

# PHP : fichiers

**Achref El Mouelhi**

Docteur de l'université d'Aix-Marseille  
Chercheur en programmation par contrainte (IA)  
Ingénieur en génie logiciel

`elmouelhi.achref@gmail.com`



# Plan

- 1 Introduction
- 2 Ouverture
- 3 Fermeture
- 4 Écriture
- 5 Lecture
- 6 Autres opérations sur les fichiers

## Fichiers

- Utilisés pour stocker et/ou échanger les données
- Pouvant être en écriture ou en lecture

© Achref EL MOUËZ

# PHP

## Fichiers

- Utilisés pour stocker et/ou échanger les données
- Pouvant être en écriture ou en lecture

## 3 étapes pour la manipulation de fichiers en **PHP**

- Ouverture
- Utilisation : écriture ou lecture
- Fermeture

## Syntaxe

```
$mon_logique = fopen (nom_physique, mode_ouverture);
```

## Modes d'ouverture

- `a` : ouvre le fichier, s'il existe, en écriture seule place le pointeur à la fin du fichier. S'il n'existe pas, il sera automatiquement créé.
- `a+` : ouvre le fichier en lecture et écriture. Si le fichier n'existe pas, il sera automatiquement créé. S'il existe, le texte sera ajouté à la fin.
- `r` : ouvre le fichier en lecture seule et place le pointeur au début du fichier. Si le fichier n'existe pas, il ne sera pas créé.
- `r+` : ouvre le fichier en lecture et écriture, et place le pointeur au début du fichier. Si le fichier n'existe pas, il ne sera pas créé.
- `w` : ouvre le fichier en écriture seule. Si le fichier existe, son contenu est effacé et le fichier est considéré comme vide. Si le fichier n'existe pas, il est créé.
- `w+` : ouvre le fichier en lecture et écriture. Si le fichier existe, son contenu est effacé et le fichier est considéré comme vide. Si le fichier n'existe pas, il est créé.
- `x` : crée et ouvre le fichier en écriture seule. Si le fichier existe déjà, l'appel à `fopen` échoue, renvoyant `false` et générant une erreur de niveau `E_WARNING`. Si le fichier n'existe pas, il est créé.
- `x+` : crée et ouvre le fichier en lecture et écriture. Si le fichier existe déjà, l'appel à `fopen` échoue, renvoyant `false` et générant une erreur de niveau `E_WARNING`. Si le fichier n'existe pas, il est créé.
- ...

## Exemple

```
$mon_fichier = fopen("file.txt", "a");
```

© Achref EL MOUELHI

## Exemple

```
$mon_fichier = fopen("file.txt", "a");
```

On peut aussi tester si le fichier (ou dossier) existe avant de l'ouvrir

```
if (file_exists("file.txt")) {  
    $mon_fichier = fopen("file.txt", "a");  
} else {  
    echo "fichier non trouvé";  
}
```



## Pour fermer un fichier

```
fclose(nom_fichier);
```

© Achref EL MOUELHI ©

## Pour fermer un fichier

```
fclose(nom_fichier);
```

## Exemple

```
if (file_exists("file.txt")) {  
    $mon_fichier = fopen("file.txt", "a");  
    // utilisation  
    fclose($mon_fichier);  
} else {  
    echo "fichier non trouvé";  
}
```

## Quelques fonctions d'écriture

- `fwrite` : écrit une chaîne de caractère dans un fichier et retourne le nombre d'octets écrits, ou `false` si une erreur survient.
- `fputs` : alias de `fwrite`
- `file_put_contents = fopen + fwrite + fclose` : Si le fichier n'existe pas, il sera créé. Sinon, le fichier existant sera écrasé.  
Ajoutez une constante `FILE_APPEND` comme troisième paramètre pour ajouter les données à la suite...

## Exemple d'écriture dans un fichier

```
if (file_exists("file.txt")) {  
    $mon_fichier = fopen("file.txt", "a");  
    $nbr_octet = fwrite($mon_fichier, "bonjour");  
    echo $nbr_octet;  
    fclose($mon_fichier);  
} else {  
    echo "fichier non trouvé";  
}
```

© Achref EL M

## Exemple d'écriture dans un fichier

```
if (file_exists("file.txt")) {  
    $mon_fichier = fopen("file.txt", "a");  
    $nbr_octet = fwrite($mon_fichier, "bonjour");  
    echo $nbr_octet;  
    fclose($mon_fichier);  
} else {  
    echo "fichier non trouvé";  
}
```

Pour refaire la même chose avec `file_put_contents`

```
if (file_exists("file.txt")) {  
    $nbr_octet = file_put_contents("file.txt", "bonjour");  
    echo $nbr_octet;  
} else {  
    echo "fichier non trouvé";  
}
```

## Quelques fonctions de lecture

- `file_get_contents` : retourne tout le contenu d'un fichier dans une chaîne de caractères.
- `fgets` : retourne la ligne courante d'un fichier dans une chaîne de caractères, `false` s'il n'y a plus de données à lire.
- `fgetss` : retourne la ligne courante du fichier et élimine les balises HTML, `false` s'il n'y a plus de données à lire.
- `fgetc` : retourne un caractère d'un fichier, `false` s'il n'y a plus de données à lire.
- `file` : retourne un fichier sous forme d'un tableau de chaînes de caractères.
- `filemtime` : retourne la date de la dernière modification du fichier.
- `readfile` : permet de lire le contenu d'un fichier et de l'envoyer directement au flux de sortie (généralement le navigateur) sans passer par une variable ou une manipulation supplémentaire.
- ...

Pour lire tout le contenu avec `file_get_contents`

```
if (file_exists("file.txt")) {  
    $contenu = file_get_contents("file.txt");  
    echo $contenu;  
} else {  
    echo "fichier non trouvé";  
}
```

## Pour lire la première ligne du fichier

```
if (file_exists("file.txt")) {  
    $mon_fichier = fopen("file.txt", "r");  
    $contenu = fgets($mon_fichier);  
    echo ($contenu);  
    fclose($mon_fichier);  
} else {  
    echo "fichier non trouvé";  
}
```



## Exercice

Écrivez un script **PHP** qui permet de lire tout le contenu d'un fichier ligne par ligne (en utilisant `fgets`).

## Première solution

```
if (file_exists("file.txt")) {  
    $mon_fichier = fopen("file.txt", "r");  
    while ($contenu = fgets($mon_fichier)) {  
        echo $contenu . '<br>';  
    }  
    fclose($mon_fichier);  
} else {  
    echo "fichier non trouvé";  
}
```

© Achref EL MOU

## Première solution

```
if (file_exists("file.txt")) {
    $mon_fichier = fopen("file.txt", "r");
    while ($contenu = fgets($mon_fichier)) {
        echo $contenu . '<br>';
    }
    fclose($mon_fichier);
} else {
    echo "fichier non trouvé";
}
```

Deuxième solution : `feof` retourne `true` si le pointeur est à la fin du fichier, `false` sinon

```
if (file_exists("file.txt")) {
    $mon_fichier = fopen("file.txt", "r");
    while (!feof($mon_fichier)) {
        echo fgets($mon_fichier) . "<br />";
    }
    fclose($mon_fichier);
} else {
    echo "fichier non trouvé";
}
```

## Pour lire le premier caractère du fichier

```
if (file_exists("file.txt")) {  
    $mon_fichier = fopen("file.txt", "r");  
    $char = fgetc($mon_fichier);  
    echo ($char);  
    fclose($mon_fichier);  
} else {  
    echo "fichier non trouvé";  
}
```

## Exercice

Écrivez un script **PHP** qui permet de lire tout le contenu du fichier caractère par caractère (en utilisant `fgetc`).

## Première solution

```
if (file_exists("file.txt")) {  
    $mon_fichier = fopen("file.txt", "r");  
    while ($char = fgetc($mon_fichier)) {  
        echo $char . '<br>';  
    }  
    fclose($mon_fichier);  
} else {  
    echo "fichier non trouvé";  
}
```

© Achref EL MOU

## Première solution

```
if (file_exists("file.txt")) {  
    $mon_fichier = fopen("file.txt", "r");  
    while ($char = fgetc($mon_fichier)) {  
        echo $char . '<br>';  
    }  
    fclose($mon_fichier);  
} else {  
    echo "fichier non trouvé";  
}
```

## Deuxième solution

```
if (file_exists("file.txt")) {  
    $mon_fichier = fopen("file.txt", "r");  
    while (!feof($mon_fichier)) {  
        echo fgetc($mon_fichier) . "<br />";  
    }  
    fclose($mon_fichier);  
} else {  
    echo "fichier non trouvé";  
}
```

**Pour lire et placer toutes les lignes du fichier dans un tableau**

```
if (file_exists("file.txt")) {  
    print_r(file("file.txt"));  
} else {  
    echo "fichier non trouvé";  
}
```



Pour lire tout son contenu et l'afficher sans passer par une variable

```
$filename = 'fichier.txt';  
  
if (file_exists($filename)) {  
    readfile($filename);  
} else {  
    echo "Le fichier n'existe pas."  
}  
?>
```

Étant donné le fichier `file.txt` ayant le contenu suivant

```
15  
12  
17  
13  
10  
16
```

© Achref

Étant donné le fichier `file.txt` ayant le contenu suivant

```
15  
12  
17  
13  
10  
16
```

### Exercice 1

Écrivez un script **PHP** qui permet de calculer la moyenne des valeurs définies dans `file.txt`.

Étant donné le fichier `phrases.txt` ayant le contenu suivant

```
Une première phrase.  
Et voici une deuxième.  
Et encore une troisième. Ciao.
```

© Achref EL MOUL

Étant donné le fichier `phrases.txt` ayant le contenu suivant

```
Une première phrase.  
Et voici une deuxième.  
Et encore une troisième. Ciao.
```

## Exercice 2

Écrivez un script **PHP** qui permet de lire les données du fichier `phrases.txt` et d'afficher le nombre total de mots, de lignes et de phrases.

Étant donné le fichier `notes.txt` ayant le contenu suivant

```
wick 17 13 15
dalton 20 9 13
maggio 16 12 20
baggio 8 7 9
```

© Achref EL MOU

Étant donné le fichier `notes.txt` ayant le contenu suivant

```
wick 17 13 15
dalton 20 9 13
maggio 16 12 20
baggio 8 7 9
```

### Exercice 3

Écrivez un script **PHP** qui permet de lire les données du fichier `notes.txt`, calculer la moyenne de chaque personne et l'écrire avec le nom dans `moyennes.txt`.

## Quelques fonctions sur les fichiers

- `is_dir` : retourne `true` si le paramètre correspond à un dossier, `false` sinon.
- `is_file` : retourne `true` si le paramètre correspond à un fichier, `false` sinon.
- `rmdir` : permet de supprimer un dossier. Elle retourne `true` en cas de succès, `false` sinon.
- `unlink` : permet de supprimer un fichier. Elle retourne `true` en cas de succès, `false` sinon.
- `mkdir` : permet de créer un dossier. Elle retourne `true` en cas de succès, `false` sinon.
- `touch` : permet de créer un fichier. Elle retourne `true` en cas de succès, `false` sinon.
- `copy` : permet de copier un fichier. Elle retourne `true` en cas de succès, `false` sinon.
- `rename` : permet de renommer un dossier ou fichier.
- `glob` : retourne un tableau contenant les noms de dossiers et fichiers respectant un motif passé en paramètre.
- `filesize` : retourne la taille d'un fichier (nombre d'octets)
- ...



Pour récupérer le nom du répertoire de travail (CWD : Current Working Directory) du fichier utilisé

```
echo "\n" . getcwd();  
// affiche C:\wamp64\www\php-fichiers
```

© Achref EL MOU

Pour récupérer le nom du répertoire de travail (CWD : Current Working Directory) du fichier utilisé

```
echo "\n" . getcwd();  
// affiche C:\wamp64\www\php-fichiers
```

Pour changer de répertoire, on utilise `chdir()`

```
chdir("a1");  
echo "\n" . getcwd();  
// affiche C:\wamp64\www\php-fichiers\a1
```

**Pour copier un fichier (valeur de retour : booléen)**

```
copy("file.txt", "fichier.txt");
```

© Achref EL MOUL

**Pour copier un fichier (valeur de retour : booléen)**

```
copy("file.txt", "fichier.txt");
```

**Pour renommer un fichier**

```
rename("fichier.txt", "f.txt");
```

**Pour créer un répertoire (dossier)**

```
mkdir("r1");
```

© Achref EL MOUELHI ©

### Pour créer un répertoire (dossier)

```
mkdir("r1");
```

### Pour créer une arborescence de répertoires

```
mkdir("r1/r2/r3", recursive: true);
```

### Pour créer un répertoire (dossier)

```
mkdir("r1");
```

### Pour créer une arborescence de répertoires

```
mkdir("r1/r2/r3", recursive: true);
```

### Pour récupérer le chemin absolu d'un fichier ou d'un dossier

```
echo realpath("file.txt");  
// affiche "C:\wamp64\...\file.txt"
```

Pour lister les fichiers `.txt` d'un dossier : (`glob` retourne le chemin absolu de chaque fichier)

```
foreach (glob("r1\*.txt") as $filename) {  
    echo "$filename <br>";  
}
```

© Achref EL MOU



Pour lister les fichiers `.txt` d'un dossier : (`glob` retourne le chemin absolu de chaque fichier)

```
foreach (glob("r1\*.txt") as $filename) {  
    echo "$filename <br>";  
}
```

On peut aussi utiliser `scandir` (qui retourne le nom brut du fichier)

```
foreach (scandir("r1") as $filename) {  
    echo "$filename <br>";  
}
```

## glob VS scandir

- `scandir()` liste tout le contenu d'un dossier sans filtrage, tandis que `glob()` permet de filtrer les résultats selon un motif spécifique (en utilisant des caractères génériques comme `*` ou `?`).
- `scandir()` est généralement plus rapide que `glob()`.

## Pour tester s'il s'agit d'un dossier ou un fichier

```
var_dump(is_dir("r1"));  
// affiche true  
  
var_dump(is_dir("file.txt"));  
// affiche false  
  
var_dump(is_file("r1"));  
// affiche false  
  
var_dump(is_file("file.txt"));  
// affiche true
```

## Pour supprimer un fichier

```
unlink("file.txt");
```

© Achref EL MOUËL

## Pour supprimer un fichier

```
unlink("file.txt");
```

**Pour supprimer un dossier vide : si le dossier est non vide, il faut commencer par supprimer son contenu**

```
rmdir("r1");
```

## Autres opérations sur les dossiers

- `opendir()` : ouvre un dossier et récupère un pointeur dessus.
- `closedir()` : ferme le pointeur sur le dossier ouvert par `opendir()`.
- `readdir()` : lit une entrée du dossier.
- `getcwd()` : retourne le dossier de travail courant en cas de réussite ou `false` en cas d'échec.
- ...

## Exercice

Écrire un script **PHP** qui permet de

- 1 créer deux répertoires `d1` et `d2`
- 2 créer trois fichiers `f1.txt`, `f2.txt` et `f3.txt` dans `d1`
- 3 déplacer tous les fichiers de `d1` dans `d2`
- 4 ajouter le préfixe `test_` au nom de chaque fichier de `d2`
- 5 lister le nouveau contenu de `d2`
- 6 supprimer `d1`, `d2` et son contenu