

Technologie Web Avancée

Chapitre 4 : PHP/MYSQL

Le contenu de ce document est soumis à la Licence de Documentation Libre (**GNU Free Documentation License**).

Plan du Chapitre

- **1. PHP**
- **2. MYSQL**
- **3. PHP/MYSQL**

Technologie Web Avancée

PARTIE 1 : PHP

Le contenu de ce document est soumis à la Licence de Documentation Libre (**GNU Free Documentation License**).

Plan du Chapitre

- 1. Introduction
- 2. Variable
- 3. Operateur
- 4. Affichage
- 5. Commentaire
- 6. Structure de Contrôle
- 7. Tableau
- 8. Fonction
- 9. Inclusion
- 10. Redirection

INTRODUCTION

PHP est un langage de programmation qui s'intègre dans vos pages HTML. Il permet entre autres de rendre automatiques des tâches répétitives, notamment grâce à la communication avec une base de données (utilisation la plus courante de PHP).

Intégration d'un script dans une page :

- ❖ Les pages web sont au format HTML. Les pages web dynamiques générées avec PHP sont au format PHP. Le code source PHP est directement insérer dans le fichier html entre cette syntaxe :

<?php ... ?>

1. Variables

- Les identificateurs de variable sont précédés du symbole « \$ ».

Exemple : **\$toto.**

2. TYPE D' ENTIER

- Les variables peuvent être de type entier (**integer**), réel (**double**), chaîne de caractères (**string**), tableau (**array**), objet (**object**), booléen (**boolean**).

2.1 Constant

L'utilisateur peut définir des constantes dont la valeur est fixée une fois pour toute. Les constantes ne portent pas le symbole \$ (dollars) en début d'identificateur et ne sont pas modifiables.

define("var",valeur) : définit la constante **var** (sans \$) de valeur **valeur**

Exemple :

```
define("author","Djama");  
echo author;
```

```
// affiche Djama
```

3. OPERATEUR

Opérateurs arithmétiques :

+ (addition), **-** (soustraction), ***** (multiplié), **/** (divisé), **%** (modulo), **++** (incrément), **--** (décrément). Ces deux derniers peuvent être pré ou post fixés

Opérateurs d'assignement :

= (affectation), ***=** (**\$x*=\$y** équivalent à **\$x=\$x*\$y**), **/=**, **+=**, **-=**, **%=**

Opérateurs logiques :

and, **&&** (et), **or**, **||** (ou), **xor** (ou exclusif), **!** (non)

Opérateurs de comparaison :

== (égalité), **<** (inférieur strict), **<=** (inférieur large), **>**, **>=**, **!=** (différence)

Opérateurs de concaténation :

4. Affichage

Les fonctions d'affichage :

echo() : écriture dans le navigateur ou **print()** : écriture dans le navigateur

Exemple :

`<?php echo 'Bonjour le monde !' ; ?>` **ou** `< ?php print 'Bonjour le monde !' ; ?>`

5. Commentaire

Comme dans tout autre langage de programmation, PHP dispose sa propre syntaxe de commentaires.

Les commentaires permettent de bien expliquer une partie du code afin qu'on puisse déchiffrer en cas de maintenance ou de travail collaboratif (plusieurs développeurs travaillant sur le même programme).

Il existe deux sortes de commentaire. Le commentaire sur une seule ligne et le commentaire multilignes.

Une seule ligne	Multilignes
<pre><?php // Ceci est un premier commentaire sur une ligne echo 'Hello World !'; ?></pre>	<pre><?php /* Ce programme a été écrit par Dr. Kadar Il affiche la chaîne 'Hello Word !' à l'écran */ echo 'Hello World !'; ?></pre>

6. STRUCTURE DE CONTROLE (1/2)

- Structures conditionnelles

```
if(condition1) {  
    # instructions à exécuter si la  
    condition1 est vraie...  
}  
elseif (condition2) {  
    instructions à exécuter si la  
    condition2 est vraie...  
}  
elseif (condition3) {  
    instructions à exécuter si la  
    condition3 est vraie...  
}  
else {  
    instructions à exécuter si  
    aucune des conditions n'est  
    vraie...  
}
```

```
switch(expression) {  
    case resultat1 :
```

```
    # instructions à exécuter si  
    l'expression vaut resultat1...  
    break ;
```

```
    case resultat2 :
```

```
    # instructions à exécuter si  
    l'expression vaut resultat2...  
    break ;
```

```
    .....
```

```
    default :
```

```
    # instructions à exécuter en  
    dernier recours...  
}
```

6. STRUCTURE DE CONTROLE (2/2)

○ Structures de boucles

for(expression1 ; condition ; expression2) {
instructions à exécuter tant que la condition est vraie...

}

Exemple : compter jusqu'à 10

```
for ($i = 1 ; $i <= 10 ; $i++) {  
echo "$i " ;  
}
```

Affiche 1 2 3 4 5 6 7 8 9 10.

While (condition){

instructions à exécuter tant que la condition est vraie...

}

Exemple : compter jusqu'à 10

```
$i = 1 ;  
while ($i <= 10) {  
echo "$i " ;  
$i++ ;  
}
```

Affiche 1 2 3 4 5 6 7 8 9 10.

do {

instructions à exécuter tant que la condition est vraie...

} **while**(condition);

Exemple : compter jusqu'à 10

```
$i = 1 ;  
do {  
echo "$i " ;  
$i++ ;  
} while ($i <= 10) ;
```

Affiche 1 2 3 4 5 6 7 8 9 10.

7. TABLEAU (1/2)

- × Une variable tableau est de type **array**. Un tableau accepte des éléments de tout type.
 - + Exemple :

- × `$tab_colors = array('red', 'yellow', 'blue', 'white');`
 - × `$tab = array('djama', 2002, 20.5, $name);`

- × Indice initiale d'un tableau est 0
- × Parcours d'un tableau :
 - + *Exemple 1 :*

```
$i=0;
while($i <count($tab)) {           // count() retourne le nombre d'éléments
    echo $tab[$i].'\n';
    $i++;
}
```

- + *Exemple 2 :*

```
foreach($tab as $i) {
    echo $i."\n";
}
```

- × La variable **\$elem** prend pour valeurs successives tous les éléments du tableau **\$tab**.

7. TABLEAU (2/2)

- × Quelques fonctions utiles:
 - + **count(\$tab)** : retournent le nombre d'éléments du tableau
 - × **foreach(\$tab as \$elem)**: permet a la variable **\$elem** de prendre pour valeurs successives tous les éléments du tableau **\$tab**.
 - + **in_array(\$var,\$tab)** : dit si la valeur de **\$var** existe dans le tableau **\$tab**
 - + **sort(\$tab)** : trie alphanumérique les éléments du tableau
 - + **strlen(\$tab)** : Calcule la taille d'une chaîne
 - + **empty(\$tab)** : Détermine si une variable est vide ou non
 - + **isset(\$tab)** : Déterminer si une variable est bien définie

8. FONCTION

- Les fonctions peuvent prendre des arguments dont il n'est pas besoin de spécifier le type. Elles peuvent de façon optionnelle retourner une valeur.

Exemple :

```
<?php function calculerSomme($ a, $ b)
{ // Déclaration de la variable locale
    $somme = $a + $b;
return $somme;
}
// Affichage du résultat
echo calculerSomme($ a, $ b);
?>
```

```
calculerSomme(8,1);
```

9. INCLUSION

- × On peut inclure dans un script php le contenu d'un autre fichier.
- + **include** évalue et insert à chaque appel (même dans une boucle) le contenu du fichier passé en argument.
 - × *Exemple :*
 - * **include**("fichier.php");
- + **require** insert dans le code le contenu du fichier spécifié même si ce n'est pas du code php. Est équivalent au préprocesseur *#include* du C.
 - × *Exemple :*
 - * **require**("fichier.php");

10. REDIRECTION

- `<?php`
`header("Location: home2.php");`
`exit();`
- `?>`

- Ce script effectue une redirection vers une autre page. C'est-à-dire que le navigateur du client en voyant cet entête *Location* va charger directement la page indiquée sans regarder la suite du script. La fonction **exit** est là pour parer au cas impossible où le script continuerait son exécution.

Technologie Web Avancée

Partie 2 : MYSQL

Le contenu de ce document est soumis à la Licence de Documentation Libre (**GNU Free Documentation License**).

Plan du Chapitre

- **1. Présentation**
- **2. Création/sélection d'une base de données**
- **3. Gestion de la base de données**
- **4. Affichage d'une table**

INTRODUCTION

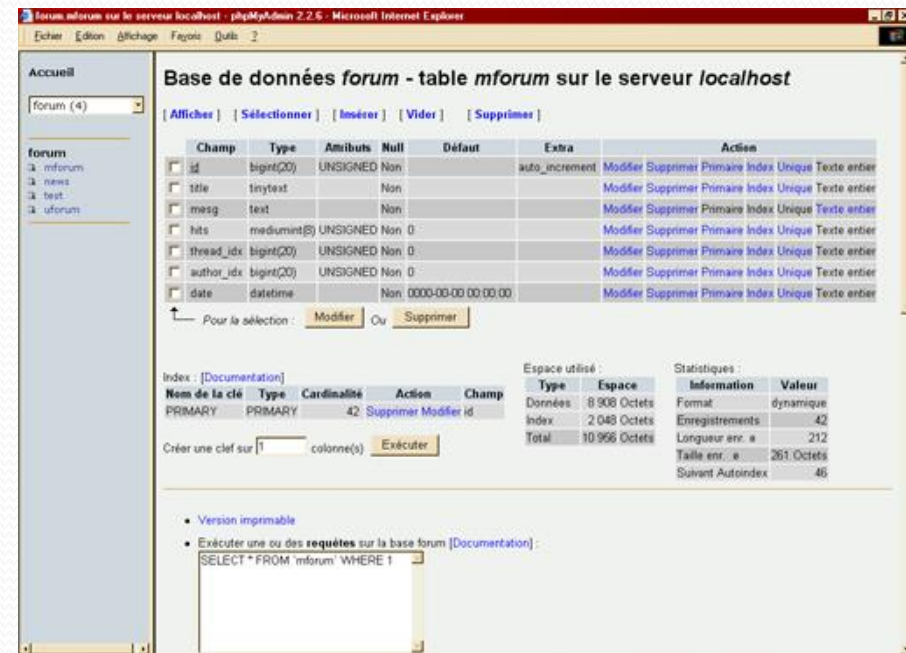
- MySQL est une base de données implémentant le langage de requête SQL un langage relationnel très connu.
- Il existe un outil libre et gratuit développé par la communauté des programmeurs libres : phpMyAdmin qui permet l'administration aisée des bases de données MySQL avec php. Il est disponible sur : <http://sourceforge.net/projects/phpmyadmin/> et <http://www.phpmyadmin.net>

1. Presentation

• L'outil phpMyAdmin est développé en PHP et offre une interface intuitive pour l'administration des base de données du serveur.

• Cet outil permet de :

- ☐ créer de nouvelles bases
- ☐ créer/modifier/supprimer des tables
- ☐ afficher/ajouter/modifier/supprimer des tups dans des tables
- ☐ effectuer des sauvegarde de la structure et/ou des données
- ☐ effectuer n'importe quelle requête
- ☐ gérer les privilèges des utilisateurs



2. Création/sélection d'une base de données

- Avant de manipuler des données, il faut créer une ou des bases de données.



ETAPE 2 : sélectionnez le nom de la base à manipuler (le nombre de tables de la base apparaît entre parenthèses)

ETAPE 1 : Écrivez le nom de la base de donnée à créer. Puis cliquez sur « Créer »

3. Gestion de la base de données

forum sur le serveur localhost - phpMyAdmin 2.2.6 - Microsoft Internet Explorer

Echier Edition Affichage Favoris Outils 2

Accueil

forum (4)

forum

- mforum
- news
- test
- uforum

Table	Action	Enregistrements	Type	Taille
<input type="checkbox"/> mforum	Afficher Sélectionner Insérer Propriétés Supprimer Vider	42	MyISAM	10,7 Ko
<input type="checkbox"/> news	Afficher Sélectionner Insérer Propriétés Supprimer Vider	4	MyISAM	2,6 Ko
<input type="checkbox"/> test	Afficher Sélectionner Insérer Propriétés Supprimer Vider	7	MyISAM	2,1 Ko
<input type="checkbox"/> uforum	Afficher Sélectionner Insérer Propriétés Supprimer Vider	5	MyISAM	2,7 Ko
4 table(s)	Somme	58	--	18,1 Ko

Tout cocher / Tout décocher Pour la sélection :

- Version imprimable
- Exécuter une ou des requêtes sur la base forum [Documentation]

☒ Réafficher la requête après exécution

Où Emplacement du fichier texte :

Parcourir...

Exécuter

- Choix d'une table à gérer en particulier

Actions sur les tables : afficher leur contenu intégral, faire une sélection sur critères, ajouter des données, gérer ses propriétés intrinsèques, supprimer, vider.

Écrire une requête MySQL à exécuter

Exécuter une requête MySQL contenue dans un fichier

4. Affichage d'une table

Base de données *forum* - table *mforum* sur le serveur *localhost*

Affichage des enregistrements 3 - 5 (42 total)

requête SQL : [Modifier]

```
SELECT * FROM 'mforum' LIMIT 3,3
```

Rappel de la requête

Rappel de la base, de la table et du serveur

<< < Afficher : 3 lignes à partir de 5 > >>
en mode horizontal et répéter les en-têtes à chaque groupe de 100

		id	titre	mesg	hits	thread_idx	author_idx	date
Modifier	Effacer	4	Concours de logo AML 4	Vous êtes tous invités à participer au concours de...	2	0	1	2002-09-18 11:13:40
Modifier	Effacer	5	Concours de logo AML 5	Vous êtes tous invités à participer au concours de...	2	0	1	2002-09-18 11:13:40
Modifier	Effacer	6	Concours de logo AML 6	Vous êtes tous invités à participer au concours de...	0	0	1	2002-09-18 11:13:40

Colonnes = noms des attributs de la table

Liste des enregistrements de la table par pages de X lignes

Supprimer un enregistrement

Accès au formulaire de modification d'un enregistrement

Permet de naviguer dans les pages de résultats

Insertion d'un nouvel enregistrement

Afficher par page de X lignes

Technologie Web Avancée

Partie 3 : PHP/MYSQL

Le contenu de ce document est soumis à la Licence de Documentation Libre (**GNU Free Documentation License**).

Plan du Chapitre

- **1. Connexion**
- **2. Envoi de données**
- **3. Récupération des données**
- **4. Session**

1. CONNEXION

- × Pour se connecter à une base depuis PHP, il faut spécifier un nom de serveur, un nom d'utilisateur, un mot de passe et un nom de base de données.
- × Les fonctions de connexion :
 - + `mysqli_connect($server,$user,$password,$base)` : permet de se connecter au serveur `$server` en tant qu'utilisateur `$user` avec le mot de passe `$password` sur la base de données `$base`.

CONNEXION

Exemple :

```
<html>
<head>
</head>
<title> exemple d'application sur php </title>
```

```
<?php
```

```
$serveur    = "localhost";
$utilisateur = "root";
$motDePasse = "";
$base       = "gestion_etudiant";
```

```
$connect=mysqli_connect($serveur, $utilisateur , $motDePasse, $base );
```

2. ENVOI DES DONNEES

- ❖ Pour envoyer des données dans une base de donnée, il existe la fonction **mysql_query(\$str)** qui prend pour paramètre une chaîne de caractères qui contient la requête écrite en SQL et retourne un identificateur de résultat ou FALSE si échec.

Exemple :

```
/* récupération des données envoyés par le formulaire */
```

```
$num = $_POST["numero_filiere"];
```

```
$nom= $_POST["nom_filiere"];
```

```
/* Création de la requête */
```

```
mysql_query($connect, "INSERT INTO filiere ( numfiliere,nomfiliere)  
VALUES ('$num','$nom')");
```

```
?>
```

3. RECUPERATION DES DONNEES

- Pour récupérer des données à partir d'une base de donnée, il existe la fonction **mysqli_fetch_row(\$result)** qui retourne une ligne de résultat sous la forme d'un tableau. Les éléments du tableau étant les valeurs des attributs de la ligne. Retourne FALSE s'il n'y a plus aucune ligne.

Exemple :

```
/* Création de la requête */
$result = mysqli_query(connect,"SELECT * FROM filiere");

/* Parcours de la requête */
while($ligne = mysqli_fetch_row($result)) {
    $num_filiere = $ligne[0];
    $nom_filiere = $ligne[1];

/* affichage de la requête */
    echo "$num_filiere, $nom_filiere <br />";
}
?>
```

- Ici, on accède aux valeurs de la ligne par leur indice dans le tableau.

4.Session

- Une session est un mécanisme technique permettant de sauvegarder **temporairement sur le serveur** des informations relatives à un internaute.
- La sécurité et l'intégrité des données sont améliorées tout au long de la session.
- Contrairement à une base de données ou un système de fichiers, la session conserve les informations pendant quelques minutes.
- Les sessions sont particulièrement utilisées pour ce type d'applications :
 - Les espaces membres et accès sécurisés avec authentification.
 - Gestion d'un caddie sur un site de vente en ligne.

- Pour utiliser les sessions, différentes fonctions PHP s'offrent à nous. Voici déjà un petit tableau vous permettant de vous familiariser avec ces différentes fonctions (que nous détaillerons bien sur dans la suite de ce tutorial):

Fonction	Signification
session_start	Démarre une session
session_register	Enregistre une variable de session
session_unregister	Efface une variable de session
session_id	Retourne l'id de la session en cours
session_name	Retourne le nom de la session en cours
session_unset	Detruit toutes les variables de la session en cours
session_destroy	Detruit la session en cours

Exemple d'Application

Etape 1: Connexion

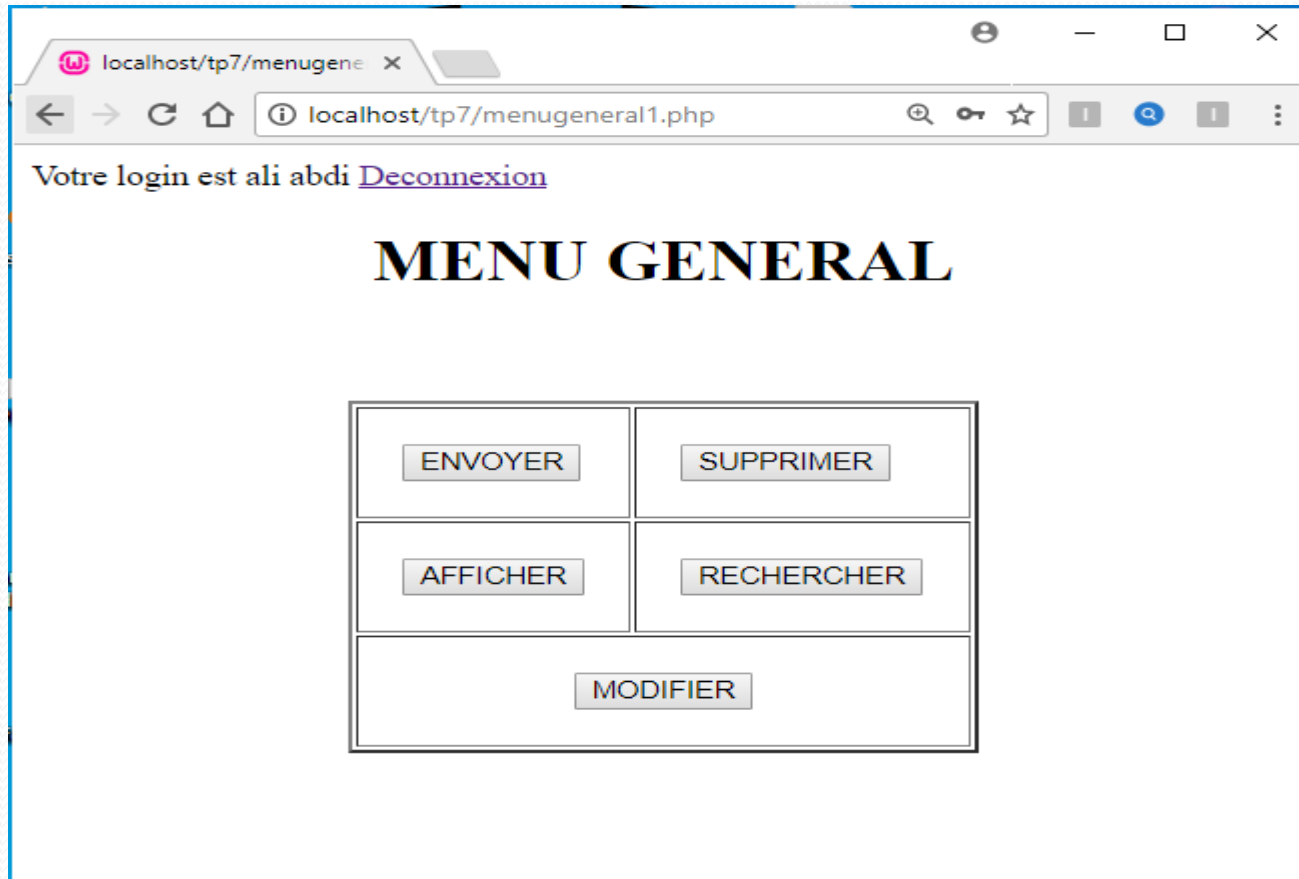
Connexion

Nom :	<input type="text"/>
password :	<input type="password"/>
	<input type="button" value="Annuler"/> <input type="button" value="Connexion"/>

Etape 1: Connexion

- // on démarre la session
`session_start ();`
- // on enregistre les paramètres de visiteur comme variables de session (\$login et \$pwd) (notez bien que l'on utilise pas le \$ pour enregistrer ces variables)
`$_SESSION['login'] = $_POST['nom'];`
`$_SESSION['pwd'] = $_POST['password'];`
- // on redirige le visiteur vers une page du section membre
`header("Location: menugeneral1.php");`

Etape 2: Menu General



Etape 2: Menu General

// On démarre la session (ceci est indispensable dans toutes les pages de notre section membre)

```
session_start ();
```

echo ‘

// On récupère nos variables de session

```
Votre login est '$_SESSION['login'].
```

Etape 3: DECONNEXION

```
<?php
// On démarre la session
    session_start ();

// On détruit les variables de notre session
    session_unset ();

// On détruit notre session
    session_destroy ();

// On redirige le visiteur vers la page d'accueil
    header ('location: connexion.html');
?>
```