Mots-clés : fichier, dossier, SGF, racine, nom absolu, nom relatif, extension, format, propriétaire, interopérabilité… traitement des fichiers en PHP…analyse de logs

Un fichier est une suite d'informations stockées sur un support physique.
-citer des technologies matérielles (ou virtuelles « cloud ») de stockage et leurs caractéristiques

Un dossier, vu du SGF (le Système de Gestion de Fichiers ou File System) , est un fichier particulier qui joue le rôle de conteneur de fichiers.
Le dossier nommé « . » (point) représente le dossier courant.
Le dossier nommé « .. » (point point) représente le dossier parent.
-citer un SGF et ses caractéristiques

Le chemin d'accès d'un fichier est la liste des dossiers à parcourir pour atteindre ce fichier.
Si cette liste commence à la racine du stockage, c'est un **chemin absolu** sinon un **chemin relatif**.
La désignation d'un fichier est unique en fournissant son chemin d'accès et son nom.

-selon les OS et leur SGF, la racine et les séparateurs de dossiers diffèrent.

	Linux ou Windows ?	Absolu ou relatif ?
C:\gustave\Exercices\D1\Fiche.odt		
..\Exercices\D1\Fiche.odt		
/home/gustave/Exercices/D1/Fiche.odt		

L'extension est le suffixe du nom de fichier (txt, docx, jpg, php…) . Elle informe sur la nature du fichier et permet au système d'exploitation d'associer un logiciel pex extension .doc => MS Word

Interopérabilité : pour qu'un logiciel puisse exploiter les fichiers produits par un autre logiciel, il doit connaître les spécifications du format des fichiers.
Un format est dit ouvert si ses spécifications sont publiées et accessibles de tous (norme/standard)
Dans le cas contraire, on parle de format propriétaire (ou fermé).
//Lien fort entre interopérabilité et format ouvert-propriétaire

	Ouvert ?	Propriétaire ?
compta.xls //idem ppt et doc		
medecins.json		
monRapportDeStage.odt //Open Office Open Document		
portefolio.html		
archive.rar		
plage.jpg //Joint Photographic Experts Group		
plage.gif //Graphic Interchange Format		
fluteEnchantee.wma		
guitare.mp3 //Moving Pictures Expert Group		

On distingue notamment deux catégories de fichiers :
•Les fichiers textes : les informations sont stockées sous forme de caractères lisibles par tout éditeur de texte.
•Les fichiers binaires : les informations ne peuvent être exploitées qu'avec le logiciel adéquat. Un binaire ouvert avec un éditeur de texte est illisible!

Exemples de fichiers textes :
-Un classeur au format ODF est constitué de fichiers XML (Extensible Markup Language), du texte enrichi à l'aide de balises textuelles
-Un fichier au format CSV contient les données séparées en colonnes (pex la virgule comme séparateur de colonnes)
-Une page web au format HTML
-Un fichier de journalisation ou log (pour tracer les évènements), pex au format CSV

Exemples de fichiers binaires :
-Un document au format Microsoft Word 97/2000/XP.
-Une photo au format d'image JPEG.
-Un fichier exécutable d'application.

Partie 1 : Traitement des fichiers en PHP

PHP fournit des fonctions pour lire, créer, modifier des fichiers.

Fichier texte exemple : films.txt

```
Alien 1979 Scott Ridley 1943
Vertigo 1958 Hitchcock Alfred 1899
Psychose 1960 Hitchcock Alfred 1899
Kagemusha 1980 Kurosawa Akira 1910
Volte-face 1997 Woo John 1946
Titanic 1997 Cameron James 1954
Sacrifice 1986 Tarkovski Andrei 1932
Intouchable 2011 Toledano Eric 1970
```

//8 lignes de texte, le tout éditable avec un bloc-notes ou IDE
//on peut créer ce fichier films.txt par un simple copier-coller
//pour chaque ligne, les champs sont de longueur variable et séparés par un espace

Mode d'ouverture : fopen()

L'ouverture d'un fichier est réalisée par la fonction fopen(), en indiquant le mode d'ouverture.

R	Read	Ouvre en lecture seule, place le pointeur en début de fichier
A	Append	Ouvre en lecture seule, place le pointeur en fin de fichier. Si le fichier n'existe pas, PHP tente de le créer.
W	Write	Ouvre en écriture, supprime tout le contenu et place le pointeur en début de fichier. Si le fichier n'existe pas, PHP tente de le créer.
X		Crée et ouvre en écriture, place le pointeur en début de fichier. Si le fichier existe déjà, fopen() retourne une erreur
C		Ouvre en écriture, place le pointeur en début de fichier (sans supprimer le contenu comme w). Si le fichier n'existe pas, PHP tente de le créer.

Ouverture d'un fichier et création d'un pointeur

fopen() ouvre un fichier et retourne un pointeur vers celui-ci (du type PHP resource).
Ce pointeur doit alors être passé en paramètre des fonctions de manipulation de fichiers.

Fermeture d'un fichier : fclose()

Fclose() ferme un fichier préalablement ouvert.

```
<?php
$fp= fopen('films.txt','r') ;        // PHP cherche par défaut dans le dossier où se trouve ce script PHP
fclose($fp);                         // fermer le fichier désigné par le pointeur $fp
//ici aucun traitement sur le contenu du fichier
?>
```

Fin de fichier : feof()

Feof() (file : end of file) retourne TRUE si le pointeur est en fin de fichier, FALSE sinon.
Indispensable lorsqu'un script traite un fichier ligne par ligne par exemple.

Lecture complète du fichier : fread(), filesize()

```
$fp= fopen('films.txt','r');
echo fread($fp, 10);                          //affiche        Alien 1979
fclose($fp);
```

fread() lit le nombre d'octets (2ème paramètre) demandé, ici 10 caractères
Ce script ne lit pas tout le fichier, juste les 10 premiers caractères (Alien 1979)

```
$fp= fopen('films.txt','r');
while (!feof($fp)){
   echo fread($fp, 10);
}                                             //affiche tout le fichier
fclose($fp);
```

La boucle While continue ici tant que le pointeur n'a pas atteint la fin du fichier.
A chaque boucle, fread() lit 10 octets qui sont affichés (echo).
Ce script lit tout le fichier.

```
$fp= fopen('films.txt','rb');
echo fread($fp, filesize('films.txt'));       //affiche tout le fichier
fclose($fp);
```

fread() récupère tout le fichier puisque son 2ème paramètre correspond à la taille du fichier.

Lecture par ligne courante :fgets()

/* squelette typique */

```
$fp = fopen('films.txt', 'r');
while (!feof($fp)) {
   echo fgets($fp, 1024);
}                                          //affiche tout le fichier, ligne par ligne de 1024 octets
fclose($fp);
```

```
$fp = fopen('films.txt', 'r');
while (false !== ($chaine = fgets($fp))) {          //à commenter ?
   echo $chaine;
}                                                   //affiche tout le fichier
fclose($fp);
```

Lecture d'un fichier CSV : fgetcsv()

Fgetcsv() , proche de fgets(), permet de rechercher et extraire les champs CSV à l'intérieur d'une chaine lue.
Elle retourne un tableau unidimensionnel à indice numérique contenant les champs .
Une ligne vide retourne un tableau avec la valeur NULL.

Lecture de la première ligne du fichier : Alien 1979 Scott Ridley 1943

```
$fp = fopen('films.txt', 'r');
$tableau=  fgetcsv($fp, 1024, " ");
fclose($fp);
var_dump($tableau);
```

```
array(5) {
  [0] =>  string(5) "Alien"
  [1] =>  string(4) "1979"
  [2] =>  string(5) "Scott"
  [3] =>  string(6) "Ridley"
  [4] =>  string(4) "1943"
}
```

Le séparateur est ici l'espace (3ème paramètre) //sinon virgule, point-virgule, etc.

/* squelette typique*/ lecture de tout le fichier

```
$ligne = 1; // compteur de lignes
$fic = fopen("films.txt", "r");
while ($tab = fgetcsv($fic, 1024, " ")) {
   $champs = count($tab);
   echo "Les " . $champs . " champs de la ligne " . $ligne . " sont :\n";
   $ligne++;
   for ($i = 0; $i < $champs; $i++) {
      echo $tab[$i] . "\n";
   }
}
```

Ecriture d'un fichier : fwrite()

```
$fp = fopen('bacasable.txt', 'w');
fwrite($fp, 'abc');
fwrite($fp, 'def');
fclose($fp);                                    //le fichier contient abcdef
```

-on relance ce script : que contient le fichier ?
-on relance ce script en modifiant le mode d'ouverture 'a' : que contient le fichier ?

Ecriture d'un fichier CSV: fputcsv()

```
$tableau=array(                 //les champs
   'colonne1',
   'colonne2',
   'colonne3'
);
$fp = fopen('bacasable.txt', 'w');
fputcsv($fp, $tableau,';',"'");     //separateur ; (pointvirgule)     delimiteur de champ ' (apostrophe)
fclose($fp);
```

bacASable.txt contient alors la ligne colonne1;colonne2;colonne3

Et si le fichier n'existe pas ? die() ou exit()

Die	exit : Strictement équivalent
<?php $filename = '/path/to/data-file'; $file = fopen($filename, 'r') or die("unable to open file ($filename)"); ?>	<?php $filename = '/path/to/data-file'; $file = fopen($filename, 'r') or exit("unable to open file ($filename)"); ?>

Die et Exit arrête le script en cours d'exécution en affichant le message « unable to open file.. »
//voir si traitement possible avec une Exception ?

Les fonctions d'accès rapide (préfixe file…)

Elles ne nécessitent pas l'ouverture préalable-la fermeture du fichier et attendent en paramètre le chemin du fichier (et non pas un pointeur comme précédemment)

file	Retourner un tableau avec toutes les lignes du fichier
file_get_contents	Retourner une chaine avec toutes les lignes du fichier
readfile	Afficher tout le fichier
file_put_contents	Ecrire dans un fichier
filesize	Taille du fichier en octets
file_exists	Booleen
is_writable	Booleen

Les commandes de « ménage » des fichiers et dossiers

copy rename chmod mkdir rmdir getcwd chdir	chmod('/var/www/test.html', 0755) ;

Ajouter un film à la fin du fichier « films.txt »

```
<?php
$unFilm="Midnight in Paris 2011 Allen Woody 1935 \n";
file_put_contents('films.txt',$unFilm,FILE_APPEND);
?>
```

Exercices

Cloner le dépôt https://github.com/herveleguern/tp10logs.git

Exercice1
Tester le code permettant d'ajouter un film au fichier films.txt
Ajouter le(s) film(s) de votre choix

Exercice2
Parcourir le fichier films.txt et afficher uniquement les titres des films.

Exercice3 : fusionner des fichiers
A partir des 2 fichiers fournis : comedies.txt et actions.txt
produire un nouveau fichier video.txt qui contient tous les films (comédies et actions)

Partie 2 : les expressions régulières (regex)

Les expressions régulières (ou expressions rationnelles ou regex) fournissent des fonctions avancées de **Recherche/Extraction/Remplacement** dans des chaines de caractères.
Les regex sont exploitables dans tous les langages de programmation, pour :

- Rechercher la présence d'une sous chaine
  exemple : repérer les mois « en R » (janvier, février…), les mois « sans R » (mai, aout…)
- Vérifier la conformité d'une chaine – contrôle de saisie, pour valider un formulaire
  exemple : format de téléphone , d'adresse mail, complexité d'un mot de passe etc.
- Remplacer-Transformer des chaines
  exemple : convertir des URL textuelles en liens hypertextes HTML
  www .bonplan.com devient cliquez sur bonplan
- Rechercher et extraire des données d'une chaine
  exemple : extraire les IP ou les noms de fichiers d'un fichier de log

Documentation de la fonction preg_match //extrait étude de cas SIO

La fonction *PHP preg_match($modele, $chaine): boolean*
recherche un modèle (pattern) bien précis au sein d'une chaîne de caractères.

Exemples de pattern

[abc]	un simple caractère : a, b ou c
[a-z]	un caractère appartenant à l'ensemble a-z
[A-Z]	un caractère appartenant à l'ensemble A-Z
[0-9]	un chiffre appartenant à l'ensemble 0-9
\W	n'importe quel caractère spécial

Le pattern en PHP est encadré par le délimiteur /

fonction verifPassword //extrait étude de cas SIO

```
    function verifPassword($mdp): bool
    {
1       $points_total = 6;
2       $longueur = strlen($mdp); // nombre de caractères de $mdp
3       $points_long=0;     // points pour la longueur, soit 0, soit 1
4       $points_comp=0;     // points pour la complexité
5       if ($longueur >=8) { $points_long=1; }

6       if(preg_match("/[a-z]/", $mdp)) { $points_comp=$points_comp+1; }

7       if(preg_match("/[A-Z]/", $mdp)) { $points_comp=$points_comp+2; }

8       if(preg_match("/[0-9]/", $mdp)) { $points_comp=$points_comp+3; }

9       $resultat = $points_long * $points_comp;
10       return ($points_total == $resultat);
    }
```

Exemple 1 : Variante pour contrôler la complexité d'un mot de passe

```
$motDePasse = "secREt12345";
$contientMajuscule = preg_match('/[A-Z]/', $motDePasse);
$contientMinuscule = preg_match('/[a-z]/', $motDePasse);
$contientChiffre = preg_match('/[0-9]/', $motDePasse);
$contientSpecial = preg_match('/[-_.!#]/', $motDePasse);
//$motDePasse est -il un mot de passe de complexité et longueur suffisantes ?
if (!$contientMajuscule || !$contientMinuscule || !$contientChiffre
      || !$contientSpecial || strlen($motDePasse) < 12)
   echo "mot de passe NON valide";
else
   echo "mot de passe valide";
```

Exemple 2 : Que « matchent » les regex suivantes ?

```
$chaine='a99a';
var_dump(preg_match('/^[0-9]*$/', $chaine));    //ne matche pas, FALSE(0)

$chaine='06-31-32-33-34';
var_dump(preg_match('/^0[1-68]([-. ]?[0-9]{2}){4}$/', $chaine));    //matche, TRUE (1)

$chaine='abcd';
var_dump(preg_match('/^[a-zA-Z0-9]{5,}$/', $chaine));    //ne matche pas, FALSE(0)

$chaine='172.31.1.2';
var_dump(preg_match('/ ^84\.254(\.[0-9]{1,3}){2}$/', $chaine));    //ne matche pas, FALSE(0)
$chaine='84.254.001.207';
var_dump(preg_match('/ ^84\.254(\.[0-9]{1,3}){2}$/', $chaine));    //matche, TRUE(1)
```

Exemple 3 : Extraire des sous-chaines

En utilisant la regex adaptée, extraire les url de la chaine suivante

```
$ch = "La publication  ftp://monserveur  et les moteurs de recherche http://www.google.fr et le
miroir ftp://mout.local sont aujourd'hui les plus utilisés.";

preg_match('/http:\/\/[a-zA-Z0-9._-]*/', $ch,$matches);
var_dump($matches);//ne trouve que Google

preg_match_all('/ftp:\/\/[a-zA-Z0-9._-]*|http:\/\/[a-zA-Z0-9._-]*/', $ch, $matches);
var_dump($matches);//trouve tout
```

Exemple 4 : Extraire des sous-chaines et les remplacer

```
$chaine1 = "Les moteurs de recherche http://www.google.fr et http://www.bing.com sont
aujourd'hui les plus utilisés.";
$chaine2 = preg_replace('/(http:\/\/[a-zA-Z0-9._-]*)/', '<a href="$1">$1</a>', $chaine1);
echo "AVANT : " . $chaine1; echo "APRES : " . $chaine2;
```

Les URL sont remplacées par des liens hypertextes vers les URL, et interprétées comme tels par un navigateur web (lien cliquable)

Application 1 : analyser les logs Apache

Access.log est le journal du serveur http Apache . C'est un fichier texte, constitué de lignes.
Chaque ligne commence par l'IP du client qui a interrogé le serveur HTTP.
Un extrait de 2 lignes :

```
192.168.1.9 - - [20/Nov/2017:09:10:34 +0100] "GET /glpi/front/computer.php?itemt
ype=Computer&sort=60&order=DESC&start=0&criteria%5B0%5D%5Bfield%5D=view&criteria
%5B0%5D%5Blink%5D=contains&criteria%5B0%5D%5Bvalue%5D=& HTTP/1.1" 200 11992 "htt
p://192.168.1.49/glpi/front/computer.php?reset=reset" "Mozilla/5.0 (Windows NT 6
.3; Win64; x64; rv:57.0) Gecko/20100101 Firefox/57.0"
192.168.1.9 - - [20/Nov/2017:09:27:56 +0100] "POST /phpmyadmin/index.php HTTP/1.
1" 200 2499 "-" "Mozilla/5.0 (Windows NT 6.3; Win64; x64; rv:57.0) Gecko/2010010
1 Firefox/57.0"
```

NB Apache permet de configurer le contenu du journal //quels champs, quel format CSV etc.

Vous disposez du dossier http contenant :
access.log
exploreHttp01.php

Tester exploreHttp01.php //voir les limites de la fonction explode et alternative regex

Coder les scripts PHP permettant
Exercice 4 : de compter le nombre d'accès par IP
Exercice 5 : d'extraire les URL des requêtes ayant menée à une erreur http 404
Exercice 6 : de repérer les différentes commandes http exploitées (GET, POST, etc.)

Application 2 : analyser les logs SSH

Vous disposez du dossier ssh contenant :
ssh.log

ssh > ssh.log

```
1  Aug  1 18:27:45 knight sshd[20325]: Illegal user test from 218.49.183.17
2  Aug  1 18:27:46 knight sshd[20325]: Failed password for illegal user test from 218.49.183.17
3  Aug  1 18:27:46 knight sshd[20325]: error: Could not get shadow information for NOUSER
4  Aug  1 18:27:48 knight sshd[20327]: Illegal user guest from 218.49.183.17
5  Aug  1 18:27:49 knight sshd[20327]: Failed password for illegal user guest from 218.49.183.17
6  Aug  1 18:27:49 knight sshd[20327]: error: Could not get shadow information for NOUSER
7  Aug  1 18:27:52 knight sshd[20329]: Failed password for admin from 218.49.183.17 port 49266 s
```

Coder les scripts PHP permettant
Exercice 7 : de compter le nombre d'accès « non autorisés » par IP
//retenir les lignes ou apparait le terme « failed password for root », et extraire l'adresse IP

Exercice 8 : de produire la liste des IP à bannir
//parcourir le tableau précédent $statIP
//extraire les IP dont le nombre est > à 3 (3 échecs d'authentification SSH)