

Plan

- Chapitre 1 : Introduction
- Chapitre 2 : Notions de base
- Chapitre 3 : La gestion des fichiers avec PHP
- Chapitre 4 : La gestion des formulaires
- Chapitre 5: Les cookies et les sessions
- Chapitre 6 : POO avec PHP
- Chapitre 7 : Interfaçage avec une base de données
- Chapitre 8 : Frameworks

La gestion des fichiers avec PHP

Introduction

- PHP prend en charge l'accès au système de fichiers du système d'exploitation du serveur.
- Les opérations sur les fichiers concernent la création, l'ouverture, la suppression, la copie, la lecture et l'écriture de fichiers



Droits d'accès accordés au propriétaire, à son groupe et aux autres utilisateurs

La gestion des fichiers avec PHP

Ouverture

- La fonction `fopen()` permet d'ouvrir un fichier, que ce soit pour le lire, le créer ou y écrire.
`entier fopen(chaine nom du fichier, chaine mode);`
- mode : Type d'opération :
 - `r` (comme read) indique une ouverture en lecture seulement
 - `w` (comme write) indique une ouverture en écriture seulement (la fonction crée le fichier s'il n'existe pas)
 - `a` (comme append) indique une ouverture en écriture seulement avec ajout du contenu à la fin du fichier (la fonction crée le fichier s'il n'existe pas)
- Lorsque le mode est suivie du caractère `+` celui-ci peut être lu et écrit

La gestion des fichiers avec PHP

Ouverture

Exemple:

```
$fp = fopen("fichier.txt","r"); //lecture du fichier
```

```
$fp = fopen("fichier.txt","w"); //écriture dans le fichier depuis le début du fichier
```



Possibilité d'ouvrir des fichiers présents sur le web grâce à leur URL

Exemple:

```
$fp = fopen("http://www.mondomaine.fr","r"); //lecture du fichier
```

La gestion des fichiers avec PHP

Ouverture

Exemple:

```
<?
$fp = fopen("fichier.txt","r");
if (!$fp) {
    echo "Echec de l'ouverture du fichier";
    exit;
} else { // votre code;
}
?>
```

Quel est le resultat ?



La gestion des fichiers avec PHP

Lecture, écriture et parcours

- Ecriture: `fputs()` : écrire une chaîne de caractères dans le fichier. Elle renvoie 0 en cas d'échec, 1 dans le cas contraire

booléen `fputs(entier Etat_du_fichier, chaine Sortie);`

- Lecture: `fgets()` : récupérer une ligne du fichier. Elle renvoie 0 en cas d'échec, 1 dans le cas contraire

`fgets(entier Etat_du_fichier, entier Longueur);`

- Parcours: `eof()` : test si la fin du fichier est atteinte.

La gestion des fichiers avec PHP

Lecture, écriture et parcours

Exemple:

```
<?
if (!$fp = fopen("fichier.txt","r"))
    {echo "Echec de l'ouverture du fichier";}
else { $Fichier="";
    while(!feof($fp)) {
        $Ligne = fgets($fp,255);
        echo $Ligne;
        $Fichier .= $Ligne. "<BR>";
    }
    fclose($fp);
}
?>
```

On affiche la ligne

On stocke l'ensemble des lignes dans une variable

On ferme le fichier

La gestion des fichiers avec PHP

Lecture, écriture et parcours

Exemple:

```
<?
$nom="Jean"; $email="jean@dupont.fr";
$fp = fopen("fichier.txt","a");
fputs($fp, "\n");
fputs($fp, $nom." | ".$email);
fclose($fp);
?>
```

ouverture du fichier en écriture

Retour à la ligne

Ecriture du nom et email dans le fichier

La gestion des fichiers avec PHP

👉 Lecture, écriture et parcours



Il y a une autre manière de lire un fichier

- `file()`: retourner dans un tableau l'intégralité d'un fichier en mettant chacune de ces lignes dans un élément du tableau

Tableau `file(chaine nomdufichier)`;

La gestion des fichiers avec PHP

👉 Lecture, écriture et parcours

Exemple:

```
<?
$Fichier = "fichier.txt";
if (is_file($Fichier)) {
    if ($TabFich = file($Fichier)) {
        for($i = 0; $i < count($TabFich); $i++)
            echo $TabFich[$i];
        else {echo "Le fichier ne peut être lu...<br>";}
    }
    else {
        echo "Désolé le fichier n'est pas valide<br> " ;
    }
}
?>
```

Lecture du fichier en entier

Parcours du tableau

La gestion des fichiers avec PHP

👉 Autres fonctions

- `is_dir()` permet de savoir si le fichier dont le nom est passé en paramètre correspond à un répertoire
- `is_executable()` permet de savoir si le fichier dont le nom est passé en paramètre est exécutable.
- `is_link()` permet de savoir si le fichier dont le nom est passé en paramètre correspond à un lien symbolique.
- `is_file()` permet de savoir si le fichier dont le nom est passé en paramètre ne correspond ni à un répertoire, ni à un lien symbolique.

La gestion des fichiers avec PHP

👉 Fermeture



Il faut fermer les fichier après le traitement avec `fclose()`

Plan

- Chapitre 1 : Introduction
- Chapitre 2 : Notions de base
- Chapitre 3 : La gestion des fichiers avec PHP
- **Chapitre 4 : La gestion des formulaires**
- Chapitre 5: Les cookies et les sessions
- Chapitre 6 : POO avec PHP
- Chapitre 7 : Interfaçage avec une base de données
- Chapitre 8 : Frameworks