

TP PHP n°2 : accès aux données

© 2013-2016 <tvaira@free.fr>

Sommaire

Accès aux données	2
Manipulation de fichiers	2
Gestion d'une base de données	3
Annexe n°1 : Les fonctions de manipulation de fichiers	4
Annexe n°2 : les mesures dans l'industrie	5
Annexe n°3 : moyenne et médiane	6
Annexe n°4 : Les fonctions d'accès aux bases de données	6

Les objectifs de ce tp sont d'être capable de découvrir les bases du langage PHP pour réaliser des scripts serveurs pour un site web en manipulant des fichiers ou une base de données.



Il existe de très nombreux sites dédiés au PHP! Il faut au moins l'accès au manuel (notamment pour les fonctions) : www.php.net/manual/fr/index.php

Accès aux données

Manipulation de fichiers

L'objectif de cette séquence est de s'initier à la manipulation des fichiers à partir du langage PHP.

Dans le cadre d'un développement d'un site web spécialisé dans la mesure industrielle, vous participez à la réalisation d'un script en langage PHP.

L'acquisition de mesures de température (capteur pt100) va mettre dans un fichier texte une valeur toutes les minutes. Ces séries de mesures peuvent comporter des mesures incohérentes (lire l'annexe 1). Après traitement, on ne conservera que la médiane (et non la moyenne) de ces séries de mesures. La valeur médiane (lire l'annexe 2) est la valeur qui se trouve au milieu d'un ensemble de nombres triés. Si cet ensemble contient un nombre pair de nombres, la médiane sera alors la moyenne des deux nombres du milieu.

Contraintes :

- Deux fichiers de mesures sont disponibles pour les tests. Le script doit fonctionner correctement pour ces deux fichiers (qui contiennent un nombre pair et impair de mesures).
- Le calcul de la médiane se fera dans une fonction `CalculerMediane()`. Les mesures doivent être préalablement triées (la fonction de tri est fournie).
- Les fichiers `mediane_<fichier_de_mesures>.txt` seront créés dans un répertoire `fichiers_mediane` à la racine du script. Ce répertoire doit avoir les droits d'écriture pour les autres (*other*) afin que le script puisse écrire dedans.

```
<?php
// cf. http://fr.php.net/manual/fr/function.include.php
include("tri.inc.php");

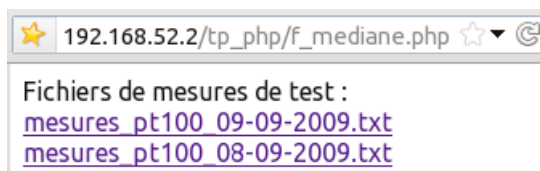
// TODO
// ...

?>
```

f_mediane_todo.php

Question 1. Écrire le script `f_mediane.php` qui permet de calculer et d'écrire dans un fichier `mediane_<fichier_de_mesures>.txt` la médiane d'une série de mesures lues dans un fichier passé en argument dans l'*url* du script.

L'appel du script sera du type : `f_mediane.php?mesure=<fichier_de_mesures>`

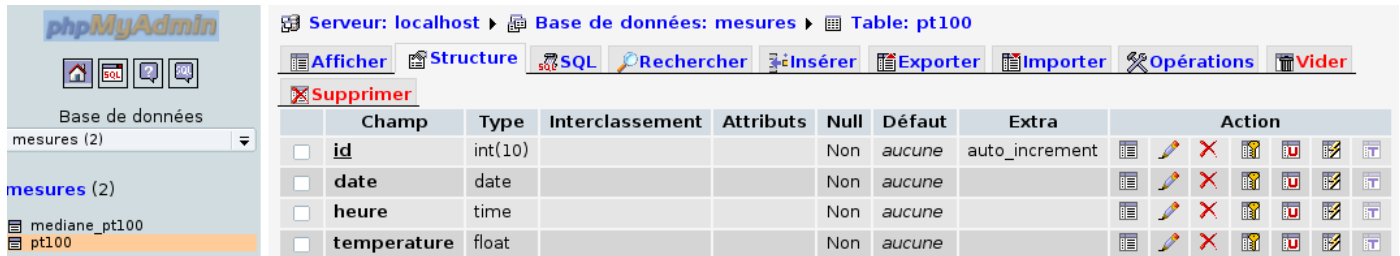


Gestion d'une base de données

L'objectif de cette séquence est de s'initier à l'utilisation des bases de données en langage PHP.

On utilise le même contexte que la séquence précédente mais les mesures sont maintenant stockées dans une base de données **mesures**. Cette base de données contient deux tables :

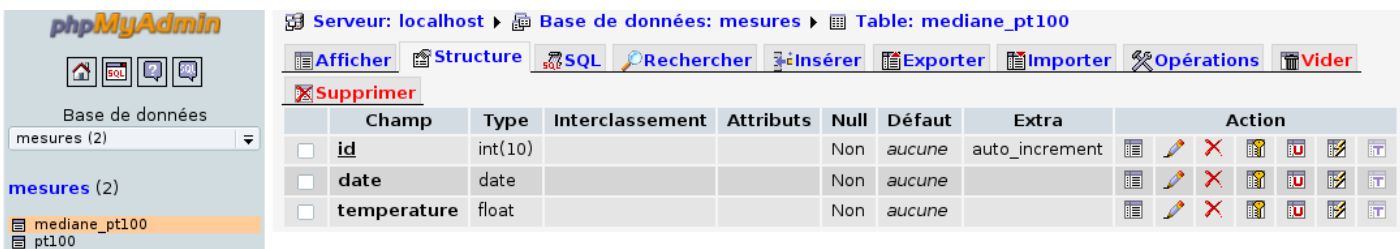
- La table **pt100** contient les mesures de températures horodatées :



Structure de la table **pt100** :

Champ	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
<input type="checkbox"/> id	int(10)			Non	aucune	auto_increment	[icône]
<input type="checkbox"/> date	date			Non	aucune		[icône]
<input type="checkbox"/> heure	time			Non	aucune		[icône]
<input type="checkbox"/> temperature	float			Non	aucune		[icône]

- La table **mediane_pt100** permet de conserver la valeur mediane (une fois calculée) d'une série de mesures horodatées :



Structure de la table **mediane_pt100** :

Champ	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
<input type="checkbox"/> id	int(10)			Non	aucune	auto_increment	[icône]
<input type="checkbox"/> date	date			Non	aucune		[icône]
<input type="checkbox"/> temperature	float			Non	aucune		[icône]



Le fichier `mesures.sql` permettant de créer cette base de données (avec des mesures déjà effectuées) sur un serveur MySQL vous est fourni.

Contraintes :

- Deux séries de mesures sont disponibles dans la table **pt100** en fonction de la *date*. Le script doit fonctionner correctement pour ces deux séries (nombre pair et impair de mesures). Le champ *heure* de la table **pt100** n'est pas utilisé dans cette séquence.
- Le calcul de la mediane se fera dans une fonction `CalculerMediane()`. Les mesures doivent être préalablement triées (la fonction de tri est fournie).
- La table **mediane_pt100** n'accepte qu'une seule mediane par *date* de mesures. On se limitera à une insertion unique mais une gestion plus fine devrait être faite.

```
<?php
// cf. http://fr.php.net/manual/fr/function.include.php
include("tri.inc.php");

// TODO
// ...

?>
```

bd_mediane_todo.php

Question 2. Écrire le script `bd_mediane.php` qui permet de calculer et d'écrire dans une table MySQL **mediane_pt100** la médiane d'une série de mesures lues dans une table MySQL **pt100** dont la date (format AAAA-MM-JJ) est passée par l'envoi d'un formulaire.

Mesures de test dans la base de données :

2009-09-09 ▼

Envoyer

Annexe n°1 : Les fonctions de manipulation de fichiers

PHP fournit de nombreuses fonctions pour manipuler des fichiers d'un système de fichiers : fr.php.net/manual/fr/book.filesystem.php.

Les appels de base pour la gestion des E/S fichiers sont : `fopen`, `fread`, `fwrite`, `fclose`, ...

L'association entre une ressource nommée et un nom physique s'effectue à l'ouverture du fichier. La ressource nommée représentant le flux est en fait un pointeur de fichier. Le nom physique du fichier est une chaîne de caractères contenant son nom et éventuellement son chemin dans l'arborescence du système de fichiers géré par l'OS. L'ouverture d'un fichier se fait suivant un mode qui spécifie le type d'accès désiré au flux (lecture seule, écriture seule, lecture/écriture, ...).

```
<?php
$filename = "datas.txt";
// Vérifier que le fichier existe
if (@file_exists($filename))
{
    // Ouvrir le fichier en lecture seule
    $fichier = @fopen($filename, "r");
    //Remarque : l'@ placé devant une fonction bloquera les messages d'erreurs (nuisibles)
    envoyés au navigateur client

    // Pas d'erreur à l'ouverture ?
    if($fichier != FALSE)
    {
        // Lire 10 octets dans le fichier
        $datas = fread($fichier, 10);
        echo "Dix premiers octets du fichier $filename : " . $datas . "<br />";
        // Remarque : la position du pointeur de fichier s'est déplacé de 10 octets

        // Lire (encore) 10 octets dans le fichier
        $datas = fread($fichier, 10);
        echo "Dix octets suivants du fichier $filename : " . $datas . "<br />";
        // Remarque : la fonction fseek() permettrait de modifier la position du pointeur de
        fichier

        // Fermer le fichier
        fclose($fichier);
    }
    else die("Erreur : ouverture impossible du fichier $filename !<br />");
}
else die("Erreur : le fichier $filename n'existe pas !<br />");
// Remarque : la fonction die() ou exit() affiche un message et termine le script courant
?>
```

Lire des octets dans un fichier



Les fonctions de lecture `fgetc()`, `fgetcsv()`, `fgets()`, `fscanf()` et `fgetss()` peuvent être intéressantes pour réaliser certains traitements spécifiques.

Certaines fonctions de lecture n'ont pas besoin de réaliser d'ouverture préalable du fichier à traiter. C'est le cas de :

- `file()` qui lit le fichier et renvoie le résultat dans un tableau
- `file_get_contents()` qui lit tout un fichier dans une chaîne

Pour écrire dans un fichier, on pourra utiliser au choix les fonctions `fwrite()`, `fputs()` ou même `file_put_contents()` et `fputcsv()` dans certains cas précis.

Annexe n°2 : les mesures dans l'industrie

Dans le cas des mesures dans l'industrie, on considère trois sources d'erreur (source wikipedia) :

- la précision de la mesure ou l'incertitude ;
- la dispersion statistique ;
- l'erreur systématique.

L'erreur totale étant la somme des trois sources d'erreurs.

Si l'on fait la comparaison avec des flèches que l'on tire sur une cible :

- la précision de mesure désigne la taille de la pointe de la flèche ;
- la dispersion statistique désigne le fait que les flèches sont proches les unes des autres, ou bien au contraire éparpillées sur la cible ;
- l'erreur systématique indique si les flèches visaient bien le centre, ou bien un autre point de la cible.

Pour la dispersion statistique, on estime que si l'on mesure plusieurs fois le même phénomène avec un appareil suffisamment précis, on obtiendra chaque fois un résultat différent. Ceci est dû à des phénomènes perturbateurs ou, pour les mesures extrêmement précises, à la nature aléatoire du phénomène.

Parmi les phénomènes perturbateurs, on peut dénombrer :

- l'erreur d'échantillonnage : c'est lorsque l'on prélève un échantillon qui n'est pas représentatif de ce que l'on veut mesurer ; le résultat dépend alors de la manière dont on choisit l'échantillon ;
- l'erreur de préparation : l'échantillon s'altère pendant le transport, le stockage ou la manipulation (pollution, dégradation, transformation physique ou chimique) ;
- la stabilité de l'appareil : celui-ci peut être sensible aux variations de température, de tension d'alimentation électrique, aux vibrations, aux perturbations électromagnétiques des appareils environnants ou bien présenter un défaut de conception ou une usure (bruit de fond électronique, pièce instable ...).

Le calcul d'erreur, ou calcul d'incertitudes est un ensemble de techniques permettant d'estimer l'erreur faite sur un résultat numérique, à partir des incertitudes ou des erreurs faites sur les mesures qui ont conduit à ce résultat. L'erreur de mesure détermine la sensibilité (capacité à sélectionner les bons « candidats ») et la sélectivité (capacité à éliminer les mauvais « candidats ») d'une méthode.

Annexe n°3 : moyenne et médiane

L'utilisation de la médiane à la place de la moyenne est fréquent pour les mesures dans l'industrie.

Exemple, soit deux listes de mesures provenant d'un capteur sur une période de 1mn30s :

- L1 : 35,53°C, 35,23°C, 35,10°C, 35,02°C, 34,45°C
- L2 : 35,53°C, 35,23°C, 35,10°C, 34,45°C, 12,22°C

Dans la série L2, la mesure incohérente (12,22°C) serait pris en compte dans la moyenne et fausserait donc le résultat obtenu (la moyenne sans cette valeur est de 35,07°C contre 30,50°C si on en tient compte) :

Liste	L1	L2
Moyenne	35,066°C	30,506°C
Médiane	35,10°C	35,10°C

Ici, l'utilisation de la médiane comme technique de sélectivité permet d'atténuer ce type de problème.

L'utilisation de la valeur médiane est donc préférable à la valeur moyenne. Cependant, son utilisation implique le tri des données au préalable.

Annexe n°4 : Les fonctions d'accès aux bases de données

Parmi les nombreux atouts du langage PHP, un des plus connus est son interfaçage avec la majorité des bases de données du marché. Parmi les plus connues, on peut citer : **MySQL**, SQLite, PostgreSQL, Oracle, Ingres, Interbase, Informix, Microsoft SQL Server, mSQL, Sybase, FrontBase, dBase, etc ...

La base de donnée la plus utilisée avec PHP est sans aucun doute : MySQL, un SGDBR (Système de Gestion de Base de Données Relationnelle) GPL implémentant le langage de requête SQL (*Structured Query Language*).



Il existe un outil libre et gratuit développé en PHP par la communauté des programmeurs libres : php-MyAdmin, qui permet l'administration aisée des bases de données MySQL avec PHP.

Avec MySQL vous pouvez créer plusieurs bases de données sur un serveur. Une base est composée de tables contenant des enregistrements.

PHP offre 3 APIs différentes pour se connecter à MySQL : les extensions **mysql**, **mysqli** et **PDO**. PDO fournit une interface d'abstraction à l'accès de données, ce qui signifie que vous utilisez les mêmes fonctions pour exécuter des requêtes ou récupérer les données quelque soit la base de données utilisée.



Il est recommandé d'utiliser soit l'extension **mysqli**, soit l'extension **PDO_MySQL** car l'ancienne extension **mysql** est obsolète depuis PHP 5.5 et sera supprimée dans un futur proche.

PHP fournit un grand choix de fonctions permettant de manipuler une base de données MySQL. Toutefois, parmi celles-ci quatre fonctions sont essentielles :

- La fonction de connexion au serveur (**mysqli_connect** ou **mysqli_real_connect**)
- La fonction de choix de la base de données (**mysqli_select_db**)
- La fonction de requête (**mysqli_query**)
- La fonction de déconnexion (**mysqli_close**)

```

<?php
// mysqli : http://fr.php.net/manual/fr/book.mysql.php
// Style fonctionnel
if (!extension_loaded('mysqli'))
    die("L'extension mysqli n'est pas présente !");

$link = mysqli_connect('localhost', 'root', 'password', 'test');
if (!$link)
    die('Echec de connexion au serveur de base de données : ' . mysqli_connect_error() . '(' .
        mysqli_connect_errno() . ') ');

echo 'Fonctions mysqli : succès ... ' . mysqli_get_host_info($link) . " - MySQL server
    version : " . mysqli_get_server_info($link) . "<br />\n";

mysqli_close($link);
?>

```

Les fonctions mysqli

Avec les autres APIs :

```

<?php
// mysqli : http://fr.php.net/manual/fr/book.mysql.php
// Style POO
if (!class_exists('mysqli'))
    die("La classe mysqli n'est pas présente !");
// ou :
if(!in_array("mysqli", get_declared_classes()))
    die("La classe mysqli n'est pas présente !");

$mysqli = new mysqli("localhost", "root", "password", "test");
if ($mysqli->connect_error)
    die('Echec de connexion au serveur de base de données : ' . $mysqli->connect_error . '(' .
        $mysqli->connect_errno . ') ');

echo 'Classe mysqli : succès ... ' . $mysqli->host_info . " - MySQL server version : " . $
    mysqli->server_info . "<br />\n";

$mysqli->close();
?>

```

La classe mysqli

```

<?php
// MySQL Functions : http://www.php.net/manual/en/ref.mysql.php
// This extension is deprecated as of PHP 5.5.0, and will be removed in the future.
if (!extension_loaded('mysql'))
    die("L'extension mysql n'est pas présente !");

if($link = mysql_connect("localhost", "root", "password"))
{
    $id_db = mysql_select_db("test");
    if(!$id_db) die('Echec de connexion à la base : ' . mysql_error() . '(' . mysql_errno() .
        ') ');
}

```

```

}
else die('Echec de connexion au serveur de base de données : ' . mysql_error() . '(' .
    mysql_errno() . ') ');

echo 'Fonctions mysql : succès ... ' . mysql_get_host_info($link) . " - MySQL server version
    : " . mysql_get_server_info() . "<br />\n";

mysql_close($link);
?>

```

Les fonctions mysql (obsolète depuis PHP 5.5)

```

<?php
// PDO : http://www.php.net/manual/en/book.pdo.php
// pdo_mysql : http://www.php.net/manual/en/ref.pdo-mysql.php

if(!in_array("PDO", get_loaded_extensions()))
    die("L'extension PDO n'est pas présente !<br><br>");

if(!in_array("pdo_mysql", get_loaded_extensions()))
    die("L'extension pdo_mysql n'est pas présente !<br><br>");

$pdo_db = new PDO('mysql:host=localhost;dbname=test', 'root', 'password') or die("Echec de
    la création de l'instance PDO !");

echo "Classe PDO : succès ... <br />\n";

unset($pdo_db);
?>

```

La classe PDO

L'exécution d'une requête `SELECT` avec `mysqli_query()` retournera un objet résultat de type `mysqli_result` (ou `TRUE` pour les autres types de requêtes).

Les fonctions de traitements de résultat d'une requête sont au choix :

- `mysqli_fetch_row()` : récupère une ligne de résultat sous forme de tableau indexé
- `mysqli_fetch_array()` : retourne une ligne de résultat sous la forme d'un tableau associatif, d'un tableau indexé, ou les deux
- `mysqli_fetch_assoc()` : récupère une ligne de résultat sous forme de tableau associatif
- `mysqli_fetch_object()` : retourne la ligne courante d'un jeu de résultat sous forme d'objet
- et `mysqli_free_result()` : libère la mémoire associée à un résultat

```

<?php
$link = mysqli_connect('localhost', 'root', 'password', 'mysql');
if (!$link)
    die('Echec de connexion au serveur de base de données : ' . mysqli_connect_error() . '(' .
        mysqli_connect_errno() . ') ');

if ($result = mysqli_query($link, "SELECT Host, User FROM `user` ORDER BY User DESC LIMIT 0
    , 30"))
{
    printf("Fonctions mysqli : la requête a retourné %d enregistrement(s).<br />\n",
        mysqli_num_rows($result));
    /* Tableau indexé */
}

```



```

// $row = mysqli_fetch_array($result, MYSQLI_NUM);
// printf ("%s - %s<br />\n", $row[0], $row[1]);
// ou tous les résultats de la requête
// while(list($host, $user) = mysqli_fetch_row($result))
// {
//   echo "$host - $user<br />";
// }

/* Tableau associatif */
// $row = mysqli_fetch_array($result, MYSQLI_ASSOC);
// printf ("%s - %s<br />\n", $row["Host"], $row["User"]);
// ou tous les résultats de la requête :
while($row = mysqli_fetch_array($result))
{
    $host = $row["Host"];
    $user = $row["User"];
    echo "$host - $user<br />";
}

/* Libération des résultats */
mysqli_free_result($result);
}

mysqli_close($link);
?>

```

Traitement des résultats avec les fonctions mysqli

Avec les autres APIs :

```

<?php
$mysqli = new mysqli('localhost', 'root', 'password', 'mysql');
if ($mysqli->connect_error)
    die('Echec de connexion au serveur de base de données : ' . $mysqli->connect_error . '(' .
        $mysqli->connect_errno . ') ');

if ($result = $mysqli->query("SELECT Host, User FROM `user` ORDER BY User DESC LIMIT 0 , 30"
))
{
    printf("Classe mysqli : la requête a retourné %d enregistrement(s).<br />\n", $result->
        num_rows);
    /* Tableau indexé */
    // $row = $result->fetch_array(MYSQLI_NUM);
    // printf ("%s - %s<br />\n", $row[0], $row[1]);
    // ou tous les résultats de la requête :
    // while(list($host, $user) = $result->fetch_array())
    // {
    //   echo "$host - $user<br />";
    // }

    /* Tableau associatif */
    // $row = $result->fetch_array(MYSQLI_ASSOC);
    // printf ("%s - %s<br />\n", $row["User"], $row["Host"]);
    // ou tous les résultats de la requête :
    while($row = $result->fetch_array())

```

```

{
    $host = $row["Host"];
    $user = $row["User"];
    echo "$host - $user<br />";
}

/* Libération des résultats */
$result->free();
}

$mysqli->close();
?>

```

Traitement des résultats avec la classe mysqli

```

<?php
if($link = mysql_connect("localhost", "root", "password"))
{
    $id_db = mysql_select_db("mysql", $link);
    if(!$id_db) die('Echec de connexion à la base : ' . mysql_error() . '(' . mysql_errno() .
        ') ');
}
else die('Echec de connexion au serveur de base de données : ' . mysql_error() . '(' .
    mysql_errno() . ') ');

if ($result = mysql_query("SELECT Host, User FROM `user` ORDER BY User DESC LIMIT 0 , 30", $
    link))
{
    printf("Fonctions mysql : la requête a retourné %d enregistrement(s).<br />\n",
        mysql_num_rows($result));
    /* Tableau indexé */
    //$row = mysql_fetch_array($result, MYSQLI_NUM);
    //printf ("%s - %s<br />\n", $row[0], $row[1]);
    // ou tous les résultats de la requête :
    //while(list($host, $user) = mysql_fetch_row($result))
    //{
    // echo "$host - $user<br />";
    //}

    /* Tableau associatif */
    //$row = mysql_fetch_array($result, MYSQLI_ASSOC);
    //printf ("%s - %s<br />\n", $row["Host"], $row["User"]);
    // ou tous les résultats de la requête :
    while($row = mysql_fetch_array($result))
    {
        $host = $row["Host"];
        $user = $row["User"];
        echo "$host - $user<br />";
    }

    /* Libération des résultats */
    mysql_free_result($result);
}

```

```
mysql_close($link);  
?>
```

Traitement des résultats avec les fonctions mysql (obsolète depuis PHP 5.5)

```
<?php  
$pdo_db = new PDO('mysql:host=localhost;dbname=mysql', 'root', 'password') or die("Echec de  
la création de l'instance PDO !");  
  
if ($result = $pdo_db->query("SELECT Host,User FROM `user` LIMIT 0 , 30"))  
    printf("Classe PDO : la requête a retourné %d enregistrement(s).<br />\n", $result->  
        rowCount());  
  
if ($result = $pdo_db->query("SELECT Host, User FROM `user` ORDER BY User DESC LIMIT 0 , 30"  
    ))  
{  
    printf("Classe PDO : la requête a retourné %d enregistrement(s).<br />\n", $result->  
        rowCount());  
    /* Tableau indexé */  
    //$row = $result->fetch(PDO::FETCH_NUM);  
    //printf ("%s - %s<br />\n", $row[0], $row[1]);  
  
    /* Tableau associatif */  
    //$row = $result->fetch(PDO::FETCH_ASSOC);  
    //printf ("%s - %s<br />\n", $row["User"], $row["Host"]);  
  
    // ou tous les résultats de la requête :  
    $datas = $result->fetchAll();  
    echo "<pre>"; print_r ($datas); echo "</pre>";  
}  
  
unset($pdo_db);  
?>
```

Traitement des résultats avec la classe PDO