

# Les fichiers

# Table des matières



<b>Objectifs</b>	3
<b>Introduction</b>	4
<b>I - Créer et ouvrir un fichier</b>	5
1. La démarche pour utiliser un fichier .....	5
2. Création et ouverture d'un fichier .....	5
<b>II - Exercice : Activité d'auto-évaluation No 1</b>	8
<b>III - Lecture et écriture dans un fichier</b>	9
1. Écriture dans un fichier .....	9
2. Lecture dans un fichier .....	10
3. Lecture et écriture rapide dans un fichier .....	12
<b>IV - Exercice : Activité d'auto-évaluation No 2</b>	14
<b>V - Les fonctions de manipulation des fichiers</b>	16
1. Quelques fonctions très pratiques de manipulation de fichier .....	16
2. Exemple de manipulation de fichier .....	18
<b>VI - Exercice : Activité d'auto-évaluation No 3</b>	23
<b>VII - Lire et écrire dans un fichier structuré</b>	24
1. Enregistrement d'informations de façon structurée dans un fichier texte simple .....	24
2. Enregistrement d'informations de façon structurée dans un fichier texte csv .....	28
3. Travaux dirigés .....	30
<b>VIII - Exercice : Activité d'auto-évaluation No 4</b>	37

# Objectifs

À la fin de cette leçon, vous serez capable de :

- Créer et ouvrir un fichier
- Lire et écrire dans un fichier non structuré
- Identifier quelques fonctions de manipulation des fichiers
- Lire et écrire dans un fichier structuré

# Introduction



Jusqu'à présent, nous savons stocker des informations de manière temporaire grâce aux variables. La durée de vie d'une variable n'est en effet jamais très longue. Or, vous aurez certainement besoin sur votre site de stocker des informations définitivement.

Pour stocker ces informations longtemps, il faut les écrire sur le disque dur. Pour cela, la méthode la plus simple est d'utiliser les fichiers. PHP permet justement d'enregistrer des données dans des fichiers sur le disque dur du serveur.

Les fichiers vont donc nous offrir une alternative aux bases de données que nous étudierons dans la suite de ce cours et qui servent également à stocker des informations de manière organisée et définitive.

PHP nous fournit différentes fonctions qui vont nous permettre de créer, lire, éditer, fermer ou encore écrire dans des fichiers. Nous allons étudier chacune d'entre elles dans cette leçon.

# Créer et ouvrir un fichier

I

## 1. La démarche pour utiliser un fichier

Avant de lire ou écrire dans un fichier, il faut :

- Créer le fichier s'il n'existe pas, ou l'ouvrir s'il existe grâce à la fonction *fopen()*
- Après la création ou l'ouverture du fichier
  - on lit (récupérer les informations) dans le fichier grâce à la fonction *fread()* ou *fgets()*
  - on écrit dans le fichier grâce à la fonction *fwrite()* ou *fputs()*
- Après avoir fini de lire ou écrire dans le fichier, on ferme le fichier grâce à la fonction *fclose()*

*NB* : Il existe une manière plus rapide de lire et écrire dans un fichier sans passer par cette démarche grâce à d'autres fonctions qu'on verra un peu plus bas.

## 2. Création et ouverture d'un fichier

### *Création d'un fichier*

Pour créer un fichier, on utilise la fonction *fopen()*, la syntaxe est : `$nom_variable_fichier = fopen ($nomFichier, $mode);`

- `$nom_variable_fichier` est une variable que nous créons qui représente le fichier, c'est par cette variable que nous manipulons le fichier
- `$nomFichier` est le nom que nous donnons à notre fichier sur le disque, ce nom s'écrit entre quote ou entre double quotes et comprend l'extension du fichier. Dans le cas des fichiers textes, on utilise l'extension *.txt*  
Sachez que `$nomFichier` est une URI (le chemin ou répertoire d'accès au fichier), il peut être défini de façon absolue, ou de façon relative en fonction de l'emplacement de votre page PHP. Si votre fichier est ou doit être dans un dossier particulier, il faut indiquer le chemin d'accès au fichier.
- `$mode` représente le mode d'ouverture du fichier, nous allons voir à la suite les différents modes d'ouverture. Dans le cas d'une création de fichier, cette valeur peut être *a*, *a+*, *w*, *w+*, *x* ou *x+*

*Exemple de création de fichier :*

```
<?php
$listeLivre= fopen("livre.txt", "w+") ;
?>
```

Si vous exécutez ce code, vous verrez un fichier se créer automatiquement là où votre page php se trouve.

Si votre fichier livre.txt doit se trouver dans un dossier docs/, au même emplacement que votre page php, il faut faire :

```
<?php
$listeLivre= fopen("docs/livre.txt","w+") ;
?>
```

Sachez que vous devez créer le dossier *docs* sur votre ordinateur d'abord, là où se trouve votre page php, sinon vous aurez une erreur. Nous verrons un peu plus bas comment créer un dossier directement en php.

*NB* : il est possible de créer un fichier directement sur l'ordinateur sans passer par PHP, puis nous accédons après à ce fichier en utilisant le PHP

### *Les modes d'ouverture d'un fichier*

Pour ouvrir un fichier on utilise la fonction *fopen()* de la même manière que nous avons vu pour la création d'un fichier. Toute fois, en fonction de la manière dont nous aimerons ouvrir le fichier, il faut designer le mode d'ouverture approprié. Dans cette partie, nous allons voir les différentes manières d'ouvrir un fichier.

<i>Mode d'ouverture</i>	<i>Description</i>
r	Ouvre un fichier en <i>lecture seule</i> . Il est impossible de modifier le fichier.
r+	Ouvre un fichier en <i>lecture et en écriture</i> . Si le fichier n'existe pas une erreur se produit. Si le fichier existe les nouvelles informations sont ajoutées au début du fichier.
a	Ouvre un fichier en <i>écriture seule</i> en conservant les données existantes. Les informations sont ajoutées à la fin des informations existantes dans le fichier. Si le fichier n'existe pas, il crée un fichier.
a+	Ouvre un fichier en <i>lecture et en écriture</i> en conservant les données existantes. Son fonctionnement est similaire au mode "a"
w	Ouvre un fichier en <i>écriture seule</i> . Si le fichier existe, les informations existantes seront supprimées. S'il n'existe pas, crée un fichier.
w+	Ouvre un fichier en <i>lecture et en écriture</i> . Si le fichier existe, les informations existantes seront supprimées. S'il n'existe pas, crée un fichier.
x	Crée un nouveau fichier accessible en <i>écriture seulement</i> . Retourne false et une erreur si le fichier existe déjà.
x+	Crée un nouveau fichier accessible en <i>lecture et en écriture</i> . Retourne false et une erreur si le fichier existe déjà.

Comme vous pouvez le constater, le choix du mode influe fortement sur les opérations que nous allons pouvoir réaliser sur le fichier en question.

Par exemple lorsqu'on ouvre un fichier en lecture seulement toute modification de ce fichier est impossible, ce qui n'est pas le cas si on choisit un mode permettant l'écriture dans ce fichier.

# Exercice : Activité d'auto-évaluation No 1



II

## Exercice

---

Pour fermer le flux sur un fichier ouvert, on utilise la fonction `fclose()`

## Exercice

---

Le mode `r` ouvre un fichier en lecture seule. Il est impossible de modifier le fichier.

## Exercice

---

Le mode `w` ouvre un fichier en écriture seule. Si le fichier existe, les informations existantes seront supprimées. S'il n'existe pas, crée un fichier.

## Exercice

---

Le mode `a` crée un nouveau fichier accessible en lecture et en écriture. Retourne false et une erreur si le fichier existe déjà.



# Lecture et écriture dans un fichier



## 1. Écriture dans un fichier

Pour écrire dans un fichier, il faut l'ouvrir en utilisant la fonction `fopen()` avec un mode autorisant l'écriture dans le fichier, puis, il faut utiliser la fonction `fwrite()` ou `fputs()` pour écrire les informations dans le fichier.

### Exemple : Écriture dans un fichier avec `fwrite()`

- Pour écrire dans le fichier on utilise la fonction `fwrite` qui prend en paramètre la variable fichier et le text à écrire dans le fichier
- le symbole `\r\n` permet de faire un retour à la ligne dans le fichier
- le mode `"a"` permet d'ouvrir un fichier en mode écriture uniquement, `"a+"` en lecture et écriture. Si le fichier n'existe pas il sera créé, s'il existe déjà, l'information est ajouté à la fin des informations existante dans le fichier. Nous pouvons choisir un autre mode d'écriture (`a+`, `x`, `x+`, `w`, `w+`, `r+`) mais le comportement ne sera pas le même.
- Après avoir fini d'écrire dans le fichier, on ferme le flux d'écriture grâce à la fonction `fclose` qui prend en paramètre la variable fichier créée.

```
1 <?php
2 $fichier = fopen("infos.txt", "a");
3 $texte="Mariam joue à la balle \r\n";
4 fwrite($fichier, $texte);
5 $texte="Irié va à la pêche";
6 fwrite($fichier, $texte);
7 fclose($fichier);
8 echo "Ecriture terminée <br/>";
9 ?>
```

- Si vous exécutez ce code, un fichier du nom de `infos.txt` sera créé là où se trouve votre page php, et vous verrez les deux phrases ci-dessus à l'intérieur. Si vous exécutez plusieurs fois le fichier, ces deux phrases font s'ajouter au fur et à mesure.
- Si vous choisissez le mode `w` ou `w+`, chaque fois que vous allez exécuter le fichier le fichier sera créé à nouveau et les anciennes informations vont disparaître
- Si vous choisissez le mode `"r+"`, si le fichier n'est pas encore créé sur le disque (n'existe pas encore), il y aura une erreur.

### Écriture dans un fichier avec `fputs()`

- On décide cette fois ci de créer le fichier dans un dossier docs. Il faut donc créer d'abord un dossier docs là où se trouve la page php

- Avant d'ouvrir le fichier, on teste d'abord si le répertoire *docs/* existe d'abord

```

1 <?php
2 $rep="docs/";
3 if(is_dir($rep)){
4     $fichier = fopen("$rep"."infos.txt", "a");
5     $texte="Mariam joue à la balle \r\n";
6     fputs($fichier,$texte);
7     $texte="Irié va à la pêche \r\n";
8     fputs($fichier,$texte);
9     fclose($fichier);
10    echo "Ecriture terminée <br/>";
11 }else echo "Le repertoire <strong> $rep </strong> n'existe pas";
12
13 ?>

```

## 2. Lecture dans un fichier

Pour lire les informations d'un fichier, il faut :

- d'abord l'ouvrir en utilisant la fonction *fopen()* avec un mode autorisant la lecture des informations dans le fichier (mode "r", "r+", "a+", "w+", ou "x+"),
- puis, il faut utiliser la fonction *fread()* ou *fgets()* pour lire les informations dans le fichier.
- les fonctions *fread* et *fgets* prennent en paramètre la variable fichier, puis le nombre de caractères à lire dans le fichier. Dans le cas de la fonction *fgets*, si le nombre de caractère n'est pas précisé, c'est une seule ligne qui est lue. Quand à la fonction *fread*, il faut obligatoirement préciser le nombre de caractère à lire

### Exemple : Exemple de lecture dans un fichier

Dans cet exemple, on lit seulement les 2 premières lignes du fichier

Si le code ne s'affiche pas correctement en format web, veuillez cliquer sur le lien en bleu

```

1 <?php
2 $fichier = fopen("infos.txt", "r+");
3 $texte= fgets($fichier);
4 echo "<i>1ère ligne:</i> $texte <br/>";
5 $texte= fgets($fichier);
6 echo "<i>2è ligne :</i> $texte <br/>";
7 fclose($fichier);
8 ?>

```

On peut aussi procéder comme suit (on indique le nombre de caractères à lire) :

```

1 <?php
2 $fichier = fopen("infos.txt", "r+");
3 $texte= fgets($fichier, 255);
4 echo "<i>1ère ligne:</i> $texte <br/>";
5 $texte= fgets($fichier, 255);
6 echo "<i>2è ligne :</i> $texte <br/>";
7 fclose($fichier);
8 ?>

```

Si on veut lire une partir de la première ligne, on peut faire :

```

1 <?php
2 $fichier = fopen("infos.txt", "r+");
3 $texte= fgets($fichier, 18);
4 echo "$texte <br/>";
5 fclose($fichier);
6 ?>

```

Pour lire avec la fonction *fread*, on procède de la même manière que *fgets*., mais avec *fread*, le 2<sup>e</sup> paramètre est obligatoire. Pour pouvoir lire une seule ligne, on est obligé de connaître le nombre de caractères de la ligne.

```

1 <?php
2 $fichier = fopen("infos.txt", "r+");
3 $texte= fread($fichier, 22);
4 echo "<i>1ère ligne:</i> $texte <br/>";
5 $texte= fread($fichier, 22);
6 echo "<i>2è ligne :</i> $texte <br/>";
7 fclose($fichier);
8 ?>

```

### *Lire toutes les informations d'un fichier*

Pour lire toutes les informations au sein d'un fichier :

- on utilise la fonction `filesize()` qui retourne le nombre de caractères dans un fichier. La fonction `filesize()` prend en paramètre le nom du fichier (chemin d'accès) sur le disque
- Ou bien on utilise une boucle pour parcourir ligne par ligne tout le fichier.

#### *Exemple : Lire toutes les informations dans un fichier avec fread et filesize*

la balise `<pre>` `</pre>` qui encadre le texte, permet de respecter les retours à la ligne pendant l'affichage du texte

```

1 <?php
2 echo "Le contenu du fichier est : <br/>";
3 $fichier = fopen("infos.txt", "r");
4 $texte= fread($fichier, filesize("infos.txt"));
5 echo "<pre>";
6 echo $texte;
7 echo "</pre>";
8 fclose($fichier);
9 ?>

```

#### *Exemple : Parcourir un fichier avec une boucle pour lire les informations*

- La fonction `feof` renvoie `false` si on n'a pas encore atteint la fin du fichier et `true` dans le cas contraire

```

1 <?php
2 echo "Le contenu du fichier est : <br/>";
3 $fichier = fopen("infos.txt", "r");
4 while(!feof($fichier)) {
5     echo fgets($fichier);
6     echo "<br/>";
7 }
8 fclose($fichier);
9 ?>

```

### 3. Lecture et écriture rapide dans un fichier

Il existe trois fonctions permettant la lecture rapide d'un fichier texte en PHP: `file_get_contents()`, `readfile()` et `file()`

#### Exemple : Exemple avec la fonction `file_get_contents()`

Elle permet de lire le contenu d'un fichier et le retourne dans une chaîne de caractères. Elle prend en paramètre le nom du fichier (chemin d'accès + nom)

```
1 <?php
2 $contenuFichier = file_get_contents('infos.txt');
3 echo "<pre>";
4 echo $contenuFichier;
5 echo "</pre>";
6 ?>
```

#### Exemple : Exemple avec la fonction `readfile()`

Elle permet de lire le contenu d'un fichier et retourne le nombre de caractères du fichier et affiche automatiquement le contenu.

```
1 <?php
2 echo "<pre>";
3 $nombre = readfile('infos.txt');
4 echo "</pre>";
5 echo "<br /> Le nombre de caractères du fichier est : ".$nombre;
6 ?>
```

#### Exemple : Exemple avec la fonction `file()`

Elle permet de lire le contenu d'un fichier et retourne le contenu dans un tableau

```
1 <?php
2 $tableau = file('infos.txt');
3 foreach ($tableau as $ligne) {
4     echo $ligne."<br />";
5 }
6 ?>
```

#### Écriture dans un fichier avec la fonction `file_put_contents()`

La fonction `file_put_contents()` permet d'écrire le contenu d'une chaîne de caractères dans un fichier. Elle prend en paramètres le nom du fichier et la variable contenant le texte à insérer dans le fichier. Si le fichier existe déjà, son contenu est écrasé.

```
1 <?php
2 $texte = "Bonjour les étudiants.";
3 file_put_contents("infos.txt", $texte);
4 echo "Ecriture terminée";
5 ?>
```

### *Liens utiles*

*<https://www.youtube.com/watch?v=ln8sz11hayk>*

*<https://fr.wikiversity.org/wiki/PHP/Fichiers>*

*<http://creersonsiteweb.net/page-php-lire-et-ecrire-un-fichier>*

*<http://www.opentuto.com/les-fichiers-en-php/>*

*[https://fr.wikibooks.org/wiki/Programmation\\_PHP/Fichiers](https://fr.wikibooks.org/wiki/Programmation_PHP/Fichiers)*

*<https://www.commentcamarche.com/contents/791-php-les-fichiers>*

*<https://www.pierre-giraud.com/php-mysql/cours-complet/php-lire-ouvrir-fermer-fichier.php>*

# Exercice : Activité d'auto-évaluation No 2

IV

## Exercice

Veuillez compléter le code ci-dessous pour écrire dans un fichier sans utiliser la fonction fwrite()

```
<?php
$rep="docs/";

( is_dir( ) ){

$fichier = ( "$rep"."infos.txt","a");

$texte="Iré va chercher des vers de terre \r\n";

( , $texte);

( $fichier);

}

?>
```

## Exercice

Veuillez compléter le code ci-dessous pour parcourir un fichier et afficher son contenu sans utiliser la fonction fread()

```
<?php
echo "Le contenu du fichier est : <br/>";

$fichier = ( "infos.txt","r");

( ! ( $fichier) ){

echo ( $fichier);

echo "<br/>";

}

( $fichier);

?>
```

## Exercice

La fonction ( ) permet de lire tout le contenu d'un fichier et le retourne dans une chaîne de caractères. Elle prend en paramètre le nom du fichier

## Exercice

La fonction `getchar()` permet de lire le contenu d'un fichier et retourne le nombre de caractères du fichier et affiche automatiquement le contenu.

### Exercice

---

La fonction `getchar()` permet de lire le contenu d'un fichier et retourne le contenu dans un tableau

# Les fonctions de manipulation des fichiers

V

## 1. Quelques fonctions très pratiques de manipulation de fichier

Parmi ces fonctions, nous avons déjà vu quelques unes, ci dessous une liste non exhaustive de quelques fonctions :

- *basename* : Retourne le nom de la composante finale d'un chemin
- *copy* : Copie un fichier. *Syntaxe* : `copy ($source , $dest)` ; Fait une copie du fichier source vers le fichier dest.  

```
<?php
$file = 'example.txt';
$newfile = 'example.txt.bak';
if (!copy($file, $newfile)) {
    echo "La copie $file du fichier a échoué...\n";
}
?>
```
- *dirname* : Renvoie le chemin du dossier parent
- *disk\_free\_space* : Renvoie l'espace disque disponible sur le système de fichiers ou la partition  

```
<?php
// $df contient le nombre d'octets libres sur "/"
$df = disk_free_space("/");
// Sous Windows:
$df_c = disk_free_space("C:");
$df_d = disk_free_space("D:");
?>
```
- *disk\_total\_space* : Retourne la taille d'un dossier ou d'une partition
- *fgetc* : Lit un caractère dans un fichier
- *file\_exists* : Vérifie si un fichier ou un dossier existe  

```
<?php
$filename = '/path/to/foo.txt';
if (file_exists($filename)) {
    echo "Le fichier $filename existe.";
} else {
    echo "Le fichier $filename n'existe pas.";
}
?>
```
- *fileatime* : Renvoie la date à laquelle le fichier a été accédé pour la dernière fois



- *filemtime* : Lit la date de dernière modification du fichier
- *fileowner* : Lit l'identifiant du propriétaire d'un fichier
- *filetype* : Retourne le type de fichier. Les réponses possibles sont : fifo, char, dir, block, link, file socket et unknown.
- *flock* : Verrouille le fichier
- *fseek* : Modifie la position du pointeur de fichier

```
<?php
$fp = fopen('somefile.txt', 'r');
// lit quelques données
$data = fgets($fp, 4096);
// retourne au début du fichier
// identique à rewind($fp);
fseek($fp, 0);
?>
```

- *is\_dir* : Indique si le fichier est un dossier
- *is\_executable* : Indique si le fichier est exécutable
- *is\_file* : Indique si le fichier est un véritable fichier
- *is\_readable* : Indique si un fichier existe et est accessible en lecture
- *is\_writable* : Indique si un fichier existe et est accessible en écriture

```
<?php
$filename = 'test.txt';
if (is_writable($filename)) {
    echo 'Le fichier est accessible en écriture.';
} else {
    echo 'Le fichier n\'est pas accessible en écriture !';
}
?>
```

- *mkdir* : Crée un dossier

```
<?php
mkdir("/path/to/my/dir");
?>
```

- *pathinfo* : Retourne des informations sur un chemin système

```
<?php
$path_parts = pathinfo('/www/htdocs/inc/lib.inc.php');
echo $path_parts['dirname'], "\n";
echo $path_parts['basename'], "\n";
echo $path_parts['extension'], "\n";
echo $path_parts['filename'], "\n"; // depuis PHP 5.2.0
?>
```

L'exemple ci-dessus va afficher :

```
/www/htdocs/inc
lib.inc.php
php
lib.inc
```

- *rename* : Renomme un fichier ou un dossier. *Syntaxe* : `rename ( $oldname , $newname)`, Tente de renommer `oldname` en `newname`, en le déplaçant de répertoire si nécessaire. Si vous renommez un fichier et que `newname` existe, il sera écrasé. Si vous renommez un répertoire et que `newname` existe, cette fonction émet un avertissement.

```
<?php
rename( " /tmp/tmp_file.txt" , " /home/user/login/docs/my_file.txt" );
?>
```

- *rewind* : Replace le pointeur de fichier au début

- *rmdir* : Efface un dossier

```
<?php
rmdir('docs');
?>
```

- *touch* : Modifie la date de modification et de dernier accès d'un fichier

- *unlink* : Efface ou supprime un fichier

```
<?php
unlink('test.html');
?>
```

Lien utile : <http://php.net/manual/fr/ref.filesystem.php>

## 2. Exemple de manipulation de fichier

### Libellé

Écrire un script permettant d'enregistrer un fichier pdf dans un dossier nommer "fichiers" à partir d'un formulaire.

- On doit vérifier que le fichier est de type pdf
- On doit vérifier que la taille du fichier inférieure à l'espace libre sur le disque avant de l'enregistrer
- On doit vérifier que le dossier "fichiers" existe, s'il n'existe pas, on le crée
- On doit vérifier si le fichier pdf existe déjà dans le dossier, si tel est le cas, on ajoute le timestamp au nom du fichier
- On doit avoir une page permettant de voir tous les fichiers pdf enregistrés avec la possibilité de les ouvrir ou de les supprimer

### Correction

- Nous allons créer une page appelée `index.php` (c'est la page d'accueil) qui contient le formulaire permettant d'envoyer le fichier
- La page pour enregistrer le fichier s'appellera `enregistrement.php`
- La page pour afficher les fichiers s'appellera `fichiers.php`
- Pour un meilleur affichage des fichiers, on affiche une icône pdf à coté du nom de chaque fichier pdf permettant de le télécharger
- Lorsqu'on clique sur le bouton supprimer dans le fichier "fichiers.php" on appelle la même page php en lui transmettant le nom du fichier à supprimer
- Pour afficher le contenu d'un dossier on va utiliser deux fonctions que sont : `opendir()` et `readdir()`

Le contenu de la page `index.php`

```
<?php
// ...
```

```

1 <!DOCTYPE HTML>
2 <html>
3 <head>
4   <meta http-equiv="content-type" content="text/html" />
5   <title>Inscription</title>
6   <style type="text/css">
7     form {
8       padding: 10px;
9       background-color: rgb(245,245,245);
10      padding: 10px;
11      border-radius: 7px 7px 0px 0px;
12      display: inline-block;
13      box-shadow: 0px 2px 6px rgba(32,32,32,0.5);
14      margin-top: 25px;
15    }
16    form table td {
17      padding: 5px 10px;
18    }
19    input[type="text"],input[type="email"],input[type="password"],input[type=
20    "tel"],input[type="date"],select {
21      width: 300px;
22      height: 30px;
23      border-radius: 7px;
24      border: none;
25      padding-left: 5px;
26    }
27    input[type="reset"],input[type="submit"], .bt-fichiers{
28      background-color: rgb(0,0,150);
29      color: white;
30      border: none;
31      border-radius: 7px;
32      padding: 5px 10px;
33      transition: all 0.5s;
34    }
35    input[type="reset"]{
36      background-color: red;
37    }
38    input[type="reset"]:hover,input[type="submit"]:hover, .bt-fichiers:hover{
39      background-color: rgb(255,128,0);
40      padding: 5px 25px;
41      border-radius: 10px;
42      cursor: pointer;
43    }
44    fieldset legend {
45      font-size: 20px;
46    }
47    .bt-fichiers {
48      text-decoration: none;
49      margin: 25px 50px;
50      display: inline-block;
51    }
52  </style>
53 </head>
54 <body>
55   <a class="bt-fichiers" href="fichiers.php">Afficher les fichiers uploadés</a>
56   <center>
57     <form method="post" action="enregistrement.php" enctype="multipart/form-
data">
    <fieldset>

```

```

58     <legend>Choisissez un fichier pdf</legend>
59     <table>
60         <tr>
61             <td>Nom* : </td> <td><input type="file" name="fichier" required=""
accept=".pdf" /></td>
62         </tr>
63         <tr>
64             <td><td><input type="submit" value="Envoyer"/></td>
65         </tr>
66     </table>
67 </fieldset>
68 </form>
69 </center>
70 </body>
71 </html>

```

Afficher les fichiers uploadés

### Le contenu de la page enregistrement.php

```

1 <?php
2 if(isset($_FILES["fichier"]) && !empty($_FILES["fichier"])){
3     $rep="fichiers/";
4     if(!is_dir($rep)){
5         mkdir("fichiers");
6     }
7
8     $nomfichier=utf8_encode($_FILES['fichier']['name']);
9     $tmp=$_FILES['fichier']['tmp_name'];
10    if (is_uploaded_file($tmp))
11    {
12        $taille= $_FILES['fichier']['size'];
13        if($taille<disk_free_space("C:")){
14            $tabfile = getFileExt($nomfichier); // cette fonction est definie en
bas
15            $extension = $tabfile[1];
16            if($extension == "pdf"){
17                if(file_exists($rep.$nomfichier)){
18                    $nomfichier = $tabfile[0].time().".pdf";
19                }
20                if(move_uploaded_file($tmp,$rep.$nomfichier)){
21                    echo "Le fichier a été enregistré avec succès <br/>";
22                }
23            }else echo "Le fichier envoyé n'est pas un fichier pdf <br/>";
24        }
25    }
26    }else echo "Le fichier n'a pas été uploadé <br/>";

```

```

27
28 }
29 /*
30 Cette fonction prend en paramètre un nom de fichier puis retourne un tableau dont
31 les valeurs
32 sont respectivement le nom du fichier sans son extension et l'extension du
33 fichier.
34 Ex: si le nom du fichier est maths.pdf, la fonction renvoie le tableau : array
35 ("maths","pdf");
36 */
37 function getFileExt($nomfichier) {
38     $extensionPresumee = explode('.', $nomfichier);
39     $extension= strtolower($extensionPresumee[count($extensionPresumee)-1]);
40     $nom_sans_extention=$extensionPresumee[0];
41     for ($i=1;$i<count($extensionPresumee)-1;$i++){
42         $nom_sans_extention = $nom_sans_extention .".". $extensionPresumee[$i];
43     }
44     return array($nom_sans_extention,$extension);
45 }
46
47 ?>

```

Le contenu de la page fichiers.php

```

1 <style type="text/css">
2 h2 {
3     background: aqua;
4     text-align: center;
5     margin: 25px;
6 }
7
8 .iconpdf {
9     width: 60px;
10    height: 60px;
11    vertical-align: middle;
12 }
13 ul {
14     list-style: none;
15     box-shadow: 0px 2px 8px rgba(32,32,32,0.5);
16     border-radius: 5px;
17     margin: 25px;
18 }
19 ul li {
20     border-bottom: 1px solid rgb(32,32,32,0.3);
21     height: 60px;
22 }
23 ul li:hover{
24     background-color: rgb(240,240,240);
25 }
26 </style>
27
28 <h2>Les fichiers uploadés sont:</h2>
29 <?php
30
31 $rep="fichiers/";
32
33 /*
34 Cette partie du code permet de supprimer un fichier pdf lorsqu'on clique sur le
35 bouton supprimer en bas.
36 */
37 if(isset($_GET) && ! empty($_GET["nomfichier"])){

```

```

37     unlink($rep.$_GET["nomfichier"]);
38 }
39
40
41 /*
42 Cette partie du code permet de lister les fichiers pdf du dossier "fichiers"
43 */
44 if(is_dir($rep)){
45     if($dossier= opendir($rep)){
46         echo '<ul>';
47         while($fichier=readdir($dossier)){
48             if($fichier!="." && $fichier!=".."){
49 ?>
50                 <li>
51                     <a href="<?php echo $rep.$fichier; ?>"></a>
53                     <a href="fichiers.php?nomfichier=<?php echo $fichier; ?>">Supprimer</a>
54 > | <?php echo $fichier; ?>
55                 </li>
56 <?php
57             }
58         }
59         echo '</ul>';
60     }
61 }else echo "Le dossier <strong>fichiers</strong> n'existe pas";
62 ?>

```

### Les fichiers uploadés sont:

	<a href="#">Supprimer</a>   13807.pdf
	<a href="#">Supprimer</a>   138071535316660.pdf
	<a href="#">Supprimer</a>   CVPERSO.pdf
	<a href="#">Supprimer</a>   Emploi_du_temps-prof-semaine-20-novembre.pdf
	<a href="#">Supprimer</a>   Taxonomie de Bloom.pdf

# Exercice : Activité d'auto-évaluation No 3

VI

Exercice

---

La fonction  () indique si le fichier est un véritable fichier

Exercice

---

La fonction  () vérifie si un fichier ou un dossier existe

Exercice

---

La fonction  ( \$oldname , \$newname), tente de renommer oldname en newname

Exercice

---

La fonction  () renvoie l'espace disque disponible sur le système de fichiers ou la partition

Exercice

---

La fonction  retourne des informations sur un chemin système

# Lire et écrire dans un fichier structuré

## VII

Jusqu'à présent, nous avons appris à lire et à écrire dans des fichiers texte simple non structuré. Pour une meilleure organisation des informations au sein d'un fichier, il est crucial d'enregistrer les informations de façon structurée afin de faciliter la manipulation des données.

- Une première possibilité est de nous même définir la manière de stocker des informations en utilisant des caractères spéciaux pour séparer des informations conformément à l'organisation que nous souhaitons. Une telle méthode est intéressante, toute fois, c'es nous seul qui comprenons l'organisation de nos données.
- Une deuxième méthode est d'utiliser des formats de fichiers structurés normalisés et universels qui sont connus de tous. Dans ce cas, nous avons des formats de fichiers structurés tels que les fichiers *xml*, *json*, *csv*, *yaml*, etc.

Dans cette partie du cours nous allons apprendre à enregistrer des informations de façon structurée selon un format que nous allons nous même définir, puis nous verrons comment stocker des informations dans un fichier structuré *csv*.

## 1. Enregistrement d'informations de façon structurée dans un fichier texte simple

### *Libellé*

On considère un formulaire permettant d'enregistrer des utilisateurs, contenant les champs :

- nom
- tel
- date de naissance
- email
- Ville

Écrire un script php permettant d'enregistrer dans un fichier les utilisateur qui s'inscrivent. On doit avoir aussi une page permettant de consulter les utilisateurs inscrits.

### *Correction*

- Nous allons définir comment structurer les informations à enregistrer :



- Nous enregistrerons les informations d'un utilisateur sur une seule ligne
- les informations seront enregistrées dans l'ordre énuméré et séparées par des virgules (on pourrait utiliser comme séparateur n'importe quel caractère ( : , # , \* , etc.))
- Nous appellerons le fichier dans lequel nous enregistrerons les utilisateurs users.txt
- La page qui contient le formulaire s'appellera index.php, c'est la page d'accueil
- La page d'accueil contiendra un bouton "Afficher les utilisateurs" qui permet de voir tous les utilisateurs inscrits
- La page permettant d'enregistrer les utilisateurs s'appellera inscription.php
- La page permettant de lister les utilisateurs inscrits s'appellera utilisateurs.php

Voici le contenu de la page index.php

```

1 <!DOCTYPE HTML>
2 <html>
3 <head>
4   <meta http-equiv="content-type" content="text/html" />
5   <title>Inscription</title>
6   <style type="text/css">
7     form {
8       padding: 10px;
9       background-color: rgb(245,245,245);
10      padding: 10px;
11      border-radius: 7px 7px 0px 0px;
12      display: inline-block;
13      box-shadow: 0px 2px 6px rgba(32,32,32,0.5);
14      margin-top: 25px;
15    }
16    form table td {
17      padding: 5px 10px;
18    }
19    input[type="text"],input[type="email"],input[type="password"],input[type=
"tel"],input[type="date"],select {
20      width: 300px;
21      height: 30px;
22      border-radius: 7px;
23      border: none;
24      padding-left: 5px;
25    }
26    input[type="reset"],input[type="submit"], .bt-users{
27      background-color: rgb(0,0,150);
28      color: white;
29      border: none;
30      border-radius: 7px;
31      padding: 5px 10px;
32      transition: all 0.5s;
33    }
34    input[type="reset"]{
35      background-color: red;
36    }
37    input[type="reset"]:hover,input[type="submit"]:hover, .bt-users:hover{
38      background-color: rgb(255,128,0);
39      padding: 5px 25px;
40      border-radius: 10px;
41      cursor: pointer;
42    }
43    fieldset{

```

```

44     border: 2px dashed rgb(255,128,0);
45 }
46 fieldset legend {
47     font-size: 20px;
48     color: rgb(0,0,100);
49     font-style: italic;
50 }
51 .bt-users{
52     display: inline-block;
53     margin: 10px 50px;
54     text-decoration: none;
55 }
56 </style>
57 </head>
58 <body>
59     <a class="bt-users" href="utilisateurs.php">Voir les utilisateurs</a>
60     <center>
61         <form method="post" action="inscription.php">
62             <fieldset>
63                 <legend>Inscription Utilisateur</legend>
64                 <table>
65                     <tr>
66                         <td>Nom* : </td> <td><input type="text" name="nom" required=""
67 /></td>
68                     <tr>
69                         <td>Email : </td> <td><input type="email" name="email" /></td>
70                     <tr>
71                         <td>Tel* : </td> <td><input type="tel" name="tel" required="" /></td>
72                     <tr>
73                         <td>Date de naissance : </td> <td><input type="date" name=
74 "date_naissance" placeholder="Ex: 20/10/2000" /></td>
75                     <tr>
76                         <td>Ville de residence : </td>
77                         <td>
78                             <select name="ville">
79                                 <option></option>
80                                 <option>Abidjan</option>
81                                 <option>Abengourou</option>
82                                 <option>Bouaké</option>
83                                 <option>Daloa</option>
84                                 <option>Gagnoa</option>
85                                 <option>Korogho</option>
86                                 <option>Man</option>
87                                 <option>Yamoussoukro</option>
88                             </select>
89                         </td>
90                     <tr>
91                         <td><input type="reset" value="Annuler" /></td> <td><input type=
92 "submit" value="Valider"/></td>
93                     <tr>
94                         <td><input type="reset" value="Annuler" /></td> <td><input type=
95 "submit" value="Valider"/></td>
96                     <tr>
97                         <td><input type="reset" value="Annuler" /></td> <td><input type=
98 "submit" value="Valider"/></td>
99                 </table>
100             </fieldset>
101         </form>

```

```

100     </center>
101 </body>
102 </html>

```

Voir les utilisateurs

Voici le contenu de la page inscription.php

```

1 <?php
2 if(isset($_POST) && !empty($_POST)){
3     $rep="users.txt";
4     $fichier= fopen($rep,"a+");
5     $texte= $_POST['nom'].".".$_POST['tel'].".".$_POST['date_naissance'].".".$_POST['email'].".".$_POST['ville']."\r\n";
6     if(fputs($fichier,$texte)){
7         echo "<h2><center> Enregistrement effectué avec succès! </center></h2>";
8     }else{
9         echo "<h2><center>Enregistrement échoué!</center></h2>";
10    }
11    fclose($fichier);
12 }
13
14 ?>

```

Voici le contenu de la page utilisateurs.php

```

1 <style type="text/css">
2     table {
3         border-collapse: collapse;
4     }
5     tr:nth-child(odd){
6         background-color: rgb(245,245,245);
7     }
8     td,th {
9         font-size: 1.2em;
10        padding: 5px 15px;
11        border: 1px solid rgba(32,32,32,0.5);
12    }
13    h2 {
14        text-align: center;
15        margin: 25px 0px;
16        color: rgb(0,0,150);

```

```

17     background-color: aqua;
18 }
19 </style>
20 <h2>Les utilisateurs inscrits sont:</h2>
21 <?php
22 $rep="users.txt";
23 if(file_exists($rep)){
24     $fichier= fopen($rep,"r");
25     echo "<center><table>";
26     echo " <tr><th>Nom</th> <th>Tel</th> <th>Date de Naissance</th> <th>Email<
    /th> <th>Ville</th></tr>";
27     while(!feof($fichier)){
28         $texte= fgets($fichier);
29         if(!empty($texte)){
30             $user = explode(",", $texte); //on transforme la chaine en tableau
31             echo "<tr>";
32             echo "<td>$user[0]</td> <td>$user[1]</td> <td>$user[2]</td> <td>$user[
33 3]</td> <td>$user[4]</td>";
34             echo "</tr>";
35         }
36     }
37     echo "</table> </center>";
38     fclose($fichier);
39 }else echo "<h2><center>Le Fichier $rep n'a pas été trouvé </center></h2>";
40 ?>

```

### Les utilisateurs inscrits sont:

Nom	Tel	Date de Naissance	Email	Ville
Yeo Zan	09874526	2018-08-07	yeo@yahoo.fr	Bouaké
Zamblé Jean	78956329	2018-08-20	jean@gmail.com	Korogho
Anne Loto	05789632	2010-08-20	loto@gmail.com	Daloa
Jean Jacques Kouamé	01785423	2008-07-20	jacque@gmail.com	Yamoussoukro

### Remarque

Dans l'exemple ci-dessus, nous avons opté pour une structure de fichier proche du format csv. Toute fois, nous étions libre d'organiser les informations comme bon nous semble. Par exemple, enregistrer chaque champ sur une ligne et séparer les informations des différents utilisateurs par une ligne vide ou une ligne contenant un caractère spécial. On pouvait même intégrer le nom des champs dans l'enregistrement des informations, etc. En fonction de l'organisation définie pour enregistrer les informations, nous devons nous référer pour récupérer les informations.

Dans la suite du cours nous allons voir comment organiser des informations selon le format csv, qui impose une manière bien définie de structurer les informations.

## 2. Enregistrement d'informations de façon structurée dans un fichier texte csv

### Présentation d'un fichier csv

- Le sigle CSV signifie Comma-Separated Values et désigne un fichier informatique de type tableur.

- Chaque ligne du fichier correspond à une ligne du tableau. Les valeurs de chaque colonne du tableau sont séparées par un caractère de séparation, en général une virgule ou un point-virgule. Chaque ligne est terminée par un caractère de fin de ligne (line break). Généralement, les valeurs sont entre double quote.
- Toutes les lignes contiennent obligatoirement le même nombre de valeurs (donc le même nombre de caractères de séparation). Les valeurs vides doivent être exprimées par deux caractères de séparation contigus.
- La taille du tableau est le nombre de lignes multiplié par le nombre de valeurs dans une ligne.
- La première ligne du fichier peut être utilisée pour exprimer le nom des colonnes.
- Le format CSV est un format de texte simple qui est utilisé dans de nombreux contextes lorsque de grandes quantités de données doivent être fusionnées sans être directement connectées les unes aux autres.
- L'extension de ce type de fichiers est .csv, et ils peuvent être utilisés entre différents outils informatiques et bases de données, lorsqu'on souhaite déployer le contenu d'une base de données sur une feuille de calcul.
- Des tableurs tels qu'Excel (Microsoft) ou Calc (LibreOffice) et des bases de données telles que MySQL et Oracle sont capables d'importer et exporter des fichiers CSV. Toutefois, en raison de sa structure basique, le format de fichier CSV ne convient que pour des données structurées simples

#### Exemple de fichier csv

Fichier au format .csv	Représentation tabulaire		
Sexe,Prénom,Année de naissance M,Alphonse,1932 F,Béatrice,1964 F,Charlotte,1988	Sexe	Prénom	Année de naissance
	M	Alphonse	1932
	F	Béatrice	1964
	F	Charlotte	1988

#### Écrire et lire dans un fichier csv

- Nous pouvons créer un fichier csv et écrire des informations à l'intérieur comme nous le faisons dans un fichier texte simple, mais en respectant le format des informations au sein d'un fichier csv
- Nous pouvons également créer un fichier csv et écrire des informations à l'intérieur en faisant appel à une fonction php conçu pour ce fait.

Dans cette partie, nous allons voir la démarche pour écrire et lire des informations dans un fichier csv selon une méthode définie par php. La méthode consiste à :

- mettre dans un tableau les informations à écrire sur une ligne
- puis on appelle la fonction `fputcsv` pour écrire dans le fichier. La *syntaxe* est `fputcsv ($fichier_csv, $ligne, $delimiteur);`



```
1 <?php
2 $rep="users.csv";
3 $delimiteur=",";
4 $fichier= fopen($rep,"a");
5 $info = array("Koffi Jean","78541236","28-05-2015","jean@yahoo.fr");
6 fputcsv($fichier,$info,$delimiteur);
7 $info = array("Zamblé Paul","05412369","01-12-1990","paul@gmail.com");
8 fputcsv($fichier,$info,$delimiteur);
9 fclose($fichier);
10 echo "<h2><center>Informations enregistrées avec succès!</center></h2>";
11 ?>
```

Si vous voulez écrire dans un format exportable sous excel, il faut utiliser comme délimiteur le point virgule ";"

## Lire dans un fichier csv

Pour lire dans un fichier csv, on utilise la fonction *fgetcsv* dont la *syntaxe* est : `fgetcsv($fichier, nombre_caractère, $delimiteur);`

- Si le nombre de caractères et le délimiteur sont renseignés, *fgetcsv* retourne un tableau dont les éléments sont les différentes valeurs que sépare le délimiteur
- Si le nombre de caractères ou le délimiteur n'est pas précisé, *fgetcsv* retourne un tableau à un seul élément contenant une chaîne de caractères représentant les informations de la ligne.

```
1 <?php
2 $rep="users.csv";
3 $delimiteur=",";
4 $fichier= fopen($rep,"r");
5 $user= fgetcsv($fichier,1024,$delimiteur);
6 print_r($user);
7 echo "<br/>";
8 $user= fgetcsv($fichier,1024,$delimiteur);
9 print_r($user);
10 fclose($fichier);
11 ?>
```

### 3. Travaux dirigés

*Libellé*

Dans la leçon 4 du cours, nous avons réalisé à la fin de la leçon un formulaire qui permet d'inscrire des étudiants. Malheureusement, les informations disparaissent une fois on quitte la page.

Dans ce travail dirigé, vous devez enregistrer les informations du formulaire dans un fichier csv, puis on crée une page où on affiche les informations (avec les photos) de tous les étudiants inscrits avec la possibilité de les supprimer.







### Correction

- Pour le contenu de la page index.php, veuillez voir la leçon précédente. Toute fois, il faut mettre un bouton sur cette page permettant de lister les étudiants. Le bouton à mettre est :

```
<a class="btn btn-sm btn-primary bt-users" href="etudiants.php">Voir  
les étudiants</a>
```

- Le fichier contenant les informations sur les étudiant s'appellera `etudiants.csv`, la première ligne de ce fichier va représenter les différents champs
- On utilisera comme délimiteur le point virgule ( ;) afin de faciliter l'ouverture du fichier sous excel.
- Le fichier permettant de lister les étudiants s'appellera `etudiants.php`
- Les images seront dans un dossier `users/` comme à la leçon 4
- Les mots de passe seront cryptés et décryptés grâce à deux fonctions que nous allons écrire. En effet, pour une question de sécurité, il n'est pas intéressant de stocker directement les mots de passe dans un fichier sans les crypter.
- Les différentes fonctions que nous aurons à écrire seront dans un fichier php nommé `fonctions.php` que nous devons inclure dans les autres pages afin de pouvoir les utiliser.

### Structure du projet

Disque local (C:) ▸ wamp ▸ www ▸ helloWorld			
Nom	Modifié le	Type	Taille
 bootstrap	19/08/2018 10:45	Dossier de fichiers	
 js	27/08/2018 01:55	Dossier de fichiers	
 etudiants.php	27/08/2018 01:49	PHP Script	3 Ko
 fonctions.php	27/08/2018 00:41	PHP Script	2 Ko
 index.php	27/08/2018 00:34	PHP Script	8 Ko
 inscription.php	27/08/2018 00:59	PHP Script	2 Ko

- Le fichier `etudiants.csv` et le dossier `users` seront automatiquement créés lorsque le programme sera exécuté.
- Contrairement au projet de la leçon 4, le dossier `bootstrap` ne porte pas ici le numéro de version. Il faut donc l'adapter dans l'entête de la page

La page `index.php`

[Voir les étudiants](#)

Nom* :	<input type="text"/>	Au moins 2 Caractères et commencé par une majuscule
Tel* :	<input type="text"/>	
Email :	<input type="text"/>	@
Date de Naissance :	<input type="text" value="jj/mm/aaaa"/>	
Matricule* :	<input type="text" value="Ex: UV4578AX"/>	Le matricule doit commencer par UV suivi de 4 chiffres, puis se termine par deux lettres Majuscule
Ville	<input type="text"/>	
Sexe :	<input type="radio"/> Masculin <input type="radio"/> Féminin	
Mot de passe:	<input type="password"/>	Minimum 8 caractères
Confirmez le mot de passe:	<input type="password"/>	Minimum 8 caractères
Photo:	<input type="button" value="Choisir une image"/>	
<input type="button" value="Valider"/>		

Voici le contenu de la page fonctions.php

```

1 <?php
2
3 function getFileUniqueName($nomfichier,$repertoire) {
4     while(file_exists($repertoire.$nomfichier)){
5         $stab= getFileExt($nomfichier);
6         $suffixe= rand(1,10000000);
7         $nomfichier=$stab[0].$suffixe.".". $stab[1];
8     }
9     return $nomfichier;
10 }
11 function getFileExt($nomfichier) {
12     $extensionPresumee = explode('.', $nomfichier);
13     $extension= strtolower($extensionPresumee[count($extensionPresumee)-1]);
14     $nom_sans_extension=$extensionPresumee[0];
15     for ($i=1;$i<count($extensionPresumee)-1;$i++){
16         $nom_sans_extension = $nom_sans_extension .".". $extensionPresumee[$i];
17     }
18     return array($nom_sans_extension,$extension);
19 }
20
21 function crypter($chaine,$cle="Blender blendé 100000W"){
22     $chaine_crypt = mcrypt_ecb(MCRYPT_TripleDES,$cle,$chaine,MCRYPT_ENCRYPT);
23     return $chaine_crypt;
24 }
25 function decrypter($chaine_crypt,$cle="Blender blendé 100000W"){
26     $chaine = mcrypt_ecb(MCRYPT_TripleDES,$cle,$chaine_crypt,MCRYPT_DECRYPT);
27     return $chaine;
28 }
29 ?>

```

Voici le contenu de la page inscription.php

```

1 <style type="text/css">
2 table {

```



```

3     border-collapse: collapse;
4 }
5 table td {
6     padding: 5px 15px;
7     border: 1px solid rgba(32,32,32,0.5);
8 }
9 table tr:nth-child(odd) {
10     background-color: aqua;
11 }
12 .photo {
13     height: 120px;
14     width: auto;
15     border-radius: 7px;
16 }
17 </style>
18 <?php
19 //Inclusion du fichier fonction.php;
20 require_once("fonctions.php");
21
22 $repertoire="users/";
23
24 // si le dossier users n'existe pas on le crée'
25 if(!is_dir($repertoire)){
26     mkdir("users");
27 }
28
29 //Enregistrement de l'image
30 $photo="";
31 if(isset($_FILES['photo']) && !empty($_FILES['photo']['name'])){
32     $nomfichier= utf8_encode($_FILES['photo']['name']);
33     $tmp=$_FILES['photo']['tmp_name'];
34     if (is_uploaded_file($tmp))
35     {
36         if(file_exists($repertoire)){
37             if(file_exists($repertoire.$nomfichier)){
38                 $nomfichier =getFileUniqueName($nomfichier,$repertoire);
39             }
40             if(move_uploaded_file($tmp,$repertoire.$nomfichier)){
41                 $photo=$nomfichier;
42             }
43         }else echo "Le dossier $repertoire n'existe pas <br/>";
44         }else echo "Le fichier n'a pas été uploadé <br/>";
45 }else echo "Pas de photo uploadée <br/>";
46
47 //Enregistrement des informations dans le fichier etudiants.txt
48
49 $_POST["photo"]=$photo;
50 //On crypte le mot de passe
51 $_POST["motdepasse"]=crypter($_POST["motdepasse"]);
52 $delimiteur="";
53 // Si le fichier n'existe pas, on le crée et on insère comme première ligne les
    noms des champs
54 if(!file_exists("etudiants.csv")){
55     $cles = array("Nom","Tél","Email","Date de Naissance","Matricule","Ville",
        "Sexe","Mot de passe","Photo");
56     $fichier= fopen("etudiants.csv","a+");
57     fputcsv($fichier,$cles,$delimiteur);
58 }else {
59     $fichier= fopen("etudiants.csv","a+");

```

```

60 }
61 //on met les valeurs du formulaire dans un tableau
62 $values = array_values($_POST);
63 fputcsv($fichier,$values,$delimiteur);
64 fclose($fichier);
65 echo "<h2><center>Inscription effectuée avec succès!</center></h2>";
66 ?>
67

```

Voici e contenu de la page etudiants.php

```

1 <style type="text/css">
2     table {
3         border-collapse: collapse;
4     }
5     tr:nth-child(odd) {
6         background-color: rgb(250,250,250);
7     }
8     td,th {
9         font-size: 1.2em;
10        padding: 5px 15px;
11        border: 1px solid rgba(32,32,32,0.5);
12    }
13    h2 {
14        text-align: center;
15        margin: 25px 0px;
16        color: rgb(0,0,150);
17        background-color: aqua;
18    }
19    .photo {
20        height: 100px;
21        width: auto;
22    }
23 </style>
24
25 <h2>Liste des étudiants:</h2>
26 <?php
27 //Inclusion du fichier fonction.php;
28 require_once("fonctions.php");
29
30 $rep="etudiants.csv";
31 $delimiteur=";";
32 /*
33 Cette partie du code permet de supprimer un étudiant lorsqu'on clique sur
34 supprimer en bas.
35 */
36 if(isset($_GET) && ! empty($_GET["matricule"])){
37     $files= file($rep);
38     $fichier= fopen($rep,"r");
39     $i=0;
40     while(!feof($fichier)){
41         $user= fgetcsv($fichier,1024,$delimiteur);
42         if(!empty($user)){
43             if(in_array($_GET["matricule"],$user)){
44                 unset($files[$i]);
45                 //on supprime la photo de l'étudiant si elle existe
46                 $photo=$user[sizeof($user)-1];
47                 if(!empty($photo)){
48                     unlink("users/".$photo);
49                 }
50             }
51         }
52     }
53     $files=array_values($files);
54     $delimiteur=";";
55     $fichier=fopen($rep,"w");
56     foreach($files as $user){
57         fputcsv($fichier,$user,$delimiteur);
58     }
59     fclose($fichier);
60 }
61 //on met les valeurs du formulaire dans un tableau
62 $values = array_values($_POST);
63 fputcsv($fichier,$values,$delimiteur);
64 fclose($fichier);
65 echo "<h2><center>Inscription effectuée avec succès!</center></h2>";
66 ?>
67

```

```

48         }
49
50     }
51 }
52 $i++;
53 }
54 fclose($fichier);
55 $filerecord=implode(',',$files);
56 file_put_contents($rep,$filerecord);
57 }
58
59
60 /*
61 Cette partie du code permet de lister les fichiers pdf du dossier "fichiers"
62 */
63 if(file_exists($rep)){
64
65     $fichier= fopen($rep,"r");
66     echo "<center><table>";
67     $entete= fgetcsv($fichier,1024,$delimiteur);
68     echo "<tr>";
69     foreach($entete as $valeur){
70         echo "<td> $valeur </td>";
71     }
72     echo "<td>Supp</td>";
73     echo "</tr>";
74     while(!feof($fichier)){
75         $user= fgetcsv($fichier,1024,$delimiteur);
76         if(!empty($user)){
77             echo "<tr>";
78             for($i=0;$i<sizeof($user)-2;$i++){
79                 $valeur= $user[$i] ;
80                 echo "<td> $valeur </td>";
81             }
82             // Si le mot de passe est non vide,on le décrypte et on l'affiche
83             if(!empty($user[$i])){
84                 $password= decrypter($user[$i]);
85                 echo "<td> $password </td>";
86             }else {
87                 echo "<td></td>";
88             }
89             $i++;
90             // Si la photo est non vide,on l'affiche, dans le cas contraire, on
            affiche une image par défaut
91             if(!empty($user[$i])){
92                 $source= "users/".$user[$i];
93                 echo "<td> <img src='$source' class='photo' /> </td>";
94             }else {
95                 echo "<td> <img src='users/user.png' class='photo' /> </td>";
96             }
97             echo "<td> <a href='etudiants.php?matricule=$user[4] '>XXx</a></td>";
98             echo "</tr>";
99         }
100
101     }
102     echo "</table></center>";
103     fclose($fichier);
104

```

```

105 }else echo "Le fichier <strong>$rep</strong> n'existe pas";
106
107 ?>

```

#### Liste des étudiants:

Nom	Tél	Email	Date de Naissance	Matricule	Ville	Sexe	Mot de passe	Photo	Supp
Jean ping	78452396	ping@yahoo.fr	2018-08-08	UV7845UB	Bouaké	M	ping0001		<a href="#">xXx</a>
Fatim Coulibaly	01789652	fatim@yahoo.fr	2018-08-10	UV7241UB	Abidjan	F	fatima001		<a href="#">xXx</a>
Jean jacque rousseau	45789632	jack@yahoo.fr	2018-08-30	UV7240DC	Korogho	M	rousseau		<a href="#">xXx</a>
Penelope	67896542	penel@gmail.com	2018-08-15	UV7896PO	Abidjan	F			<a href="#">xXx</a>

# Exercice : Activité d'auto-évaluation No 4

VIII

## Exercice

---

Quelques fichiers structurés sont : (4 réponses)

- ☐ yaml
- ☐ cvs
- ☐ xml
- ☐ json
- ☐ jackson
- ☐ pdf
- ☐ csv
- ☐ word

## Exercice

---

Veuillez compléter le code ci-dessous pour écrire dans un fichier csv

```
<?php
$rep="users.csv";
$delimiteur=",";
$fichier= ( $rep,"a+");
$info = ("Ben coulibaly","07896542","22-07-2000","ben@yahoo.fr");
( $fichier,$info, );
( $fichier);
?>
```

## Exercice

---

Veuillez compléter le code ci-dessous pour lire dans un fichier csv

```
<?php
$rep="users.csv";
```

```
$delimiteur=",";  
$fichier= ( $rep,"r");  
$user= ( ,1024, );  
print_r($user);  
( $fichier);  
?>
```