

# *Apprentissage et Pratique du Langage PHP*

*Intégration de Fichiers externes,  
Gestion des sessions,  
Gestion des fichiers,  
Atelier*

*Par  
Elhassan Abdelwahed  
Département d'Informatique  
Faculté des Sciences Semlalia Marrakech*

# Fichiers externes

Lorsqu'un même script doit être utilisé dans des pages Html différentes et les parties de contenu qui sont communes à toutes vos pages (entête de page, pied de page, menu principal..) sont enregistrées dans différents fichiers.

Lorsque vous créez une nouvelle page vous pouvez appeler ces fichiers et les replacer au on endroit grâce à la fonction **include()** ou **require()**. Au endroit ou les fichiers sont inclus, PHP recopiera le contenu de chaque fichier appelé.

La différence importante entre require() et include() se situe au niveau de la gestion d'erreur. Include() renvoi qu'un warning en cas d'erreur, alors que require() provoquera une erreur fatale() qui empêchera l'exécution des scripts qui suivront.

```
<html>
  <head>    <title>Titre de la page</title> </head>
  <body>
    <?php include("entete_inc.php"); ?>
    <?php include("menu_inc.php"); ?>
    <p>contenu html</p>
    <?php include("pied_inc.php"); ?>
  </body>
</html>
```

# Fichiers externes

Lorsqu'un même script doit être utilisé dans des pages Html différentes, afin de faciliter sa maintenance, il est conseillé d'écrire ce script dans un fichier texte séparé.

PHP possède deux fonctions, **require** et **include**, qui concernent l'inclusion de fichiers externes du type texte (HTML, JavaScript, PHP, texte, ... etc. )

**<?php require ("MesScripts.php"); ?>**

A l'exécution, cette instruction est remplacée par le contenu de « MesScripts.php » et provoque une erreur bloquante s'il est indisponible.

**<?php include ("MesScripts.php"); ?>**

Inclut le contenu de "MesScripts.php", génère un warning mais ne provoque pas d'erreur bloquante s'il est indisponible.

**Remarques :** Les fichiers externes sont cherchés dans le répertoire spécifié dans la directive "include\_path" du fichier php.ini

# Sessions

Une session assure le passage et l'accessibilité de valeurs des variables entre pages HTML (ou PHP). Ces valeurs sont en fait sauvegardées dans une variable globale `$_SESSION[ ]` qui est un tableau associatif.

Contrairement aux autres variables sa durée de vie ne se limite pas à la durée de vie de la page. Ces valeurs sont stockées sur le serveur et peuvent donc être réutilisées sur n'importe quelles pages du même site.

## Exemple:

- Les sites marchands utilisent, par exemple, les sessions pour la création de paniers.
- Les sessions sont aussi utilisées sur les sites qui proposent des accès membres via un formulaire d'identification.

# Sessions

---

**La fonction `session_start()` qui est obligatoire va démarrer une nouvelle session. Elle doit être appelée avant tout affichage de contenu dans la page.**

**Au démarrage d'une session PHP va créer un fichier texte (coté serveur). Le chemin de ce fichier sera enregistrer dans un Cookie (coté client). Par défaut l'identifiant du Cookie est `PHPSESSID`.**

**La fonction `session_start()` va donc lire ou créer le cookie pour pouvoir accéder au fichier distant. Donc si les cookies ne sont pas activés, les sessions ne seront pas utilisables.**

**Ensuite les données du fichier seront lues par PHP (mode lecture) et stockées en mémoire dans la superglobale `$_SESSION[]`. A l'inverse si vous remplissez la variable `$_SESSION[]`, PHP va à la fin du script écrire les valeurs de la superglobale dans le fichier (mode écriture).**

# Sessions

---

## **session\_start()**

Démarre une session ou appelle la session existante

## **session\_destroy()**

Met fin à la session et efface les données.

## **session\_name()**

Retourne le nom de la session en cours

## **session\_id()**

Retourne le code de la session en cours

## **session\_is\_registered()**

Vérifie si une variable est enregistrée dans la session

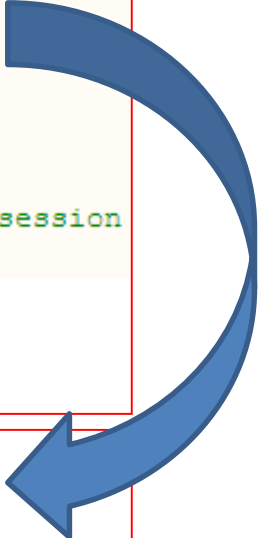
## **session\_register()**

Enregistre une variable dans la session courante: : session\_register(\$var) indique à PHP que la variable \$var doit être enregistrée dans la session en cours. En plus cela indique à PHP de modifier la valeur enregistrée dans la session lorsque la valeur dans le script change.

**\$\_SESSION :** Il s'agit du tableau global contenant toutes les variables de sessions pour la session courante.

# Sessions

```
<html>
<head> <title> Gestion des sessions </title> </head>
<body>
    <?php
        session_start(); // à mettre tout en haut de la page
        //.... autres traitements
        $nb = 2;
        $_SESSION['nombre'] = $nb; // on enregistre la valeur 2 dans la session
        $achat = array("fruit", "céréale", "légumes");
        $_SESSION['mesAchats'] = $achat; // on enregistre le tableau $achat dans la session
        echo '<br /><a href="session2.php">page 2</a>';
    ?>
</body>
</html>
```



```
<html>
<head> <title> Gestion des sessions </title> </head>
<body>
    <?php
        session_start(); // à mettre tout en haut de la page
        //.... autres traitements
        $n = $_SESSION['nombre']; // On récupère la valeur sauvegardée dans la session
        echo $n.'<br />';
        $achat = $_SESSION['mesAchats']; // on affecte $achat par la valeur contenu dans la session
        echo $achat[0].<br />';
        echo '<br /><a href="session1.php">page 1</a>';
        $_SESSION = array(); // supprime les variables de la session
        session_destroy(); // supprime le fichier sur le serveur
    ?>
</body>
</html>
```

## Tableaux, session, ... etc.

### Objectifs de l'atelier 3

- Reprendre les travaux réalisés dans le cadre de l'atelier 2 pour la gestion des notes et l'améliorer en prenant en considération plusieurs étudiants
- Mise en œuvre des différentes notions acquises (tableaux, sessions, ... etc.) dans le cadre du projet de gestion des notes.

#### Remarque

Le programme de gestion des notes va évoluer en y intégrant à chaque nouveau chapitre les différentes notions étudiées.



# Fichiers

- La manipulation de fichier se fait grâce à un identifiant de fichier.
- Quelques fonctions:

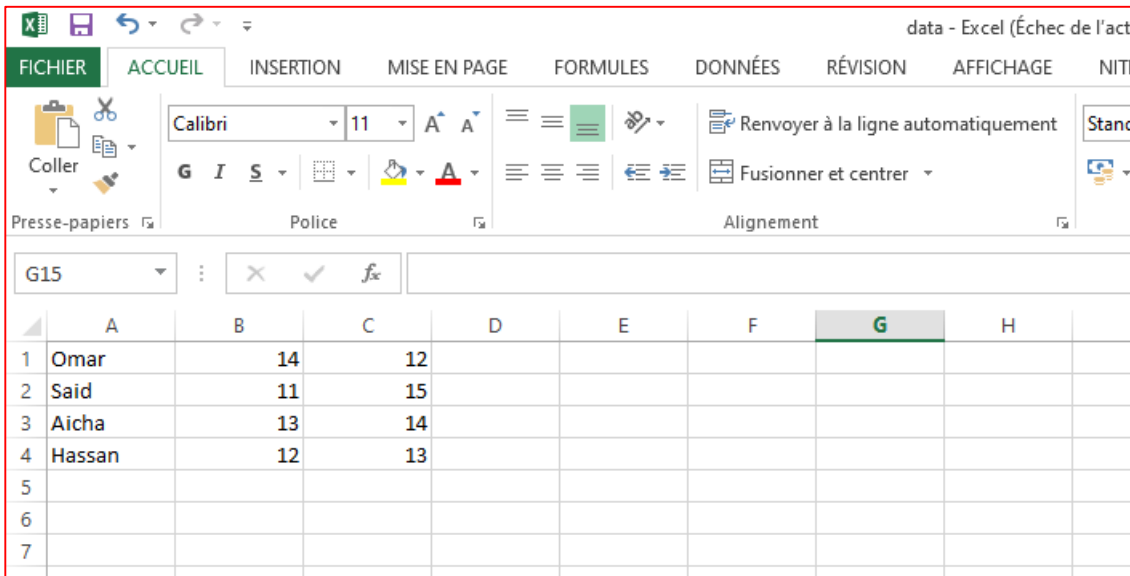
<b>fopen(\$file [, \$mode]) :</b>	ouverture du fichier identifié par son nom <b>\$file</b> et dans un mode <b>\$mode</b> particulier, retourne un identificateur <b>\$fp</b> de fichier ou <b>FALSE</b> si échec
<b>fopen(\$fp) :</b>	ferme le fichier identifié par le <b>\$fp</b>
<b>fgets(\$fp, \$length) :</b>	lit une ligne de <b>\$length</b> caractères au maximum
<b>fputs(\$fp, \$str) :</b>	écrit la chaîne <b>\$str</b> dans le fichier identifié par <b>\$fp</b>
<b>fgetc(\$fp) :</b>	lit un caractère
<b>feof(\$fp) :</b>	teste la fin du fichier
<b>file_exists(\$file) :</b>	indique si le fichier <b>\$file</b> existe
<b>filesize(\$file) :</b>	retourne la taille du fichier <b>\$file</b>
<b>filetype(\$file) :</b>	retourne le type du fichier <b>\$file</b>
<b>unlink(\$file) :</b>	détruit le fichier <b>\$file</b>
<b>copy(\$source, \$dest) :</b>	copie le fichier <b>\$source</b> vers <b>\$dest</b>
<b>rename(\$old, \$new) :</b>	renomme le fichier <b>\$old</b> en <b>\$new</b>

## Modes de fichiers possibles

- r:** ouverture en lecture seulement
- w:** ouverture en écriture seulement (la fonction crée le fichier s'il n'existe pas)
- a:** ouverture en écriture seulement avec ajout du contenu à la fin du fichier (la fonction crée le fichier s'il n'existe pas)
- r+:** ouverture en lecture et écriture
- w+:** ouverture en lecture et écriture (la fonction crée le fichier s'il n'existe pas)
- a+:** ouverture en lecture et écriture avec ajout du contenu à la fin du fichier (la fonction crée le fichier s'il n'existe pas)

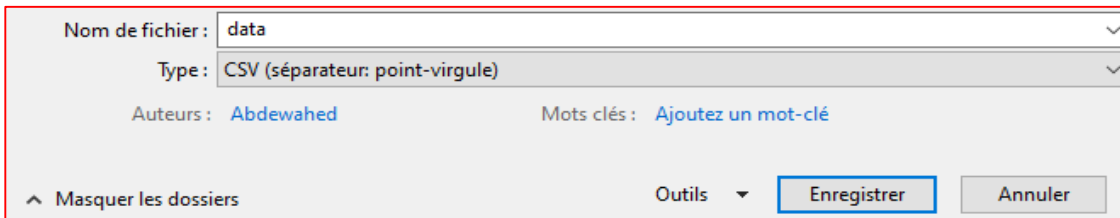
```
<html>
<head> <title> Fichier en PHP </title> </head>
<body>
    <?php
        if (!$fp = fopen("data.txt","r")) { echo "Echec de l'ouverture du fichier"; exit; }
        else {
            $Fichier = "";
            while(!feof($fp)) {
                $Ligne = fgets($fp,255);
                echo $Ligne."<br/>";
                $Fichier .= $Ligne."<br/>";
            }
            fclose($fp);
        }
        echo $Fichier;
    ?>
</body>
</html>
```

# Exemple : Fichier CSV, Sauvegarde de données



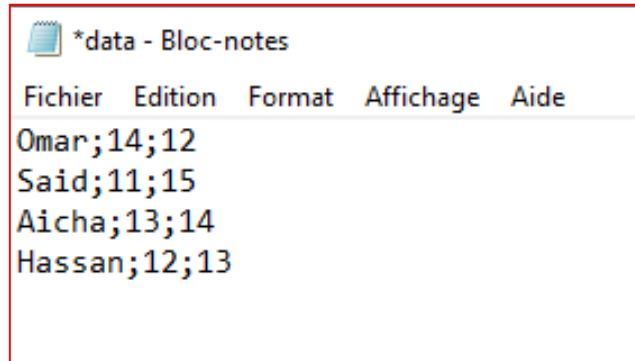
The screenshot shows the Microsoft Excel interface. The ribbon is set to 'FICHIER'. The data is as follows:

	A	B	C	D	E	F	G	H
1	Omar	14	12					
2	Said	11	15					
3	Aïcha	13	14					
4	Hassan	12	13					
5								
6								
7								



The 'Enregistrer' (Save) dialog box is shown. The file name is 'data'. The type is 'CSV (séparateur: point-virgule)'. The author is 'Abdewahed'. The 'Enregistrer' button is highlighted.

Nom de fichier : data  
Type : CSV (séparateur: point-virgule)  
Auteurs : Abdewahed  
Mots clés : Ajoutez un mot-clé  
Masquer les dossiers Outils Enregistrer Annuler



The Notepad window shows the saved CSV data, with each row separated by a line break and fields separated by semicolons.

```
*data - Bloc-notes
Fichier Edition Format Affichage Aide
Omar;14;12
Said;11;15
Aïcha;13;14
Hassan;12;13
```

# Exemple

## Transfert de données de la RAM vers le disque dur

### Ecriture des données issues de tableaux vers un fichier au format CSV

```
<?php
    $NbEtudiants = 3;
    $Nom = array('Mostafa', 'Hassan', 'Mostafa');
    $Info = array(14, 15, 15);
    $Math = array(12, 13, 14);
    $fp = fopen("data.txt", "w");
    for ($c=0; $c < $NbEtudiants; $c++) {
        $Ligne[0] = $Nom[$c];
        $Ligne[1] = $Info[$c];
        $Ligne[2] = $Math[$c];
        $L = $Ligne[0].";". $Ligne[1].";". $Ligne[2]."\n";
        fwrite($fp, $L);
    }
    fclose($fp);
?>
```

Version 1

```
<?php
    $NbEtudiants = 3;
    $Nom = array('Jamal', 'Hassan', 'Amina');
    $Info = array(14, 15, 15);
    $Math = array(12, 13, 14);
    $fp = fopen("data.txt", "w");
    for ($c=0; $c < $NbEtudiants; $c++) {
        $Ligne[0] = $Nom[$c];
        $Ligne[1] = $Info[$c];
        $Ligne[2] = $Math[$c];
        fputcsv($fp, $Ligne, ";");
    }
    fclose($fp);
?>
```

Version 2

# Exemple

## Transfert de données du disque dur vers la RAM

### Lecture des données stockées dans un fichier au format CSV vers des tableaux

<?php

Version 1

```
if (!$fp = fopen("data.txt","r"))
{ echo "Echec de l'ouverture du fichier"; exit; }
else {
while(!feof($fp)) {
    $Ligne = fgets($fp,250);
    if ( strlen ( $Ligne ) > 0 )
    {
        $Tab = explode(';', $Ligne, 3); echo "$Tab[0] $Tab[1] $Tab[2] <br/>"; }
    }
fclose($fp);
}
```

?>

<?php

Version 2

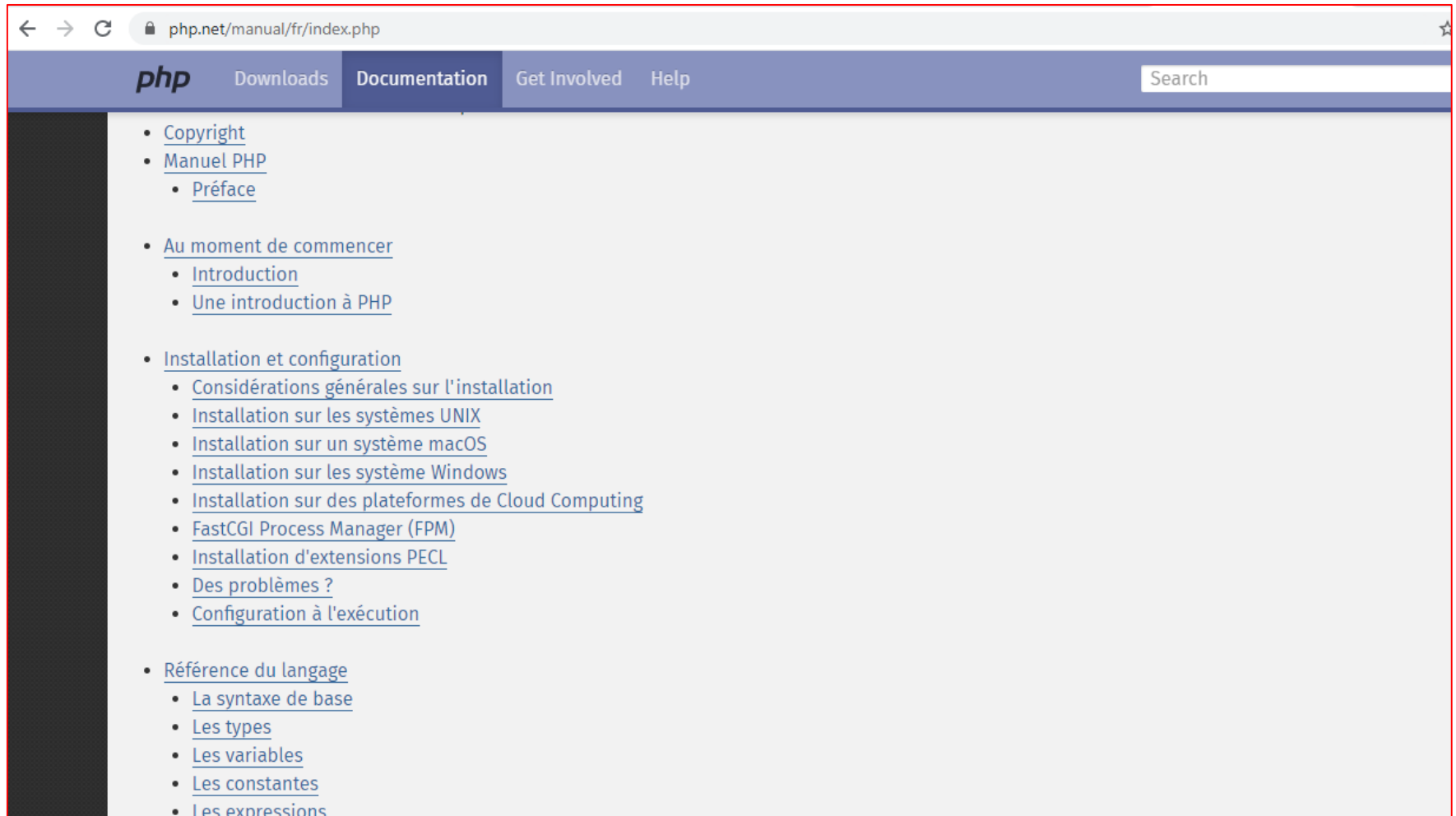
```
$NbLigne = 0;
if (!$fp = fopen("data.txt","r")) { echo "Echec de l'ouverture du fichier"; exit; }
else
{
    while (($Ligne = fgetcsv($fp, 1000, ";")) !== FALSE) {
        $NbLigne++;
        $num = count($Ligne);
        for ($c=0; $c < $num; $c++) { echo $NbLigne." ".$Ligne[$c] . "<br />\n"; }
    }
    fclose($fp);
}
```

?>

# Manuel officiel de PHP

Pour plus d'informations (fonctions, ...etc.) Consultez le manuel officiel de PHP :

<https://www.php.net/manual/fr/index.php>



## Gestion des fichiers

### Objectifs de l'atelier 4

- Mise en œuvre des différentes notions acquises (gestion des fichiers) dans le cadre du projet de gestion des notes.
- Reprendre les travaux réalisés dans le cadre de l'atelier 3 pour la gestion des notes et l'améliorer en permettant la sauvegarde des notes des étudiants sur disque dur dans un format donné (CSV par exemple).
- Inversement permettre de lire les notes des étudiants qui sont stockées dans un fichier et les transférer dans des tableaux au sein de la RAM

#### Remarque

Le programme de gestion des notes va évoluer en y intégrant à chaque nouveau chapitre les différentes notions étudiées.