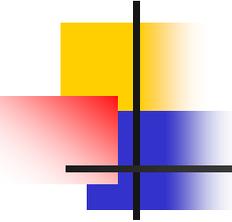


PHP 5

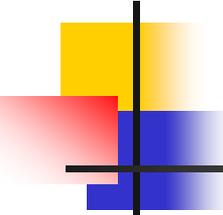
Les fichiers



Les fichiers

■ Généralités

- De nombreuses applications travaillent avec des fichiers
- Les fichiers permettent de sauvegarder des données sur disque et les rendent en quelque sorte pérennes
- On peut les utiliser dans des sessions ultérieures
- Il existe de nombreuses fonctions pour les lire, les remplir, les supprimer et même changer leurs attributs
- Dans ce chapitre on va s'intéresser aux opérations de lecture/écriture puis nous regarderons quelques opérations de leur manipulation sur disque



Les fichiers texte

■ Fichier texte

- PHP ne lit et n'écrit dans les fichiers que sous forme de texte
- Un fichier texte est un fichier caractère, construit sous un éditeur de texte
- Il est organisé sous la forme d'une suite de lignes de caractères
- Le fichier est suffixé par .txt

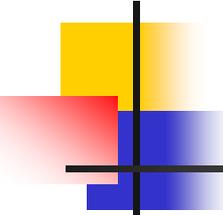
■ Comment travailler sur un fichier texte ?

- Pour travailler sur un fichier texte, il faut l'ouvrir
- On utilise pour cela fonction **fopen**
- Exemple : ici on ouvre le fichier en lecture : " r"

```
<?php
    $f=fopen("exemple.txt","r");
?>
```

■ Les différents modes d'ouverture

- **'r'** ouvre en lecture seule : le pointeur est placé au début du fichier
- **'r+'** ouvre en lecture/écriture : le pointeur est placé au début du fichier
- **'w'** ouvre en écriture seule : le pointeur est placé au début du fichier. Réduit la taille du fichier à 0. Tentative de création si celui ci n'existe pas
- **'w+'** ouvre en lecture/écriture : Le pointeur est placé au début du fichier. Réduit la taille du fichier à 0. Tentative de création si celui ci n'existe pas
- **'a'** ouvre en écriture seule : Le pointeur est placé au début du fichier. Tentative de création si celui ce n'existe pas.
- **'a+'** ouvre en lecture/écriture : Le pointeur est placé à la fin du fichier. Tentative de création si celui ce n'existe pas.
- **'x'** orée et ouvre le fichier en lecture seule : Le pointeur est placé au début du fichier. Si le fichier existe déjà, fopen va échouer
- **'x+'** orée et ouvre le fichier en lecture/écriture : le pointeur est placé au début du fichier. Si le fichier existe déjà, fopen va échouer



Les fichiers texte

■ Note

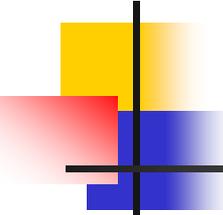
- Si la fonction fopen() échoue lors de l'ouverture du fichier, celle ci retourne 0
- Exemple :

```
<?php
    if (!($f=fopen("exemple.txt","r")))
        exit("Unable to open file!");
?>
```

■ Fermer un fichier

- La fonction fclose() est utilisée pour fermer un fichier

```
<?php
    fclose($f);
?>
```



Les fichiers texte

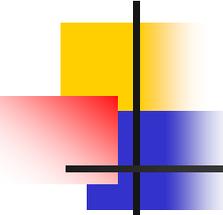
- **Écrire dans un fichier**

- L'écriture dans un fichier texte se fait avec la fonction `fwrite()`. Pour faire des retours à la ligne vous devez utiliser : `"\n"`

```
<?php
    $f = 'exemple6.txt';
    $text = "truc";
    $var_entier=34;
    $var_decimal=2500.567;
    $handle = fopen($f,"w");
    fwrite($handle, $text);
    fwrite($handle, "\n" );
    fwrite($handle, $var_entier);
    fwrite($handle, "\n" );
    fwrite($handle, $var_decimal);
    fclose($handle);
```

```
?>
```

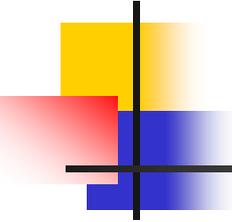
- Les entiers et les décimaux sont écrits sous forme de chaînes de caractères et non sous forme binaire



Les fichiers texte

- Écrire dans un fichier, de manière contrôlée

```
<?php
    $f = 'exemple.txt';
    $text = "ma chaine de caractères";
    $handle = fopen($f,"w");
    // Regarde si le fichier est accessible en écriture
    if (is_writable($f)) {
    // Écrit $text et vérifie
        if (fwrite($handle, $text) === FALSE) {
            echo 'Impossible d'écrire dans le fichier '.$f.';
            exit;
        }
        echo 'Ecriture terminée';
        fclose($handle);
    }
    else {
        echo 'Impossible d'écrire dans le fichier '.$f.';
    }
?>
```

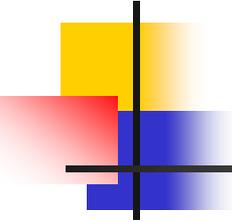


Les fichiers texte

- Trouver la fin d'un fichier

- La fonction `feof()` est utilisée pour tester la fin du fichier

```
<?php
    if (feof($f))
        echo 'Fin du fichier';
?>
```

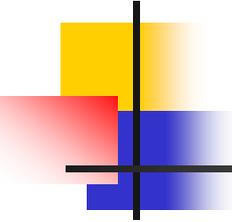


Les fichiers texte

■ Comment lire caractère par caractère ?

- Par `fgetc()`
- Exemple :

```
<?php
if (!($f=fopen("exemple.txt","r")))
exit("Impossible d'ouvrir le fichier.");
while (!feof($f))
{
    $texte=fgetc($f);
    echo $texte;
}
fclose($f);
?>
```

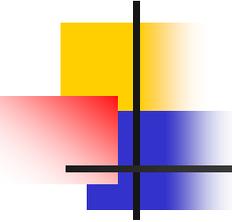


Les fichiers texte

■ Comment lire un fichier ligne par ligne ?

- Par `fgets()` :
- Exemple

```
<?php
    $ressource_fichier = fopen('test.txt', 'r');
    if($ressource_fichier)
    {
        $contenu_fichier = "";
        while(!feof($ressource_fichier)) //tant que l'on est pas à la
        fin du fichier
        {
            $contenu_fichier .= fgets($ressource_fichier);//récupère
            la ligne en cours et l'ajoute au contenu de la variable
            $contenu_fichier
        }
        fclose($ressource_fichier);
        echo $contenu_fichier; //affiche le contenu du fichier
    }
?>
```



Les fichiers texte

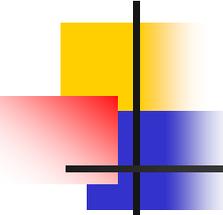
■ Lire un fichier sous forme de tableau

- Chaque ligne du tableau correspondra à une ligne du fichier
- Ceci est très pratique quand vous avez besoin d'une ligne en particulier dans votre fichier (dont vous connaissez le numéro) ou encore pour faire des opérations de tri alphabétique, numérique...
- La fonction que nous allons utiliser s'appelle `file()`
- Comme son nom ne l'indique pas, elle va vous retourner un tableau contenant tout votre fichier
- Voici comment on l'utilise :

```
<?php
```

```
    $tableau = file('test.txt'); //Place le contenu du fichier dans  
    un tableau, on a supposé ici que le fichier existe sous peine  
    d'avoir une erreur
```

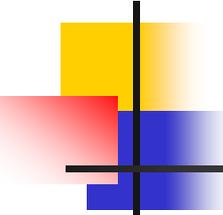
```
?>
```



Les fichiers texte

- L'exemple complet :

```
<?php
    $tableau = file('test.txt');
    if(is_array($tableau)) //Si la variable $tableau est bien un
        tableau, on peut continuer
    {
        $contenu_fichier = "";
        foreach($tableau AS $ligne)
        {
            $contenu_fichier .= $ligne;
        }
        echo $contenu_fichier; //Affiche le contenu du fichier
        , notez que l'on a pas besoin d'utiliser fclose() ici
    }
?>
```



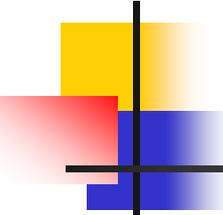
Les fichiers texte

■ Positionnement dans le fichier

- Placer le pointeur au début du fichier : `rewind()`
 - Permet de revenir en début de fichier, à l'octet 0

- Exemple

```
<?php
    $fp=fopen("monfichier.txt", "rb");
    //On affiche le fichier ligne par ligne
    //rb : pour lire ('r ') un fichier enregistré sous forme binaire ('b')
    //On prévient ainsi PHP qui ne lit que du texte
    while($ligne=fgets($fp,1024){
        echo $ligne;
    }
    //On se place au début
    rewind($fp);
    //On réaffiche la première ligne
    $ligne1 = fgets($fp,1024);
    echo $ligne1;
?>
```

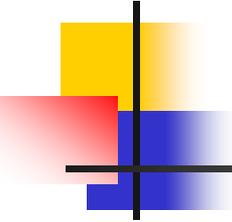


Les fichiers texte

■ Positionnement dans le fichier

- Placer le pointeur en un point du fichier : fseek()
- Exemple

```
<?php
    $fp=fopen("monfichier.txt", "rb");
    //On se positionne au 42ème caractère
    fseek($fp,42);
    echo fgetc($fp);
    //On se positionne 10 caractères plus loin, donc au 52
    fseek($fp,10, SEEK_CUR);
    echo fgetc($fp);
    //Puis 20 caractères avant la fin du fichier
    fseek($fp,-20, SEEK_END);
    echo fgetc($fp);
    //On revient au 42e caractère à partir du début
    fseek($fp,42, SEEK_SET);
    echo fgetc($fp);
?>
```



Les fichiers texte

■ Positionnement dans le fichier

- fseek() est utilisée de concert avec la fonction ftell() qui renseigne sur la position du pointeur
- Exemple

<?

```
$fp=fopen("monfichier.txt", "rb");
```

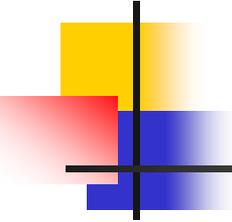
```
//On se positionne au 42ème octet
```

```
fseek($fp,42);
```

```
//On récupère l'emplacement du pointeur
```

```
$position=ftell($fp);
```

?>



Les fichiers texte

■ Création et effacement

- On peut utiliser la fonction touch() pour modifier la date de modification d'un fichier
- Exemple

<?

```
//ici on décide de la date de modification du fichier
```

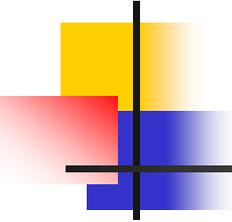
```
$time=mktime(0,0,0,10,10,2010);
```

```
$fichier='fantome.txt';
```

```
touch($fichier,$time);
```

```
}
```

```
?>
```

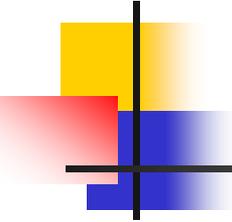


Les fichiers texte

■ Effacement de fichier

- Unlink()
- Exemple

```
<?
    $fichier='fantome.txt';
    if (file_exists($fichier)){
        unlink($fichier);
    }
?>
```

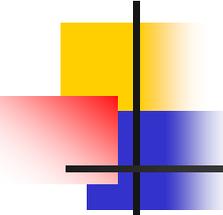


Les fichiers texte

■ Parcourir un répertoire

- Il s'agit de parcourir un répertoire et d'afficher la liste des noms de fichiers contenus : lecture_repertoire.php

```
<?php
    $dir=dir('.'); //le . Indique le répertoire courant
    while($nom=$dir->read()){
        echo $nom,'<br>';
    }
    $dir->close();
?>
```

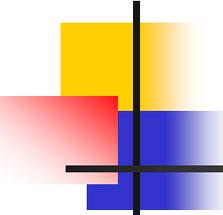


Les fichiers texte

- **Parcourir un répertoire**

- Cela donne pour le répertoire courant : L3-ISC-Exemples-Fichiers

```
.  
..  
ch11exo1.php  
ch11exo2.php  
ch11exo3.php  
ch11exo3b.php  
ch11exo3c.php  
ch11exo3d.php  
compteur.txt  
connex.txt  
fichiers1.php  
fichiers2.php  
fichiers3.php  
fichiers4.php  
fichiers5.php  
fichiers6.php  
fichiers7.php  
html.txt  
javascript.txt  
lecture_rapide.php  
lecture_repertoire.php  
lecture_repertoire2.php  
livre.txt  
php.txt  
votes.txt
```



Les fichiers texte

■ Parcourir un répertoire

- Utilisation de `opendir()`, `readdir()` et `closedir()` avec le même résultat que précédemment : `lecture_repertoire2.php`

```
<?
```

```
//script récupérant tous les fichiers d'un répertoire
```

```
//dans un tableau $fichier
```

```
$dir=opendir(".");
```

```
$fichiers=array();
```

```
while($nom=readdir($dir)){
```

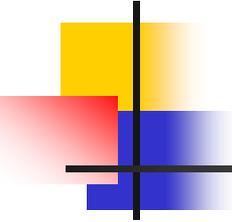
```
    $fichiers[]=$nom;
```

```
    echo $nom.'<br>';
```

```
}
```

```
closedir($dir);
```

```
?>
```



Les fichiers texte

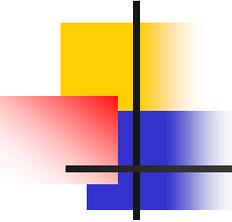
■ Parcourir un répertoire

- Utilisation de masque par `glob()` : `lecture_repertoire3.php`, avec le même résultat que précédemment

```
<?php
    $fichiers=glob('./*');
    print_r($fichiers);
?>
```

- Ici, on va sélectionner tous les fichiers `.jpg`

```
<?php
    $fichiers=glob('./*.jpg');
    print_r($fichiers);
?>
```



Les fichiers texte

■ Position dans l'arborescence

- Getcwd() permet de connaître le nom du répertoire courant

```
<?php
```

```
    echo 'repertoire courant : ' . getcwd();
```

```
?>
```

- chdir() pour remonter dans l'arborescence

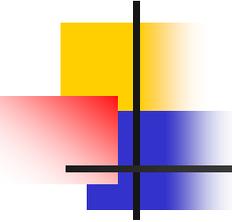
```
<?php
```

```
    echo 'repertoire courant : ' . getcwd(). '<br>';
```

```
    chdir('..');
```

```
    echo 'repertoire courant : ' . getcwd(). '<br>';
```

```
?>
```



Les fichiers texte

■ Date de modification des fichiers

- filectime() : donne la date de création
- filemtime() : donne la date de dernière modification
- fileatime() : donne la date de dernier accès

```
<?php
```

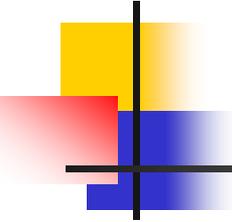
```
    if(filemtime('monfichier.txt')>filectime('monfichier.txt')){  
        echo 'le fichier a été modifié depuis sa création';
```

```
    }
```

```
    else
```

```
        echo 'le fichier n\'a été modifié';
```

```
?>
```



Les fichiers texte

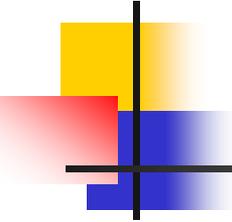
- Taille de fichier : filesize()

```
<?php
```

```
    $fichier='monfichier.txt';
```

```
    echo $fichier . ' fait ' . filesize($fichier). ' octets ';
```

```
?>
```



Les fichiers texte

■ Sécurité

- Découvrez par vous-mêmes plusieurs fonctions permettant de connaître le propriétaire, la permission d'accès, de changement de ces permissions...