

Introduction à PHP, MySQL et AJAX

Contenu du cours

- Introduction à PHP
 - Syntaxe
 - Variables
 - Boucles
 - ...
- PHP avancé
 - Sessions
 - Manipulation de fichiers
 - Cookies
 - ...
- PHP+MySQL
- AJAX

Qu'est-ce que PHP?

- PHP signifie P HP: H ypertext P reprocessor
- PHP est un langage de script côté serveur ,
comme l'ASP
- Les scripts PHP sont exécutés sur le serveur
- PHP supporte de nombreuses bases de données
(MySQL, Informix, Oracle, Sybase, Solid,
PostgreSQL, ODBC génériques, etc)
- PHP est un logiciel open source
- PHP est gratuit à télécharger et à utiliser

Qu'est-ce que qu'un fichier PHP ?

- Les fichiers PHP peuvent contenir du texte, des balises HTML et des scripts
- Les fichiers PHP sont retournés au navigateur HTML
- Les fichiers PHP ont une extension de fichier ".Php", ".Php3", ou ".Phtml"

Qu'est-ce que MySQL?

- MySQL est un serveur de base de données
- MySQL est idéal pour les petites et les grandes applications
- MySQL supporte la norme SQL
- MySQL compile sur un certain nombre de plateformes
- MySQL est gratuit à télécharger et à utiliser
- PHP associé à MySQL sont multi-plateforme (vous pouvez développer sous Windows et déployer sous Unix)

Pourquoi PHP ?

- PHP fonctionne sur différentes plates-formes (Windows, Linux, Unix, etc)
- PHP est compatible avec presque tous les serveurs utilisés aujourd'hui (Apache, IIS, etc)
- PHP est gratuit à télécharger sur www.php.net
- PHP est facile à apprendre et fonctionne de manière efficace côté serveur

Syntaxe de base en PHP

- Un bloc de script PHP commence par `<? Php` et se termine avec `?>`
- Un bloc de script PHP peut être placé n'importe où dans le document.
- Certains serveurs supportent la syntaxe `<? et ?>`
- Pour une compatibilité maximale, il est recommandé d'utiliser la forme standard (`<? Php`) plutôt que la forme abrégée.
- Un fichier PHP contient normalement des balises HTML et quelques scripts PHP

Exemple

```
<html>
<body>
<? php
echo "Hello World"; echo "Bonjour tout le monde";
?>
</ body>
</ html>
```

- Chaque ligne de code en PHP doit se terminer par un point-virgule
- La virgule est un séparateur et est utilisée pour distinguer une série d'instructions d'une autre

Commentaires en PHP

- En PHP, on utilise `//` pour faire une seule ligne de commentaire ou `/*` et `*/` pour un bloc

```
<html>
<body>
<? Php
// Ceci est un commentaire
/*
Ici c'est
Un
bloc
*/
?>
</ body>
</ html>
```

Les variables en PHP

- Les variables sont utilisées pour stocker des valeurs, comme les chaînes de texte, des chiffres ou des tableaux
- Quand une variable est définie, elle peut être utilisée plusieurs fois dans votre script
- Toutes les variables en PHP débutent par le symbole \$

```
<? php  
$ txt = "Bonjour tout le monde!";  
$ nombre = 16;  
?>
```

Variables PHP non typées

- En PHP, une variable PHP n'a pas besoin d'être déclarée avant d'être fixé
- Dans l'exemple précédent, on a pas besoin de définir le type de données que va contenir la variable
- PHP convertit automatiquement la variable au bon type de données, selon la façon dont ils sont fixés (comme en javascript)

Variables : règles de nommage

- Un nom de variable doit commencer par une lettre ou un caractère de soulignement "_" »
- Un nom de variable ne peut contenir que des caractères alphanumériques et des traits de soulignement (az, AZ, 0-9 et _)
- Un nom de variable ne devrait pas contenir d'espaces
- Si un nom de variable est en plusieurs mots, il devra être séparé par le caractère de soulignement (\$my_string), ou avec majuscule (\$maChaine)

Chaînes de caractères

- Exemple :

```
<?php
$txt="Hello World";
echo $txt;
?>
```
- Concaténation (.) :

```
<?php
$txt1="Hello World";
$txt2="1234";
echo $txt1 . " " . $txt2;
?>
```
- Longueur d'une chaîne (strlen()) :

```
<?php
echo strlen("Hello world!");
?>
```
- Position d'une sous-chaîne (strpos()) :

```
<?php
echo strpos("Hello world!","world");
?>
```

Plus de fonctions sur http://w3schools.com/php/php_ref_string.asp

Opérateurs en PHP

- Opérateurs arithmétiques

Operator	Description	Example	Result
+	Addition	x=2 x+2	4
-	Subtraction	x=2 5-x	3
*	Multiplication	x=4 x*5	20
/	Division	15/5 5/2	3 2.5
%	Modulus (division remainder)	5%2 10%8 10%2	1 2 0
++	Increment	x=5 x++	x=6
--	Decrement	x=5 x--	x=4

Opérateurs en PHP

- Opérateurs d'affectation

Operator	Example	Is The Same As
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
=	x=y	x=x*y
/=	x/=y	x=x/y
.=	x.=y	x=x.y
%=	x%=y	x=x%y

Opérateurs en PHP

- Opérateurs de comparaison

Operator	Description	Example
==	is equal to	5==8 returns false
!=	is not equal	5!=8 returns true
>	is greater than	5>8 returns false
<	is less than	5<8 returns true
>=	is greater than or equal to	5>=8 returns false
<=	is less than or equal to	5<=8 returns true

Opérateurs en PHP

- Opérateurs logiques

Operator	Description	Example
&&	and	x=6 y=3 (x < 10 && y > 1) returns true
	or	x=6 y=3 (x==5 y==5) returns false
!	not	x=6 y=3 !(x==y) returns true

If...else en PHP

- Syntaxe

```
if (condition)
    code to be executed if condition is true;
else
    code to be executed if condition is false;
```

- Exemples :

```
<html>
<body>
<?php
$d=date("D");
if ($d=="Fri")
    echo "Have a nice weekend!";
else
    echo "Have a nice day!";
?>
</body>
</html>
```

```
<html>
<body>
<?php
$d=date("D");
if ($d=="Fri")
{
    echo "Hello!<br />";
    echo "Have a nice weekend!";
    echo "See you on Monday!";
}
?>
</body>
</html>
```

elseif en PHP

- **Syntaxe :**
if (condition)
 code to be executed if condition is true;
elseif (condition)
 code to be executed if condition is true;
else
 code to be executed if condition is false;
- **Exemple :**

```
<html>
<body>
<?php
$d=date("D");
if ($d=="Fri")
    echo "Have a nice weekend!";
elseif ($d=="Sun")
    echo "Have a nice Sunday!";
else
    echo "Have a nice day!";
?>
</body>
</html>
```

switch en PHP

- **Syntaxe :**

```
switch (expression)
{
case label1:
    code to be executed if expression = label1;
    break;
case label2:
    code to be executed if expression = label2;
    break;
default:
    code to be executed if expression is different from both label1 and label2;
}
```
- **Exemple :**

```
<?php
switch ($x)
{
case 1:
    echo "Number 1";
    break;
case 2:
    echo "Number 2";
    break;
default:
    echo "No number between 1 and 2";
}
?>
```

Tableaux en PHP

- Il existe trois différents types de tableaux:
 - Un tableau avec une clé ID numérique
 - Un tableau où chaque ID clé est associée à une valeur
 - Un tableau contenant un ou plusieurs tableaux

Tableaux numériques

- Il existe différentes façons de créer un tableau numérique :

```
$names = array("Peter","Quagmire","Joe");
```

```
<?php
```

```
$names[0] = "Peter";
```

```
$names[1] = "Quagmire";
```

```
$names[2] = "Joe";
```

```
echo $names[1] . " and " . $names[2] . " are " . $names[0] . "'s neighbors";
```

```
?>
```

Tableaux associatifs

- Avec des tableaux associatifs, nous pouvons utiliser les valeurs de clés et leur assigner des valeurs:

```
$ages = array("Peter"=>32, "Quagmire"=>30, "Joe"=>34);
```

```
<?php
```

```
$ages['Peter'] = "32";  
$ages['Quagmire'] = "30";  
$ages['Joe'] = "34";
```

```
echo "Peter is " . $ages['Peter'] . " years old."  
?>
```

Tableaux multidimensionnels

- Exemple :

```
$families = array  
(  
    "Griffin"=>array("Peter","Lois","Megan"),  
    "Quagmire"=>array("Glenn"),  
    "Brown"=>array("Cleveland","Loretta","Junior")  
);
```
- Equivalent à :

```
$families = array  
(  
    [Griffin] => Array([0] => Peter,[1] => Lois,[2] => Megan)  
    [Quagmire] => Array([0] => Glenn)  
    [Brown] => Array([0] => Cleveland,[1] => Loretta,[2] => Junior)  
)
```


Boucles PHP

- While :

```
<?php
$i=1;
while($i<=5)
{
    echo "The number is " . $i . "<br />";
    $i++;
}
?>
```

- Do ... while :

```
<?php
$i=0;
do
{
    $i++;
    echo "The number is " . $i . "<br />";
}
while ($i<5);
?>
```

Boucles PHP

- **For :**

```
<?php
for ($i=1; $i<=5; $i++)
{
    echo "Hello World!<br />";
}
?>
```
- **Foreach :**

```
<?php
$arr=array("one", "two", "three");

foreach ($arr as $value)
{
    echo "Value: " . $value . "<br />";
}
?>
```

Fonctions en PHP

- Une fonction est un bloc de code qui peut être exécuté à chaque fois nous en avons besoin
- Création de fonctions en PHP:
 - Toutes les fonctions commencent par le mot 'function ()'
 - Le nom des fonctions doit être explicite
 - Les noms de fonctions commencent par une lettre ou underscore
 - () pour définir les paramètres
 - {} pour le corps de la fonction

Exemple

```
<?php
function writeMyName()
{
    echo "Kai Jim Refsnes";
}

echo "Hello world!<br />";
echo "My name is ";
writeMyName();
echo "<br />That's right, ";
writeMyName();
echo " is my name.";
?>
```

Paramètres de fonctions

```
<?php
function writeMyName($fname,$punctuation)
{
    echo $fname . " Refsnes" . $punctuation . "<br />";
}
```

```
echo "My name is ";
writeMyName("Kai Jim",".");
```

```
echo "My name is ";
writeMyName("Hege","!");
```

```
echo "My name is ";
writeMyName("Ståle","...");
?>
```

Valeur de retour d'une fonction

```
<?php
function add($x,$y)
{
    $total = $x + $y;
    return $total;
}

echo "1 + 16 = " . add(1,16);
?>
```

Formulaires et champs d'entrée

- Exemple :

```
<html>
<body>
<form action="welcome.php" method="post">
  Name: <input type="text" name="name" />
  Age: <input type="text" name="age" />
  <input type="submit" />
</form>
</body>
</html>
```
- Le fichier welcome.php :

```
<html>
<body>

Welcome <?php echo $_POST["name"]; ?>.<br />
You are <?php echo $_POST["age"]; ?> years old.

</body>
</html>
```

\$_GET, \$_POST, \$_REQUEST

- Suivant la manière d'envoyer les données sur le serveur, on utilisera soit \$_GET soit \$_POST afin de récupérer les champs et valeurs des formulaires
- \$_REQUEST contient à la fois \$_GET, \$_POST et \$_COOKIE

PHP avancé : quelques fonctions

Date() en PHP

- Syntaxe : date (format, timestamp)
 - Format : spécifie le format d'horodatage
 - Timestamp : temps écoulé depuis le 01/01/70 (facultatif)
- Exemple :

```
<?php
echo date("Y/m/d");
echo "<br />";
echo date("Y.m.d");
echo "<br />";
echo date("Y-m-d");
?>
```
- mktime (heure, minute, seconde, jour, mois, année, is_dst) :
pour définir un timestamp

Inclure des fichiers en PHP

- Syntaxe : `include(url)`
- Exemple :

```
<html>
<body>

<?php include("header.php"); ?>

<h1>Welcome to my home page</h1>

<p>Some text</p>

</body>
</html>
```

Inclure des fichiers en PHP

- Permet facilement de faire un squelette d'un site web
- Exemple :

```
<html>
<body>
<?php

include('header.php');
include('menu.php');
include('body.php');
include('footer.php');

?>
</body>
</html>
```

Inclure des fichiers en PHP

- Syntaxe : `require(url)`
- Lors d'une erreur, la fonction `include()` permet de continuer le chargement de la page
- `require()` arrête complètement celui ci
- Exemple :

```
<html>
<body>
<?php
include("wrongFile.php");
echo "Hello World!";
?>
</body>
</html>
```

Inclure des fichiers en PHP

- **Sortie avec include :**
Warning: include(wrongFile.php) [function.include]:
failed to open stream:
No such file or directory in C:\home\website\test.php on line 5

Warning: include() [function.include]:
Failed opening 'wrongFile.php' for inclusion
(include_path='.;C:\php5\pear')
in C:\home\website\test.php on line 5

Hello World!
- **Et avec require :**
Warning: require(wrongFile.php) [function.require]:
failed to open stream:
No such file or directory in C:\home\website\test.php on line 5

Fatal error: require() [function.require]:
Failed opening required 'wrongFile.php'
(include_path='.;C:\php5\pear')
in C:\home\website\test.php on line 5

Ouverture/fermeture de fichiers en PHP

- Syntaxe open(file,mode)
 - File : url du fichier
 - Mode parmi r,r+,w,w+,a,a+,x,x+
- Exemple :

```
<?php
$file=fopen("welcome.txt","r") or exit("Unable to open file!");
?>
```

- Fermeture : fclose()

```
<?php
$file = fopen("test.txt","r");
//some code to be executed
fclose($file);
?>
```

Fin de fichier et lecture en PHP

- Tester la fin d'un fichier : `feof($file)`

```
if (feof($file)) echo "End of file";
```

- Lecture ligne par ligne : `fgets($file)`
- Lecture caractère par caractère : `fgetc($file)`
- Exemple :

```
<?php
$file=fopen("welcome.txt","r") or exit("Unable to open file!");
while (!feof($file))
{
    echo fgetc($file);
}
fclose($file);
?>
```


Upload de fichiers en PHP

- Formulaire d'upload :

```
<html>
<body>
<form action="upload_file.php" method="post"
enctype="multipart/form-data">
<label for="file">Filename:</label>
<input type="file" name="file" id="file" />
<br />
<input type="submit" name="submit" value="Submit" />
</form>
</body>
</html>
```

- Script d'upload (serveur):

```
<?php
if ($_FILES["file"]["error"] > 0)
{
    echo "Error: " . $_FILES["file"]["error"] . "<br />";
}
else
{
    echo "Upload: " . $_FILES["file"]["name"] . "<br />";
    echo "Type: " . $_FILES["file"]["type"] . "<br />";
    echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
    echo "Stored in: " . $_FILES["file"]["tmp_name"];
}
?>
```

Upload de fichiers en PHP

- On récupère ensuite le fichier sur le serveur grâce à la variable `$_FILES` (tableau à deux dimensions)
 - * `$_FILES["file"]["name"]` : nom du fichier
 - * `$_FILES["file"]["type"]` : type du fichier
 - * `$_FILES["file"]["size"]` : taille en octets du fichier
 - * `$_FILES["file"]["tmp_name"]` : nom temporaire du fichier sur le serveur
 - * `$_FILES["file"]["error"]` : code d'erreur

Upload de fichiers en PHP

- Sauvegarde du fichier :

```
<?php
if ((($_FILES["file"]["type"] == "image/gif")
|| ($_FILES["file"]["type"] == "image/jpeg")
|| ($_FILES["file"]["type"] == "image/pjpeg"))
&& ($_FILES["file"]["size"] < 20000))
{
    if ($_FILES["file"]["error"] > 0)
        {echo "Return Code: " . $_FILES["file"]["error"] . "<br /> »;}
    else
    {
        echo "Upload: " . $_FILES["file"]["name"] . "<br />";
        echo "Type: " . $_FILES["file"]["type"] . "<br />";
        echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
        echo "Temp file: " . $_FILES["file"]["tmp_name"] . "<br />";

        if (file_exists("upload/" . $_FILES["file"]["name"]))
            {echo $_FILES["file"]["name"] . " already exists. »;}
        else
        {
            move_uploaded_file($_FILES["file"]["tmp_name"],
            "upload/" . $_FILES["file"]["name"]);
            echo "Stored in: " . "upload/" . $_FILES["file"]["name"];
        }
    }
}
else
{
    echo "Invalid file";
}
?>
```

Cookies en PHP

- Seul mécanisme utilisable afin de stocker des informations sur le client
- Création de cookies : `setcookie(name, value, expire, path, domain);`
- Exemple :

```
<?php  
setcookie("user", "Alex Porter", time()+3600);  
?>
```

Cookies en PHP

- Récupération d'un cookie : `$_COOKIE`

- Exemple :

```
<?php
if (isset($_COOKIE["user"]))
    echo "Welcome " . $_COOKIE["user"] . "!\n";
else
    echo "Welcome guest!\n";
?>
```

- La fonction `isset()` permet de vérifier l'existence du cookie

Cookies en PHP

- Suppression d'un cookie: lui mettre une date d'expiration antérieure à la date actuelle
- Exemple :

```
<?php  
// set the expiration date to one hour ago  
setcookie("user", "", time()-3600);  
?>
```

Sessions en PHP

- Comment donner la possibilité à l'utilisateur de visiter un site sans pour autant perdre des informations qu'il voudrait stocker : mécanisme de sessions
 - Permet de conserver des valeurs d'un formulaire sans passer par les cookies
 - Gestion d'un panier d'articles dans le cadre d'un site e-commerce
- Les variables et objets de sessions sont stockés côté serveur

Sessions en PHP

- Commencer une session : `session_start()`
- Exemple :

```
<?php session_start(); ?>
<html>
<body>
</body>
</html>
```
- À insérer obligatoirement avant le tag html

Sessions en PHP

- Stocker des objets dans la session : `$_SESSION`
- Exemple :

```
<?php
session_start();
// store session data
$_SESSION['views']=1;
?>
<html>
<body>
<?php
//retrieve session data
echo "Pageviews=". $_SESSION['views'];
?>
</body>
</html>
```

Sessions en PHP

- Tester si une variable existe : isset()
- Exemple (compteur de pages):

```
<?php
session_start();
if(isset($_SESSION['views']))
    $_SESSION['views']=$_SESSION['views']+1;
else
    $_SESSION['views']=1;
echo "Views=". $_SESSION['views'];
?>
```

- Les variables stockées dans les sessions sont visibles dans toutes les pages du même site

Sessions en PHP

- Pour supprimer une variable stockée dans la session : `unset()`

- Exemple :

```
<?php
unset($_SESSION['views']);
?>
```

- Pour supprimer la session : `session_destroy()`

- Exemple :

```
<?php
session_destroy();
?>
```

Envoyer des e-mails en PHP

- La fonction mail : mail(to,subject,message,headers,parameters)
 - To : l'adresse du destinataire
 - Subject : le titre du mail
 - Message : le corps du message (\n pour une nouvelle ligne (limite 70 caractères))
 - Headers : entêtes spécifiques (comme cc et Bcc)
 - Parameters : (paramètres spécifiques pour le serveur de mail)
- Pour pouvoir l'utiliser il faut obligatoirement qu'un serveur de mail soit installé (ex: sendmail)

Envoyer des e-mails en PHP

- Exemple (simple mail):

```
<?php  
  
$to = "someone@example.com";  
$subject = "Test mail";  
$message = "Hello! This is a simple email message.";  
$from = "someoneelse@example.com";  
$headers = "From: $from";  
mail($to,$subject,$message,$headers);  
echo "Mail Sent."  
  
?>
```

Envoyer des e-mails en PHP

- Récupération à partir d'un formulaire:

```
<html><body>
<?php
if (isset($_REQUEST['email']))
//if "email" is filled out, send email
{
//send email
$email = $_REQUEST['email'] ;
$subject = $_REQUEST['subject'] ;
$message = $_REQUEST['message'] ;
mail( "someone@example.com", "Subject: $subject",
$message, "From: $email" );
echo "Thank you for using our mail form";
}
else//if "email" is not filled out, display the form
{
echo "<form method='post' action='mailform.php'>
Email: <input name='email' type='text' /><br />
Subject: <input name='subject' type='text' /><br />
Message:<br />
<textarea name='message' rows='15' cols='40'>
</textarea><br />
<input type='submit' />
</form>";
}
?>
</body></html>
```

Injection e-mail

- Dans l'exemple précédent : il est possible de modifier l'entête de la requête afin d'envoyer un mail à plusieurs personnes
- Utilisation de la fonction `filter_var($field,param)`
 - `$field` : variable à vérifier
 - Param :
 - `FILTER_SANITIZE_EMAIL` : supprime les caractères illégaux
 - `FILTER_VALIDATE_EMAIL` : valide ou non la chaîne comme un mail valide
- Plus d'informations : http://w3schools.com/php/php_filter.asp

Intercepter des erreurs en PHP

- Exemple : on veut ouvrir un fichier qui n'existe pas

Warning: fopen(welcome.txt) [function.fopen]: failed to open stream:
No such file or directory in C:\webfolder\test.php on line 2

- Utilisation de la fonction die() :

```
<?php
if(!file_exists("welcome.txt"))
{die("File not found");}
else
{$file=fopen("welcome.txt","r");}
?>
```

- Possibilité de définir ses propres interceptions et déclencheurs

Exceptions en PHP

- Exemple :

```
<?php
//create function with an exception
function checkNum($number)
{
    if($number>1)
    {
        throw new Exception("Value must be 1 or below");
    }
    return true;
}
//trigger exception in a "try" block
try
{
    checkNum(2);
    //If the exception is thrown, this text will not be shown
    echo 'If you see this, the number is 1 or below';
}
//catch exception
catch(Exception $e)
{
    echo 'Message: ' . $e->getMessage();
}
?>
```

Filtres en PHP

- Mécanisme permettant de vérifier la validité de certains champs, informations
- Exemple (test d'un entier) :

```
<?php
$int = 123;
if(!filter_var($int, FILTER_VALIDATE_INT))
{ echo("Integer is not valid");}
else
{echo("Integer is valid");}
?>
```

- D'autres filtres : http://w3schools.com/php/php_ref_filter.asp

MySQL

- Introduction
- Connexion à une base de données
- Création de tables
- Sélection d'informations dans une BD (where, order by)
- Insertion/maj/suppression d'éléments dans une BD

Connexion à une base de données MySQL

- `mysql_connect(server,user,password)`
 - Server : localhost:3306 (le port est modifiable)
 - User : le login
 - Password : un mot de passe si défini
- Exemple :

```
<?php
$con = mysql_connect("localhost","peter","abc123");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}
// some code
?>
```

Déconnexion de MySQL

- `mysql_close($con)`
- Exemple :

```
<?php
$con = mysql_connect("localhost","peter","abc123");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}

// some code

mysql_close($con);
?>
```

Création DB/tableaux

- `mysql_query('requete',$con)`
- Pour une base de données :

```
<?php
$con = mysql_connect("localhost","peter","abc123")
or die('Could not connect: ' . mysql_error());
if (mysql_query("CREATE DATABASE my_db",$con))
{
    echo "Database created";
}
else
{
    echo "Error creating database: " . mysql_error();
}
mysql_close($con);
?>
```

Création DB/tableaux

- Pour un tableau :

```
mysql_select_db("my_db", $con);  
$sql = "CREATE TABLE Persons  
(  
  FirstName varchar(15),  
  LastName varchar(15),  
  Age int  
)";
```

```
// Execute query  
mysql_query($sql,$con);
```

- La DB doit exister et être sélectionnée pour créer la table

Création DB/tableaux

- Pour un tableau (avec clé primaire) :

```
$sql = "CREATE TABLE Persons  
(  
  personID int NOT NULL AUTO_INCREMENT,  
  PRIMARY KEY(personID),  
  FirstName varchar(15),  
  LastName varchar(15),  
  Age int  
)";  
  
mysql_query($sql,$con);
```


Insert MySQL

- Example :

```
<?php
$con = mysql_connect("localhost","peter","abc123");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}

mysql_select_db("my_db", $con);

mysql_query("INSERT INTO Persons (FirstName, LastName, Age)
VALUES ('Peter', 'Griffin', '35')");

mysql_query("INSERT INTO Persons (FirstName, LastName, Age)
VALUES ('Glenn', 'Quagmire', '33')");

mysql_close($con);
?>
```

Insert MySQL

- Avec un formulaire :

```
<form action="insert.php" method="post">
Firstname: <input type="text" name="firstname" />
Lastname: <input type="text" name="lastname" />
Age: <input type="text" name="age" />
<input type="submit" />
</form>
```
- Le fichier insert.php :

```
<?php
$con = mysql_connect("localhost","peter","abc123") or die('Could not connect: ' .
mysql_error());
mysql_select_db("my_db", $con);
$sql="INSERT INTO Persons (FirstName, LastName, Age)
VALUES('$_POST[firstname]', '$_POST[lastname]', '$_POST[age]') ";
if (!mysql_query($sql,$con))
{
    die('Error: ' . mysql_error());
}
echo "1 record added »;
mysql_close($con)
?>
```

Select MySQL

- Exemple 1 :

```
<?php
$con = mysql_connect("localhost","peter","abc123");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}
mysql_select_db("my_db", $con);
$result = mysql_query("SELECT * FROM Persons");
while($row = mysql_fetch_array($result))
{
    echo $row['FirstName'] . " " . $row['LastName'];
    echo "<br />";
}
mysql_close($con);
?>
```

Select MySQL

- **Exemple 2 :**

```
<?php
$con = mysql_connect("localhost","peter","abc123");
mysql_select_db("my_db", $con);
$result = mysql_query("SELECT * FROM Persons");

echo "<table border='1'>
<tr>
<th>Firstname</th>
<th>Lastname</th>
</tr>";

while($row = mysql_fetch_array($result))
{
    echo "<tr>";
    echo "<td>" . $row['FirstName'] . "</td>";
    echo "<td>" . $row['LastName'] . "</td>";
    echo "</tr>";
}
echo "</table>";

mysql_close($con);
?>
```

Where MySQL

- Example :

```
<?php
$con = mysql_connect("localhost","peter","abc123");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}
mysql_select_db("my_db", $con);

$result = mysql_query("SELECT * FROM Persons
WHERE FirstName='Peter'");

while($row = mysql_fetch_array($result))
{
    echo $row['FirstName'] . " " . $row['LastName'];
    echo "<br />";
}
?>
```

Order by MySQL

- Example :

```
<?php
$con = mysql_connect("localhost","peter","abc123");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}
mysql_select_db("my_db", $con);

$result = mysql_query("SELECT * FROM Persons ORDER BY age");

while($row = mysql_fetch_array($result))
{
    echo $row['FirstName'];
    echo " " . $row['LastName'];
    echo " " . $row['Age'];
    echo "<br />";
}
mysql_close($con);
?>
```

Update MySQL

- Example :

```
<?php
$con = mysql_connect("localhost","peter","abc123");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}
mysql_select_db("my_db", $con);

mysql_query("UPDATE Persons SET Age = '36'
WHERE FirstName = 'Peter' AND LastName = 'Griffin'");

mysql_close($con);
?>
```

Delete MySQL

- Exemple :

```
<?php
$con = mysql_connect("localhost","peter","abc123");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}

mysql_select_db("my_db", $con);

mysql_query("DELETE FROM Persons WHERE LastName='Griffin'");

mysql_close($con);
?>
```


SQL références

- <http://w3schools.com/sql/default.asp>