

janvier 2015

PHP

---

### Lecture et écriture de fichiers en PHP

#### Ouverture d'un fichier

Avant de réaliser des opérations de lecture ou d'écriture sur un fichier, il est nécessaire de l'ouvrir. On dispose pour cela de la fonction `fopen` dont la syntaxe (simplifiée) est :

```
resource fopen(string $nom, string mode)
```

Le premier paramètre est le nom du fichier que l'on souhaite utiliser (éventuellement avec son chemin). Le second paramètre détermine le mode d'accès au fichier :

- mode "r" : le fichier est ouvert en lecture seule. La lecture commence au début du fichier.
- mode "r+" : le fichier est ouvert en lecture et en écriture. Ces opérations commencent au début du fichier.
- mode "w" : le fichier est ouvert en écriture seule. L'écriture commence au début du fichier.
- mode "w+" : le fichier est ouvert en lecture et en écriture. Ces opérations commencent au début du fichier.
- Pour les modes "w" et "w+", si le fichier n'existe pas, il est créé. Si le fichier existe, son contenu est effacé.
- mode "a" : le fichier est ouvert en écriture seule, et les données sont écrites en fin de fichier, à la suite de celles qui existent déjà. Si le fichier n'existe pas, il est créé.
- mode "a+" : le fichier est ouvert en lecture et écriture. Les données sont écrites en fin de fichier, à la suite de celles qui existent déjà. Les lectures se font à partir du début du fichier. Si le fichier n'existe pas, il est créé.

Le résultat de la fonction `fopen` est un identifiant de fichier, qui doit être utilisé comme premier paramètre de la plupart des fonctions de manipulation des fichiers. Il faut donc impérativement récupérer cette valeur dans une variable.

#### Lecture à partir d'un fichier, ligne par ligne

```
string fgets(resource $id_file)
```

Cette fonction lit une ligne du fichier. Si la lecture s'est bien déroulée, le résultat de la fonction est une chaîne contenant la ligne lue. Si une erreur survient (par exemple la fin du fichier), `fgets` renvoie le booléen `FALSE`.

#### Écriture dans un fichier

```
integer fputs(resource $id_file, string chaine)
```

Écrit la chaîne passée en second paramètre dans le fichier d'identifiant `$id_file`. L'entier renvoyé par cette fonction est le nombre d'octets écrits, ou `FALSE` en cas d'erreur.

## Fermeture d'un fichier

Quand on a fini d'utiliser un fichier, il faut le fermer à l'aide de la fonction `fclose` :

```
boolean fclose($id_file)
```

Le paramètre est un identifiant de fichier et le résultat est `TRUE` si l'opération s'est bien déroulée, `FALSE` sinon.

*Remarque : Pour les opérations de lecture et d'écriture, il faut que le serveur ait les droits nécessaires à la réalisation de ces opérations au niveau du système de fichiers.*

## Exemple

Le petit programme ci-dessous lit un fichier de texte dans son ensemble et affiche le nombre de lignes lues et leur longueur moyenne

```
<?php
$file = fopen('mon_fichier.txt','r'); //ouverture du fichier
$ligne = fgets($file); //tentative de lecture
$cpt = 0; $sommeLongueurs = 0;
while ($ligne !== FALSE) { // une ligne a vraiment été lue
    $cpt++;
    $sommeLongueurs += strlen($ligne);
    $ligne = fgets($file); // tentative de lecture de la ligne suivante
}
if ($cpt==0)
    echo "<p>Fichier vide</p>";
else {
    echo "<p>$cpt lignes lues. Longueur moyenne : " . $sommeLongueurs/$cpt . "</p>";
}
?>
```

### Exercice 1 : (Exercice d'application immédiate)

Créez tout d'abord sur le serveur webtp un dossier nommé `seance2`. Par la suite vous réaliserez tous les exercices de la séance d'aujourd'hui dans ce dossier.

Transférez dans ce dossier le fichier `liste_noms.txt` qui vous est fourni. Ce fichiers contient des noms, à raison d'un par ligne.

Concevez une programme PHP qui lit la totalité de ce fichier et construit une page web contenant la liste de tous les noms figurant dans le fichier (chaque nom sera un item de la liste).

**Exercice 2 :** Un terrain de jeu de forme carrée est composé de cases qui peuvent recevoir chacune un pion noir ou un pion blanc (on peut penser au jeu de Dames, par exemple, mais nos terrains auront des côtés de longueur quelconque).

Un terrain sera présenté en HTML par une `table` avec la convention suivante :

- Une cellule qui contient un pion blanc contiendra la lettre B. La cellule possèdera la classe «`blanc`»
- Une cellule qui contient un pion noir contiendra la lettre N. La cellule possèdera la classe «`noir`»
- Une cellule vide ne contient rien et n'appartient ni à la classe «`blanc`» ni à la classe «`noir`»

On suppose disposer d'un fichier contenant la description d'un terrain sous forme de texte. Une ligne de texte est une ligne du terrain. Chaque case est notée «`-`», «`B`» ou «`N`» selon qu'elle est vide ou contient un pion. Tout caractère autre que ces 3 là serait considéré comme indiquant la fin de la ligne. Par exemple

```
-B--N
B-NN-
-----
NN-BB
NBNBN
```

**Question 2.1 :** Écrire un programme PHP qui produit une page HTML affichant le terrain défini par le fichier :

1. Dans un premier temps, supposez les données correctes, sans les vérifier.
2. Puis vous testerez les cas d'erreurs : lignes de longueurs différentes, terrain non carré.  
En cas d'erreur des données vous ferez en sorte de ne pas afficher la table mais un message d'erreur. Pour ce faire, vous ne pourrez pas "afficher" les lignes au fur et à mesure de la lecture des données. Il vous faudra construire le code de la table HTML dans une chaîne et n'afficher cette chaîne qu'à la fin de la lecture, s'il n'y a pas eu d'erreurs.

**Question 2.2 :** CSS

Associer un style pour faire en sorte que les cases du terrain soient vraiment carrées et qu'elles aient une bordure. Les lettres ne seront pas affichées, mais les cases apparaîtront blanches, noires ou grisées (pour les cases vides)

**Question 2.3 :** JS (à faire en dehors de la séance)

Associer un fichier javascript pour rendre le terrain «interactif».

1. Un clic sur une case vide y mettra un pion blanc
2. Un clic sur une case blanche y mettra un pion noir
3. Un clic sur une case noire la videra

**Exercice 3 :** Consultez la documentation PHP pour comprendre les fonctions

- `trim()` qui permet de "supprimer" les espaces en début et fin de chaîne.
- `str_replace()` qui permet de remplacer une sous-chaîne par une autre
- `explode()` qui permet de découper une chaîne de caractères selon un séparateur.
- `implode()` la réciproque de la précédente

**Question 3.1 :**

Exercice d'application directe : définissez une chaîne de caractères contenant des signes plus qui y joueront le rôle de séparateur. Par exemple "Et qu'on sorte+ Vivement : +Car Clément + Le vous mande."  
Puis affichez dans des paragraphes séparés chacune des parties situées entre les signes plus, sans les espaces éventuellement situés au début et à la fin de chaque partie : `<p>Et qu'on sorte</p><p>Vivement :</p><p>`

**Question 3.2 :**

Une chaîne \$s contient une suite quelconque de noms séparés par la chaîne ' - '

Construire à partir de \$s la chaîne comportant la suite de noms situés dans des balises `span`.

par exemple si \$s vaut Dupont - Durand, le résultat vaudra `<span>Dupont</span><span>Durand</span>`

**Question 3.3 :** Construire une fonction PHP `enSpan($s)` qui implémente l'action précédente.