Formation PHP/MySQL

PHP c'est quoi ?

C'est un langage client/serveur :

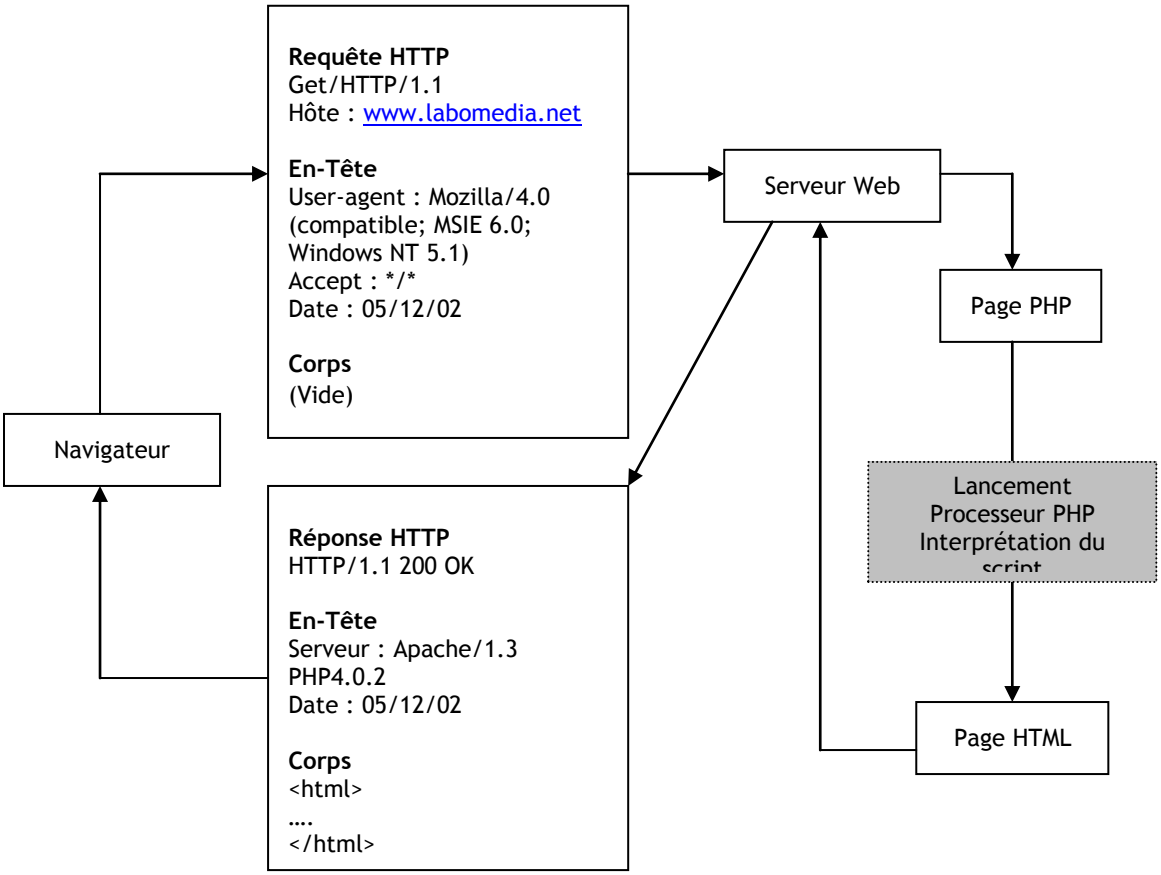
Votre navigateur, le client, interroge un serveur, en lui demandant une page d'accueil par exemple. Le serveur stocke sur son disque dur les pages php ou html qui seront demandées par les internautes. Pour que le serveur puisse vous répondre, il doit disposé d'un serveur web (APACHE, IIS,…), c'est-à-dire un programme permettant l'interprétation et la diffusion des pages demandées.

Votre navigateur demande la page, le serveur reçoit et comprend la requête, il recherche la page en question et si elle est disponible, il la renvoie. Mais il ne retourne que le code HTML.

Protocole de transport des données TCP/IP
Protocole d'attribution d'adresse permettant à TCP/IP d'acheminer les infos : http

C’est un langage de script :

Il fonctionne du côté serveur, il est en relation avec le langage HTML et permet de générer à la volée (dynamiquement) des pages web.
Un script PHP est encadré par des balises de début et de fin (< ?php … ?>). Ce sont ces balises qui permettent au serveur de savoir quand il doit interpréter ces informations comme du PHP et non du simple HTML.



L’interprétation du script peut être divisé en 2 parties :

- Tout d’abord, l’exactitude du script est vérifiée (PARSING), vérification par rapport à un ensemble de règles données.

- Ensuite, l’exécution du script, transformation en HTML. Cette opération s’effectue de façon linéaire sauf si le script mentionne un autre fonctionnement.

Les différentes versions de PHP :

Nous sommes à la version 4 du langage PHP. La compatibilité entre les différentes versions est assurée. Le langage a gagné en structure et fonctions au cours des différentes versions.

Une grande partie de la syntaxe est inspirée du PERL et du C.

Comparatifs – PHP face à ces concurrents :

PERL : un langage de script système alors que PHP est plus tourné vers le web. (Practical Extraction and Report Langague) très fort pour le parsing d'un document et la recherche d'expressions régulières.
PERL s'est adapté au web grâce à la mise en place de mod_perl et FAST CGI sur les serveurs afin d'utiliser des scripts PERL sur le web.
Beaucoup plus rigoureux et moins facilement maintenable.

ASP : active server pages. Langage de script développé par Microsoft. ASP et PHP sont très proches sur leur fonctionnement. Jusqu'à PHP3, ASP était plus attirant car nombreuses fonctions, librairies, accès aux bases de données Microsoft et sessions. Mais depuis PHP a largement comblé son retard. ASP est très dépendant des produits Microsoft (serveur IIS, base de données Access – SQL Serveur, ODBC) alors que PHP s'interface en natif avec de nombreuses bases de données et surtout sa mise en œuvre est gratuite.

JSP : java server pages. PHP a été conçu pour s'intégrer au code HTML tandis que JSP est une utilisation de Java de faire gérer des scripts intégrés au code HTML. JSP fait appel aux Beans (composants JAVA) puis génération d'une servlet. De plus, JSP permet de travailler dans une architecture n-tiers contrairement à PHP qui s'exécute dans un environnement 2-tiers.

COLDFUSION : relativement similaire à PHP. Solution propriétaire MACROMEDIA, langage de script serveur. Langage interprété. L'avantage principal étant les outils de développement de MACROMEDIA, rapidité de développement. L'inconvénient étant l'obligation d'utiliser COLDFUSION SERVER & STUDIO. COLDFUSION intègre son propre moteur de base de données.

Installation de PHP/MySQL/Apache/PHPMyAdmin :

EASYPHP
www.easyphp.org
www.manucorp.com

Détails de l'installation…

Configuration du fichier php.ini :

Voir fichier exemple : C:\Program Files\EasyPHP\www\formation_php

Présentation phpMyAdmin :

phpMyAdmin est un ensemble de script PHP permettant d'administrer une base de données à partir d'un navigateur web.

phpMyAdmin permet de :

- créer ou supprimer des bases de données
- créer ou supprimer des tables
- éditer, ajouter ou supprimer des champs
- exécuter des requêtes SQL
- de gérer les clés et index des champs

Une base de données (son abréviation est BD, en anglais DB, *database*) est une entité dans laquelle il est possible de stocker des données de façon structurée et avec le moins de redondance possible. Ces données doivent pouvoir être utilisées par des programmes, par des utilisateurs différents.

SQL (*Structured Query Language*, traduisez *Langage de requêtes structuré*) est un langage de définition de données (LDD, ou en anglais DDL *Data Definition Language*), un langage de manipulation de données (LMD, ou en anglais DML, *Data Manipulation Language*), et un langage de contrôle de données (LCD, ou en anglais DCL, *Data Control Language*), pour les bases de données relationnelles.

Le SQL est à la fois un langage de manipulation de données et un langage de définition de données. Toutefois, la définition de données est l'oeuvre de l'administrateur de la base de données, c'est pourquoi la plupart des personnes qui utilisent le langage SQL ne se servent que du langage de manipulation de données, permettant de sélectionner les données qui les intéressent.

Base de données formation - table utilisateur

```
# phpMyAdmin MySQL-Dump
# version 2.2.0rc4
# http://phpwizard.net/phpMyAdmin/
# http://phpmyadmin.sourceforge.net/ (download page)
#
# Serveur: localhost
# Généré le : December 5, 2002, 3:50 pm
# Version du serveur: 3.23.40
# Version de PHP: 4.0.6
# Base de données: `formation`
# --------------------------------------------------------

#
# Structure de la table `utilisateur`
#

CREATE TABLE utilisateur (
  pk_utilisateur int(11) NOT NULL auto_increment,
  nom varchar(50) NOT NULL default '',
  prenom varchar(50) NOT NULL default '',
  adresse varchar(255) NOT NULL default '',
  code_postal varchar(10) NOT NULL default '',
  ville varchar(100) NOT NULL default '',
  pays varchar(50) NOT NULL default '',
  login varchar(50) NOT NULL default '',
  password varchar(50) NOT NULL default '',
  email varchar(100) NOT NULL default '',
  telephone varchar(10) default NULL,
  maj_date datetime NOT NULL default '0000-00-00 00:00:00',
  PRIMARY KEY  (pk_utilisateur),
  UNIQUE KEY login (login),
  KEY nom (nom),
  KEY prenom (prenom)
) TYPE=MyISAM;
```

Séance n° 2 le 12/12/2002

Intégrer le code PHP au HTML

Le langage PHP est un langage de script qui s'insère dans les pages HTML et les parties PHP sont déclarées au moyen de balises reconnues par le serveur.

Balises	Remarque
< ?php … ?>	Ce type de balise est le plus courant et il est accepté par défaut par l'interpréteur.
<script language= »php »>…</script>	Ce type est également accepté par défaut par l'interpréteur
< ? … ?>	Ces balises courtes sont appelées « short tags ». Elles ne sont pas acceptées par défaut, cela fait partie de la configuration de PHP
<% … %>	Ces balises courtes sont appelées « short tags ». Elles ne sont pas acceptées par défaut, cela fait partie de la configuration de PHP

Les commentaires

Balises	Remarque
# commentaire	Le texte entre le signe # et la fin de la ligne sera en commentaire.
// commentaire	Le texte entre // et la fin de la ligne sera en commentaire
/* commentaire */	Tout le texte entre /* et */ sera en commentaire

Les constantes

PHP permet de définir des constantes, autrement dit des chaînes de caractères représentant une valeur figée. Elles se déclarent au moyen de DEFINE().

define(« LANGAGE », »PHP ») ;

En plus des constantes que vous définissez-vous même, PHP propose ses propres constantes qui sont pour certaines regroupés au sein de $HTTP_SERVER_VARS, $HTTP_ENV_VARS
ou avec la fonction phpinfo().

$HTTP_SERVER_VARS[REMOTE_ADDR]
$HTTP_SERVER_VARS[HTTP_USER_AGENT]
$HTTP_SERVER_VARS[HTTP_REFERER]

Les variables

Les variables sont destinées à stocker les données qui seront utilisées et pourront être modifiées lors de l'exécution du script. L'initialisation d'un variables crée une zone mémoire dans laquelle sera stockée la valeur (caractères, entier, décimal, …)

Les variables sont représentées avec le caractère $.
Pour déclarer une variable toto, il suffit de l'appeler $toto
PHP n'impose pas de fixer à l'avance le type de données contenues dans la variable contrairement à d'autres langages de programmation.

Le nom d'une variable ne peut pas commencer par un chiffre.
ATTENTION, les variables sont sensibles à la casse.

$toto != $TOTO ;

Les types de variables

Les 4 types de variables scalaires		
Type	**Description**	**Exemple**
int ou integer	le type int est utilisé pour contenir des entiers (nombres sans virgule)	$mavariable = 2 ; $mavariable = -4 ; $mavariable = 2002 ;
double ou float ou real	Le type double sert à définir des nombres décimaux, cad à virgule flottante	$mavariable = 1.0 ; $mavariable = -0.2 $mavariable = 3.1415972 ;
string	le type string définit une chaîne de caractères	$mavariable = « Vive PHP » ; $mavariable = « 1 » ;
boolean	Le type booléen définit une variable prenant des valeurs de type binaire : TRUE (vrai) ou FALSE (faux)	$mavariable = TRUE ; $mavariable = FALSE ;

Les Tableaux

Les variables, telles que nous les avons vues, ne permettent de stocker qu'une seule donnée à la fois. PHP propose des structures de données permettant de stocker l'ensemble de ces données dans une "variable commune". Ainsi, pour accéder à ces valeurs il suffit de parcourir la variable de type complexe composée de "variables" de type simple.

Les tableaux stockent des données sous forme de liste. Les données contenues dans la liste sont accessibles grâce à un index (un numéro représentant l'élément de la liste). Contrairement à des langages tels que le langage C, il est possible de stocker des éléments de types différents dans un même tableau.

Avec PHP, il n'est pas nécessaire de préciser la valeur de l'index lorsque l'on veut remplir un tableau, car il assigne la valeur 0 au premier élément (si le tableau est vide) et incrémente les indices suivants. De cette façon, il est facile de remplir un tableau avec des valeurs. Le code précédent est équivalent à :

PHP permet l'utilisation de chaînes de caractères au lieu de simples entiers pour définir les indices d'un tableau, on parle alors de tableaux associatifs. Cette façon de nommer les indices peut parfois être plus agréable à utiliser:

Tableau simple

```
$tableau[] = « Laurent » ;
$tableau[] = « Michel » ;
$tableau[] = « Benjamin » ;

$tableau = array(« Laurent », « Michel », « Benjamin ») ;
```

Tableau multidimensionnel

```
$tableau = array(
                array(« Laurent », « 11bis av Dauphine », « Orléans »),
                array(« Michel », « 234 route d'Orléans », « Olivet »)
                );
```

cf exemple array1.php

Tableau Associatif

```
$tableau = array(
                array   (
                        « Prénom »=>« Laurent »,
                        « Adresse »=> « 11bis av Dauphine »,
                        « Ville »=>« Orléans »
                        ),
                ) ;
```

Les 2 type composés		
Type	**Description**	**Exemple**
array	Désigne un type tableau qui représente un ensemble de valeurs	$mavar = array (« cle »=> « valeur »,) ; $mavar[1] = "top";

object	Désigne un type objet (variable possédant ses propres propriétés, attributs et méthodes)	class maClass{
		}
		$monobjet = new maClass() ;

Bien que le type de variables soit défini à leur initialisation, il est toujours possible de forcer une variable dans un type donné. C'est la fonction setType() ;

boolean setType($variable, $type)

$type = « string », « int », …

De nombreuses fonctions de PHP permettent de vérifier l'existence d'une variable et de son type. isSet(), unset(), empty(), is_bool(), is_int(), is_double(), is_string(), is_array()

Les opérateurs

ARITHMETIQUES	
Opérateur	**Description**
$a + $b	Somme de $a et $b
$a - $b	Différence de $a et $b
$a*$b	Produit de $a et $b
$a / $b	Quotient de $a et $b
$a % $b	Reste de la division de $a par $b (modulo)

LOGIQUE	
Opérateur	**Description**
$a and $b	VRAI si $a et $b sont vrais
$a && $b	VRAI si $a et $b sont vrais
$a or $b	VRAI si $a ou $b est vrai
$a || $b	VRAI si $a ou $b est vrai
!$a	VRAI si $a est faux

COMPARAISON		
Syntaxe	**Nom**	**Description**
$a == $b	Egal	Renvoie TRUE si $a est égal à $b
$a === $b	Identique	Renvoie TRUE si $a est égal à $b et si $a et $b sont du même type.
$a != $b	Différent	Renvoie TRUE si $a est différent de $b
$a !== $b	Non identique	Renvoie TRUE si $a est différent de $b ou de type différent
$a >= $b	Supérieur ou égal	
$a <= $b	Inférieur ou égal	

Les fonctions

PHP possède la possibilité de regrouper des portions de code sous la forme de fonctions ou procédures. Il est possible de se passer des fonctions et d'écrire son programme comme une suite d'instructions mises bout à bout. Mais il est préférable de se servir des fonctions afin de :

- Structurer votre code
- Plus simple pour débugger
- Travail en équipe plus facile
- Réutilisation de vos fonctions

```
< ?php

        function MyFonction(){

                echo « J'aime PHP » ;
        }

         echo MyFunction() ;
?>
```

La portée des variables

Comme nous venons de le voir une fonction n'est rien d'autre qu'un bout de code isolé du programme principal. Il est possible de définir des variables à l'intérieur des fonctions mais elles n'auront qu'une portée locale, cad qu'elles ne seront pas accessibles hors de la fonction. Sauf si elle est définie avec le mot GLOBAL.

Les arguments

Les arguments sont facultatifs, mais s'il n'y a pas d'argument, les parenthèses doivent rester présentes.

```
< ?php

        function MyFonction($phrase){

                echo « J'aime PHP $phrase» ;
        }

        $phrase = « comme le chocolat » ;

         MyFonction($phrase) ;
?>
```

Avec paramètres par défaut

```
< ?php

        function MyFonction($phrase= « comme la vanille »){

                echo « J'aime PHP $phrase» ;
        }

        MyFonction() ;
?>
```

Avec retour d'une valeur

```
< ?php

        function MyFonction($phrase){

                $citation =  « J'aime PHP $phrase» ;
                return $citation ;
        }

        $phrase = « comme le chocolat » ;

        echo MyFonction($phrase) ;
?>
```

Séance n° 3 le 17/12/2002

Accéder à une base de données avec PHP

La première opération consiste à se connecter à la base de données. Cette opération se réalise en précisant l'adresse du serveur de la base de données, le nom de la base de données, le nom d'utilisateur et son mot de passe.

```
$host = « localhost » ;
$base = « mabase » ;
$user = « root » ;
$pwd = «  » ;
```

Une fois connectée, vous récupérez un identifiant de connection valable jusqu'à la deconnexion ou la fin du script

```
// Connexion base de données
$link = mysql_connect($host,$user,$pwd)
or die("Désolé mais impossible de se connecter à $host : <br>".mysql_error());

// Sélection base de données
mysql_select_db($base,$link)
or die ("Désolé mais la base de données $base est introuvable : <br>".mysql_error());

// Requête SQL base de données
$sql      = "SELECT * FROM utilisateur";
$result = mysql_query($sql)
or die("Désolé mais la requête SQL a échoué : <br>".mysql_error());

$total            = mysql_num_rows($result);
$colonnes       = mysql_num_fields($result);

echo "<table width=100% cellpadding=5 cellspacing=0 border=1 class=text>\n";
echo "<tr>\n";

for($i=0;$i<$colonnes;$i++){
        echo "<th>".mysql_field_name($result,$i)."</th>\n";
}
echo "</tr>\n";

while($row = mysql_fetch_assoc($result)){
        echo "<tr>\n";
        for($i=0;$i<$colonnes;$i++){
                echo "<td>".$row[mysql_field_name($result,$i)]."</td>\n";
        }
        echo "</tr>\n";
}
echo "</table>\n";

mysql_close();
```

Fonctions PHP spécifiques à MySQL

mysql_connect
mysql_connect : Etablit une connexion (non persistante) avec le serveur de base de données
mysql_connect($nom_serveur, $utilisateur, $password)

Renvoi Identifiant de connexion ou FALSE en cas d'échec

mysql_select_db
mysql_select_db : Sélectionne une Base de données
mysql_select_db($base, $idConnexion)

Renvoi TRUE en cas de succès, FALSE sinon

* Identifiant de connexion au serveur de base de données tel que retourné par mysql_connect(). Par défaut, c'est l'identifiant de la dernière connexion qui est utilisé.

mysql_query
mysql_query : Exécute une requête SQL
mysql_query($requete,$idConnexion)

Renvoi Identifiant de requête SQL ou FALSE en cas de d'échec.

mysql_close
mysql_close : Met fin à la connexion à la base de données et libère les ressources associées.
mysql_close($idConnexion)

Renvoi TRUE si opération a été effectuée avec succès, NULL en cas d'échec

mysql_insert_id
mysql_insert_id : Retourne la valeur affectée au champ auto-incrémenté lors de la dernière requête INSERT.
mysql_insert_id($idConnexion) ;

Renvoi la valeur du champ

mysql_fetch_row
mysql_fetch_row : Retourne un enregistrement, retourné par une requête SQL, sous la forme d'un tableau indexé.
mysql_fetch_row($idResultat)

Renvoi un tableau indexé contenant autant d'éléments que de champs retournés par la requête SQL, FALSE s'il n'y a pas d'enregistrement à lire ou rien en cas de d'identifiant de résultat non valide.

mysql_fetch_array
mysql_fetch_array : Retourne un enregistrement, retourné par une requête SQL, sous la forme d'un tableau indexé et/ou associatif
mysql_fetch_array($idResultat, $mode)

$mode : Au choix l'une des constantes

- MYSQL_ASSOC : pour retourner un tableau associatif ou les clés sont les noms des champs
- MYSQL_NUM : pour retourner un tableau indexé
- MYSQL_BOTH : pour retourner un tableau à la fois indexé et associatif

Renvoi tableau indexé et/ou associatif selon le mode choisi, FALSE sinon

mysql_free_result
mysql_free_result : Libère les ressources allouées pour un résultat de requête
mysql_free_result($idResultat)

Renvoi TRUE en cas de succès, sinon FALSE

mysql_num_rows
mysql_num_rows : Retourne le nombre d'enregistrements retournés par la requête SQL
mysql_nul_rows($idResultat)

Renvoi le nombre d'enregistrements ou rien en cas d'identifiant de résultat non valide

mysql_affected_rows
mysql_affected_rows : Permet de retrouver le nombre de lignes affectées par la requête précédente.
mysql_affected_rows($idConnexion)

Renvoi le nombre de lignes affectées par la requête précédente.

mysql_errno
mysql_errno : Retourne le code d'erreur du dernier appel MySQL
mysql_errno($idConnexion)

Renvoi Code d'erreur ou 0 si la dernière opération s'est passée sans erreur

mysql_error
mysql_error : Retourne le message d'erreur associé au dernier appel MySQL.
mysql_error($idConnexion)

Renvoi message d'erreur ou une chaîne vide si la dernière opération s'est déroulée sans erreur

mysql_num_fields
mysql_num_fields : Retourne le nombre de champs d'un enregistrement
mysql_num_fields($idResultat)

Renvoi le nombre de champs retournés par la requête.

mysql_field_name
mysql_field_name : Retourne le nom des champs dans la table
mysql_field_name($idResultat,$index)

$index : Index du champ dont on souhaite connaître le nom, le nombre total peut être obtenu grâce à mysql_num_fields().

Renvoi nom du champ ou FALSE si l'index n'est pas valide.