

1- La gestion des fichiers: (2/13)

b- Ouvrir un fichier:

- ❑ Pour toute action dans un fichier, il faut au préalable l'ouvrir avec la fonction ***fopen***:

```
$fichier=fopen ("nomFichier" , "mode") ;
```

fopen: renvoie la valeur **false** si l'ouverture n'a pu avoir lieu, sinon elle renvoie le pointeur (ici : \$fichier) référençant le fichier ouvert.

nomFichier: chaîne qui représente le nom du fichier s'il existe dans le même répertoire que le script PHP ou bien avec son chemin complet.

mode: c'est le mode d'ouverture du fichier.

1- La gestion des fichiers: (3/13)

Chapitre III

Mode d'ouverture	Signification
r	Ouverture en lecture seule (<i>read</i>)
w	Ouverture en écriture seule (<i>write</i>) ; le fichier est créé s'il n'existe pas
r+	Ouverture en lecture/écriture
w+	Ouverture en lecture/écriture ; le fichier est créé s'il n'existe pas et, s'il existe, il est préalablement mis à zéro
a	Ouverture en écriture (<i>append</i>) ; le fichier est créé s'il n'existe pas et, s'il existe, le pointeur est placé à la fin du fichier
a+	Ouverture en lecture/écriture ; le fichier est créé s'il n'existe pas et, s'il existe, le pointeur est placé à la fin du fichier

1- La gestion des fichiers: (4/13)

□ Exemple:

```
<?php
$ichier = fopen("exemple.txt", "r");
if($ichier!=false)
{
    echo "Fichier ouvert correctement";
}
else
{
    echo "Problème d'ouverture du fichier";
}
?>
```

1- La gestion des fichiers: (5/13)

c- Fermer un fichier:

`fclose ($fichier) ;`

Permet de libérer le pointeur afin de libérer le fichier.

1- La gestion des fichiers: (6/13)

d- Lecture des données:

❑ Par caractère: *fgetc*

```
$fichier = fopen("fichier.txt", "r");  
while(!feof($fichier))  
{  
    $car=fgetc($fichier);  
    echo $car;  
}  
fclose ($fichier);
```

feof: permet de tester la fin du fichier que vous lisez. Si elle est atteinte, la fonction renvoie la valeur **true**.

1- La gestion des fichiers: (7/13)

□ Par ligne: *fgets*

```
$ligne = fgets ($fichier, $taille) ;
```

Lecture de `$taille-1` octets, ou bien le retour chariot (\n) est rencontré.

```
while (!feof ($fichier))  
{  
    $ligne=fgets ($fichier,255) ;  
    echo $ligne;  
}  
//lecture de 255 caractères à chaque itération  
ou bien fin de la ligne dans le fichier
```

1- La gestion des fichiers: (8/13)

❑ Lecture du fichier: *fread*

```
$contenu= fread($fichier,$taille) ;
```

fread fonctionne comme **fgets** à la différence près qu'elle lira toujours des blocs de x octets quelque soit la place du retour chariot.

Il est très pratique d'utiliser la fonction **filesize()** qui retourne la taille exacte du fichier passé en argument.

1- La gestion des fichiers: (9/13)

Lecture du fichier dans son intégralité:

```
$nomF = "fichier.txt";  
$fichier = fopen($nomF, "r");  
$cont = fread($fichier, filesize($nomF));  
echo $cont;  
fclose ($fichier);
```


1- La gestion des fichiers: (10/13)

❑ Lecture avec un tableau: *file*

Lit le fichier et renvoie le résultat dans un tableau dont chaque case correspond à une ligne du fichier.

```
$lecture = file("fichier.txt");  
foreach($lecture as $ligne)  
{  
    echo "$ligne <br/>";  
}
```

1- La gestion des fichiers: (11/13)

e- Ecriture des données:

- ❑ ***fwrite***: permet d'écrire des chaînes de caractères dans le fichier et retourne le nombre de caractères écrits.

```
$fichier = fopen("fichier.txt", "a");  
$ok = fwrite($fichier, "Ligne à ajouter");  
echo "nombre de caractères ajoutés: $ok";  
fclose ($fichier);
```

- ❑ ***fputs***: fonction similaire à ***fwrite()***.

1- La gestion des fichiers: (12/13)

f- autres fonctions:

- ❑ **file_exists**: vérifie l'existence du fichier

```
if (file_exists ("nomFichier"))
```

- ❑ **copy**: copie d'un fichier

```
copy ("fichierSource", "fichierDestination")
```

- ❑ **unlink**: permet de détruire un fichier.

```
unlink ("nomFichier") ;
```

- ❑ **rename**: renommer un fichier.

```
rename ("oldname", "newname")
```

1- La gestion des fichiers: (13/13)

- ❑ **rewind**: déplace le pointeur de fichier au début

```
$fichier = fopen($nomF, "r");  
rewind($fichier);
```

- ❑ **filetime**: déterminer depuis quand le fichier a été accédé, la valeur renvoyée correspond au nombre de secondes écoulées depuis le 1er janvier 1970

```
$duree=filetime($nomF);  
echo date("d/m/Y", $duree);
```

- ❑ **filemtime**: depuis quand le fichier a été modifié

```
$duree=filemtime($nomF);
```

- ❑