



Partie 4 : Quelques fonctions de PHP

Partie 4 : Quelques fonctions de PHP



1. Écrire un fichier
2. Lire un fichier avec fopen
3. Lire un fichier avec file_get_contents
4. Uploader un fichier
5. Travailler sur les dates

Partie 4 : Ecrire un fichier

Ouverture du fichier :

```
$monFichier = fopen('nomFichier.txt', 'a+')
```

Lit la première ligne du fichier :

```
$txtMaLigne = fgets($monFichier);
```

Ecrire une ligne dans le fichier :

```
fputs($monFichier, $txt.PHP_EOL);
```

Fermer le fichier :

```
fclose($monFichier);
```

Mode	Explication
r	Ouvre le fichier en lecture seule. Cela signifie que vous pourrez seulement lire le fichier.
r+	Ouvre le fichier en lecture et écriture. Vous pourrez non seulement lire le fichier, mais aussi y écrire (on l'utilisera assez souvent en pratique).
a	Ouvre le fichier en écriture seule. Mais il y a un avantage : si le fichier n'existe pas, il est automatiquement créé.
a+	Ouvre le fichier en lecture et écriture. Si le fichier & n'existe pas, il est créé automatiquement. Attention : le répertoire doit avoir un CHMOD à 777 dans ce cas ! À noter que si le fichier existe déjà, le texte sera rajouté à la fin.

Partie 4 : Lire un fichier avec FOPEN

La Commande fopen/ fgets ne permet de lire le fichier que ligne par ligne il faut donc créer une boucle pour lire le fichier.

```
<?php

$handle = @fopen("pizza.txt", "r");

if ($handle) {
    while (($buffer = fgets($handle, 4096)) !== false) {
        echo $buffer;
    }
    if (!feof($handle)) {
        echo "Erreur: fgets() a échoué\n";
    }
    fclose($handle);
}

?>
```

Partie 4 : Lire un fichier avec `file_get_contents`

`file_get_contents()` est la façon recommandée pour lire le contenu d'un fichier dans une chaîne de caractères.

Lire le contenu d'un fichier texte :

```
$contenuDeMonFichier = file_get_contents('pizza.txt');  
  
echo $contenuDeMonFichier;
```

Lire le contenu d'une page web :

```
$contenuDeMaPageWeb = file_get_contents('http://www.google.fr');  
  
echo $contenuDeMaPageWeb;
```

Partie 4 : Upload : formulaire d'upload

```
<form action="" method="post" enctype="multipart/form-data">  
  <input type="file" name="fichier" /><br/>  
  <input type="submit" value="Envoyer" />  
</form>
```

La balise <form> doit contenir un nouvel attribut `enctype="multipart/form-data"` autorisant l'envoi de fichiers

Le formulaire doit contenir 1 ou plusieurs champs de `type file`

Partie 4 : Upload traitement des données

Le ou les images sont téléchargés sur dans un répertoire temporaire sur le serveur, les informations sur les fichiers téléchargés sont envoyées dans la variable superglobale `$_FILES`, qui est un array.

Pour lire les informations d'un fichier uploadé par un champ nommé `fichier` :

```
$_FILES['fichier']
```

La commande `move_uploaded_file($source, $destination)`, Déplace le fichier téléchargé.

name	Nom du fichier uploadé
type	type mime
tmp_name	nom du fichier temporaire
error	Code d'erreur quand il y en a 0 si tout est ok
size	taille en octet du fichier téléchargé

Partie 4 : Upload traitement des données

```
// Verifie si un fichier est posté
if(isset($_FILES['fichier'])) {

    // Copie les infos du fichier
    $fichier = $_FILES['fichier'];

    // Si il n'y a pas d'erreur d'upload
    if($fichier['error']==0){

        // Déplace le fichier téléchargé ou je veux
        move_uploaded_file($fichier['tmp_name'], $fichier['name']);

    }else{
        echo 'Le fichier n\'est pas correctement uploadé';
    }

    print_r($_FILES['fichier']);
}
```


Partie 4 : Manipuler les dates

La première façon de gérer les dates avec PHP consiste à utiliser le timestamp. Le timestamp mesure le temps écoulé secondes depuis le début de l'époque UNIX, (1er janvier 1970 00:00:00 GMT).

La commande `time()` retourne le timestamp actuel.

La commande `date('Y-m-d')` retourne la date actuelle suivant le format définis

```
// Affiche la date + 1 Jour
$jour = 3600*24;
$timemaintenant = time();
$time1jour = $timemaintenant + $jour;

echo date('Y-m-d', $time1jour) ;
```

Partie 4 : Manipuler les dates

La manipulation des date avec timestamp est complexe, c'est pourquoi depuis php 5 la classe date à été introduite. L'utilisation de cette classe permet une manipulation plus évidente des dates.

```
$maDate = new DateTime();
```

L'objet ma date est instancié avec la date du jour, avec le fuseau horaire du serveur.

```
echo $maDate->format('Y-m-d');
```

J'affiche ma date au format définis

```
$maDate->modify('+1 day');
```

J'ajoute 1 jour à ma date

Partie 4 : Manipuler les dates

```
// Calcul l'age
$dateNaissance = new DateTime('1979-08-10');
$dateDuJour    = new DateTime();

$age = $dateNaissance->diff($dateDuJour);
echo $age->format('%y ans');
```