creat în același director în care se află fișierul PHP din care a fost apelată funcția **fopen**:

```
$myfile = fopen("testfile.txt", "w")
```

Funcția **fwrite** este folosită pentru a scrie într-un fișier. Structura funcției este următoarea:

```
fwrite(resource $stream, string $data, ?int $length = null): int|false
```

Primul parametru al **fwrite** conține numele fișierului în care doriți să scrieți, iar al doilea parametru este șirul de caractere ce urmează să fie scris. Opțional al treilea argument este un număr întreg prin care se specifică lungimea șirului de caractere ce urmează a fi scris în fișier.

Exemplu de scriere în fișier:

```
$myfile = fopen("newfile.txt", "w") or die("Unable to open file!");
$txt = "John Doe\n";
fwrite($myfile, $txt);
$txt = "Jane Doe\n";
fwrite($myfile, $txt);
fclose($myfile);
```

Observați că în exemplul de mai sus se scrie în fișierul "newfile.txt" de două ori. De fiecare dată când se scrie în fișier se menționează referința către fișier și referința către informația ce urmează să fie adăugată, prin variabila **$txt**. După finalizarea operațiunilor de scriere, închiderea fișierului se realizează folosind funcția **fclose**.

Observație: În exemplul anterior fișierul este suprascris datorita utilizării opțiunii "w". Puteți adăuga date într-un un fișier utilizând modul „a". Modul „a" adaugă text la sfârșitul fișierului, în timp ce modul „w" suprascrie (și șterge) conținutul vechi al fișierului.

7.3 Funcții pentru gestionarea fișierelor

De cele mai multe ori în manipularea fișierelor ne putem lovi de o serie de probleme ce privesc manipularea acestora în cadrul sistemului de fișiere. Pentru a rezolva aceste probleme putem apela la o parte din funcțiile puse la dispoziție de PHP și enumerate în tabelul următor.

Funcție	Descriere
`basename(string $path, string $suffix = ""): string`	Returnează numele (cu tot cu extensie) al unui fișier (excluzând calea)
`chmod(string $filename, int $permissions): bool`	Schimbă permisiunile asupra unui fișier
`chown(string $filename, string|int $user): bool`	Schimbă proprietarul unui fișier
`copy(string $from, string $to , ?resource $context = null): bool`	Folosită pentru copierea unui fișier. Returnează **true** dacă reușește sau **false** în caz contrar
`dirname(string $path, int $levels = 1): string`	Returnează directorul părinte al unui fișier
`file_exists(string $filename): bool`	Verifică dacă un director sau fișier există sau nu
`fileowner(string $filename): int|false`	Returnează id-ul proprietarului unui fișier
`fileperms(string $filename): int|false`	Afișează permisiunile asupra unui fișier
`filesize(string $filename): int|false`	Returnează dimensiunea fișierului în octeți
`filetype(string $filename): string|false`	Returnează tipul fișierului. Dacă nu recunoaște tipul returnează false
`is_readable(string $filename): bool`	Returnează **true** dacă fișierul există și poate fi citit

`is_writable(string $filename): bool`	Returnează **true** dacă fișierul există și se poate scrie în el
`mkdir( string $directory, int $permissions = 0777, bool $recursive = false, ?resource $context = null ): bool`	Creează un director, sau o cale întreagă, putând fi specificate și permisiunile asupra acestuia
`unlink(string $filename, ?resource $context = null): bool`	Șterge fișierul specificat

Teme

1. Realizați o funcție care primește ca parametri un nume de fișier, o frază și un număr. Aceasta va scrie în fișier fraza primită ca parametru de un număr de ori egal cu numărul primit ca parametru.
2. Realizați o funcție ce primește doi parametri (două nume de fișiere). Funcția va verifica dacă primul fișier există, și dacă da, va copia conținutul acestuia la finalul celui de-al doilea fișier.
3. Realizați o funcție ce primește ca parametru o cale (director) din sistemul de fișiere. Aceasta va căuta în acel director toate fișierele cu extensia **.doc**. Funcția va crea un fișier cu numele **fisiere_text.txt** și va scrie în acesta numele fișierelor găsite.