PHP - MySQL

Vincent MAZENOD – IGE CNRS
Webmestre de www.cerdi.org
Centre d'Etude et de Recherche sur le Développement International

vmazenod@free.fr

Plan du cours

- Cours 1 Présentation de PHP
- Cours 2 Syntaxe de PHP
- Cours 3 Présentation et syntaxe de SQL
- Cours 4 Présentation de MySQL interfacé avec PHP
- Cours 5 Les cookies et les sessions en PHP
- Cours 6 PHP et le E-commerce
- Cours 7 Autres fonctionnalités de PHP

Présentation du langage PHP

Pages Web statiques :

–Actualisation difficile (FTP, template, …)

–Manque d'interactivité pour l'internaute

–Aucune automatisation possible pour les tâches d'administration (test des liens morts, gestion d'utilisateurs impossible, pas de template…)

Présentation du langage PHP

Pages web dynamiques :

– La page est générée en fonction des paramètres qui lui sont passés
– Interaction avec l'internaute
– Possibilité d'interaction avec les bases de données
– Réduction du nombre de pages sur le serveur
– Tâches de maintenances automatisées ou centralisés

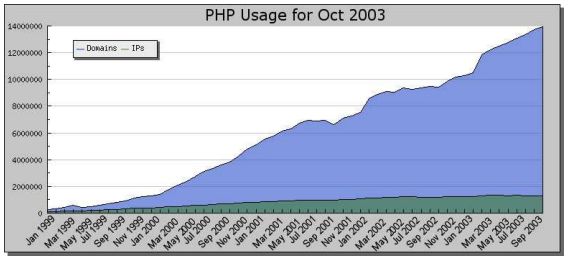
Généralités

- Ramsus Lerdof en 1994 conçoit PHP pour gérer sa page perso
- PHP4 – PHP5 Bêta
- PHP vient du monde UNIX
- Plateforme LAMP (Linux, Apache, MySQL et PHP)
- Supporté aussi par IIS, Netscape, Roxen, IPlanet …
- Damien Seguy  documentation FR

Généralités

- A base de Perl et de C
- Page PHP Traité coté Serveur – SERVER SIDE
- Page PHP = Programme qui génère du HTML
- L'internaute ne peut pas consulter le code source PHP (!= JavaScript - CLIENT SIDE)
- Interpréteur embarqué ou non au process du logiciel serveur.
- Langage moteur Zend
- PHP ⇔ ASP ⇔PHTML ⇔ZOPE
- Les Pages ont en général l'extension .php, ou .php3 ou .php4

Statistiques



Le langage

- Le langage s'insère dans le code HTML : PHP est de type embedded

```
<HTML>
<HEAD>
<TITLE>My First PHP Script</TITLE>
</HEAD>
<BODY>
    <P>
      <? print `` Hello World '';  ? >
    </P>
    <P>
      <?php echo `` nous sommes le'' . date(``d/m/y''); php?>
    </P>
</BODY>
</HTML>
```

fonctionnement

Le serveur exécute le code PHP entre les balises HTML, puis transmet le résultat au même titre qu'une page HTML statique

```
<HTML>
<BODY><? echo `` nous sommes le'' . date(``d/m/y''); ?></BODY>
</HTML>
```

Code source disponible pour l'internaute

```
<HTML>
<BODY> nous sommes le 2003-10-10</BODY>
</HTML>
```

Idem pour le code JavaScript

Pour exécuter à nouveau le script PHP il faut recharger la page!!!

Généralités Techniques

- Toutes les instructions se terminent par ;
- Les commentaires sont
  - // pour une ligne
  - /* … */ en multiligne
- TRUE (=1) et FALSE(=0) sont deux variables prédéfinies

Les Variables

- Les variables commencent par $ suivi d'une lettre ou d'un _
- Le typage est implicite, dépend de l'affectation.
- Fonctions de conversion

```
<?
        $MonEntier=3;
        $MonReel=1.23e-12;
        $MaChaine=« Hello world! »;
?>
```

Les Types

- Boolean : Booléen
- Integer : Entier
- Réel : Double
- String : Chaîne de caractères
- Array : Tableau
- Object : Objet

Fonctions sur les variables

- gettype($Var) : renvoie le type de $Var
- settype($var, ``Type'') : spécifie explicitement le type de $Var
- isset($Var) : TRUE si la variable est affectée, FALSE sinon.
- unset($Var) : détruit une variable et libère la mémoire qui lui était allouée
- empty($Var) : TRUE si la variable est non affectée ou =0 ou =``'' (chaîne vide)

Fonctions sur les variables

- is_int($Var) : TRUE si la variable est de type integer
- is_string($Var) : TRUE si la variable est de type string
- is_double($Var) : TRUE si la variable est de type double
- …

Les constantes

- define(`` var '',valeur) crée une constante var
- defined(``var'') : TRUE si var a été définie comme constante
- Les constantes ne sont pas précédées d'un $

```
<?
define(``BVN'',``Bienvenue sur ma Home Page'');
define(``Pi'',3.1415);
$Perimetre = 2 * $Rayon * Pi;
?>
```

Les références

- & permet de faire pointer des variables sur un même contenu.
- unset() sert aussi à détruire les références

```
<?
$foo = 'Pierre';// Assigne la valeur 'Pierre' $foo
$bar = &$foo;// Référence $foo avec $bar
$bar = "Mon nom est Pierre";// Modifie $bar...
echo $foo;// $foo est aussi modifie
unset($bar);//n'affecte pas $foo
?>
```

Les chaînes de caractères

- Pas de limite de taille
- Délimiteurs `` … ''

  Les variables contenues entre ces délimiteurs sont évaluées.

  \\ Antislash
  \$ caractère $
  \'' guillemet

Les chaînes de caractères

- Délimiteurs ` … '

  Les variables contenues entre ces délimiteurs ne sont pas évaluées.

  Le caractère antislash n'aura aucun effet

  Sauf

  \\ Antislash
  \' guillemet

Chaînes de caractères

- Caractères non imprimables

  \n nouvelle ligne
  \r retour à la ligne
  \t tabulation

Fonctions d'affichage

- echo () : écrit dans la navigateur
- print () : écrit dans la navigateur
- printf([$format,$arg1,$arg2]) : écriture formatée comme en C
- Print_r() : écriture des informations lisibles sur une variable complexe (tableau, objet)

Fonctions d'affichage

- echo ``Hello'';
- echo ``Hello $nom'';
- echo ``Hello ${nom}'';
- echo ``Hello''.$nom;
- Print(``Hello $nom'');
- Printf(``Hello %s'', $nom);

Les opérateurs

- Arithmétiques : +, -, * , /, %
- Affectations : =, +=, -=, *= , /=, %=, --, ++
- Logiques : &&, ||, and, or, xor, !
- Comparaisons : ==, <, >, =<, =>, !=, === (égalité des valeurs et des types)
- Concaténation : .

Les tableaux

- Constructeur : array()
  $langue=array(``FR'', ``EN'', ``DE'');
- Sans le constructeur
  $langue[]=``FR'';
  $langue[]=``EN'';
  $langue[]=``DE'';
  ou
  $langue[12]=``FR'';
  $langue[65]=``EN'';
  $langue[2]=``DE'';

Les tableaux associatifs

- Ce sont des tableaux indexés par des chaîne de caractères
- Constructeur : array()
  $langue=array(
  "FR" => "Français",
  "EN" => "English",
  "DE" => "Deutsch"
  );
- Sans le constructeur
  $langue["FR"] = "Français";
  $langue["EN"] = "English";
  $langue["DE"] = "Deutsch";

Les tableaux

- Par défaut les indices commencent à 0
- Le nombre d'éléments est le nombre d'éléments affectés
- Nombreuses fonctions disponibles : sizeof(), in_array(), array_pop(), array_push(), count()…

Parcours d'un tableau

```
For ($i = 0; $i <= count($langue); $i++){
   echo ``je parle le ''.$langue[$i].``<br>'';
}
```

Pas très propre dans le cas de remplissage discontinus…Ne marche que pour les tableaux indexés par des entiers :-(

Parcours d'un tableau

```
<SELECT name=``language''>
<?
   foreach( $langue as $elt ){
   echo ``<OPTION value=\``$elt\''>$elt``;
?>
  <OPTION>
<?}?>
</SELECT>
```

Permet de parcourir les éléments d'un tableau indexé indifféremment par des entiers ou par des chaînes de caractères.

Parcours d'un tableau

```
<SELECT name=``language''>
<?
   foreach( $langue as $key => $elt ){
   echo ``<OPTION value=\``$key\''>$elt``;
?>
  <OPTION>
<?}?>
</SELECT>
```

Permet de parcourir les éléments et les indexes d'un tableau indexé indifféremment par des entiers ou par des chaînes de caractères, sans les connaître a priori :-)

PHP – Structures de contrôle

- Les structures de contrôles en PHP ont une syntaxe similaire à celle du langage C
- Les blocs d'instructions associés aux structures de contrôle sont délimitées par des accolades :

  { … }

Structure conditionnelle

```
<?
$Cpt=12;
if( $Cpt > 0 && $Cpt < 4 ){
   echo $Cpt. “est dans l’intervalle ]0;4[“;
}
elseif( $Cpt >= 4 && $Cpt < 10){
   echo $Cpt. “est dans l’intervalle [4;10[“;
}
else{
   echo $Cpt. “est dans l’intervalle ]10;+∞)“;
}
?>
```

Structure de choix

```
<?
$Animal=‘chien’;
swicth($Animal){
   case ‘chien’ :
        echo " ouaf  ouaf ";
   break;
   case ‘chat’ :
        echo " miaou miaou ";
   break;
   default :
        echo $Animal." n’est pas un animal domestique ";
}
?>
```

Structure de boucle

```
<?
$NbDeCoup=6;
$i=0;
While($i != $NbDeCoup){
    $CoupAJouer = $NbDeCoup - $i;
   echo "il vous reste ". $CoupAJouer. " à jouer";
   jouer();
   $i++;
}
echo “la partie est finie“;
?>
```

Structure de boucle inversée

- Dans certains cas il peut être utile d'énoncer la condition de la boucle à la fin :

```
<?
$NbDeCoup=6;
$i=0;
do{
    $CoupAJouer = $NbDeCoup - $i;
   echo "il vous reste ". $CoupAJouer. " à jouer";
   jouer();
   $i++;
} While($i != $NbDeCoup);
echo “la partie est finie“;
?>
```

Structure de boucle itérative

```
<?
$tab=Array(2,4,8,9,0);
for ( $i = 1 ; $i <= 7 ; $i++ ){
   echo " élément numéro ".$i;
   echo “contient la valeur".$tab[i];
}
?>
```

Sortie forcée de boucle

- Break permet de quitter prématurément une boucle

```
<?
$i=0;
While( $nbr = $tab[$i++] ){
   echo $nbr. "<br>";
   if($nbr == "stop"){
        break;
   }
}
?>
```

Passage forcé à l'itération suivante

- Continue permet de passer à l'itération suivante d'une boucle sans exécuter le code qui suit cette instruction dans le bloc de la boucle

```
<?
$i=0;
While( $nbr = $tab[$i++] ){
   if($nbr == "Non Communiqué"){
        continue;
   }
   echo $nbr. "<br>";
}
?>
```

HTML & PHP – Ecriture du code

```
<?
echo " <SELECT name= \"language\" > ";
for( $i=0 ; $i <= 7; $i++ ){
   echo " <OPTION value=\"$tab[$i]\" >$tab[$i] ";
   echo " <OPTION> ";
}
   echo " </SELECT> ";
?>
```

Est syntaxiquement correct …

HTML & PHP – Ecriture du code

```
<SELECT name= "language" >
<?for( $i=0 ; $i <=7 ; $i++ ){?>
   <OPTION value= "<? echo "$tab[$i]";?>">
   <?= $tab[$i];?>
   <OPTION>
<?}?>
</SELECT>
```

Est beaucoup plus économique !

Les fonctions

- Les fonctions sont des portions de code (entre { … })associés à un nom
- Les fonctions peuvent prendre des arguments dont il n'est pas nécessaire de spécifier le type. La liste des arguments est entre (…) & séparés par des virgules
- function : indique le début d'une déclaration de fonction
- return : spécifie le résultat à renvoyer

Les fonctions

```
<?
   function somme ($a,$b){
        return $a + $b;
   }
?>
```

En PHP3 les fonctions devaient être absolument déclarées avant d'être appelées, PHP4 permet de placer la déclaration de la fonction n'importe où dans le code.

Les fonctions

- Appeler une fonction revient à écrire son nom suivi éventuellement des arguments à passer afin d'exécuter le code qu'elle contient

```
<?
$Total=234;
$SousTotal=123;
$Total=somme ($Total,$SousTotal);
echo $Total;// affiche 357
?>
```

Les fonctions

L'appel d'une fonction peut ne pas respecter son prototypage (Tous les arguments avec les quelles elle a été déclarée ne sont pas nécessaires)

```
<?
$Total=234;
$SousTotal=123;
$Total=somme ($SousTotal);
echo $Total;// affiche 123
?>
```

Les fonctions

- Les identificateurs de fonctions sont insensibles à la casse

```
<?
$Total=234;
$SousTotal=123;
$Total=sOmMe ($SousTotal, $Total);
echo $Total;// affiche 357 aussi
?>
```

Les fonctions

- Il est possible d'attribuer des valeurs par défaut aux arguments d'une fonction
- Les arguments ayant une valeur par défaut devraient être placés à la suite des arguments qui n'en ont pas
- Une valeur par défaut ne peut être qu'une constante!

Les fonctions

```
<?
function ServirApero($Contenant, $Boisson = "ricard"){
   return " Servir un ".$Contenant. " de ".$Boisson;
}
echo ServirApero("verre");
//Affiche Servir un verre de ricard
echo ServirApero("chaudron", "bourbon");
//Affiche Servir un chaudron de bourbon
?>
```

Les fonctions

```
<?
function ServirApero($Boisson = "ricard",$Contenant){
   return " Servir un ".$Contenant. " de ".$Boisson;
}
echo ServirApero("verre");
//Affiche Servir un de verre
?>
```

Portée d'un variable [locale]

- Une variable définie dans une fonction a une portée locale à la fonction
- Modifions l'exemple précédent

```
<?
        function somme ($a,$b){
                return $b = $a + $b;
        }
?>
```

$a et $b sont dites locales à la fonction somme

Portée d'une variable [locale]

```
<?
  $a=2;
  $b=4;
  echo somme ($a,$b);//Affiche 6
  echo somme ();//Affiche 0
  echo $a; //Affiche 2
  echo $b; //Affiche 4
?>
```

Portée d'un variable [globale]

- global : Permet de faire référence à une variable globale (visible dans tout le script)

```
<?
        function somme (){
                global $a, $b;
                return $b = $a + $b;
        }
?>
```

Portée d'une variable [globale]

- PHP met à disposition un tableau associatif $GLOBALS ayant pour élément toutes les variables globales du script

```
<?
//cette fonction est strictement
//équivalente à la précédente
  function somme (){
       $GLOBALS["b"]=$GLOBALS["a"]+$GLOBALS["b"];
  }
?>
```

Portée d'une variable [globale]

```
<?
  $a=2;
  $b=4;
  echo somme ();//Affiche 6
  echo $a; //Affiche 2
  echo $b; //Affiche 6
?>
```

Envoie de paramètres à une page

- Il existe deux méthodes pour passer des paramètres à une page web via le prtocole http:
  - La méthode GET
  - La méthode POST

La méthode GET

- Les variables et leurs valeurs transitent en clair dans l'url. → Visible dans la barre d'adresse du navigateur.
- La taille des variables est limitée
- Utilisable avec une balise <a href= " "> et une url de type :

  http://monsite.com/login.php?user=vmazenod&pass=bidon

- Utilisable avec un formulaire HTML (méthode par défaut) en spècifiant method="GET"

La méthode POST

- Les variables et leurs valeurs sont stockées dans l'en-tête de la requête HTTP→invisible au niveau du navigateur
- Taille illimitée
- Utilisable uniquement avec un formulaire en spécifiant method="POST"

Comment récupérer ces variables avec PHP?

- En les nommant !

```
<?
print $user;
//Affiche « vmazenod » dans login.php
?>
```

- En utilisant les tableaux associatifs mis à disposition par PHP:
  - $_POST[user]
  - $_GET[user]
- register globals=Off ⇔ $_POST[] ou $_GET[] est obligatoire

Inclusion de fichier

- require(fichier) inclut systématiquement le fichier
- include(fichier) inclut le fichier

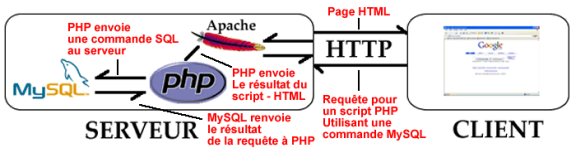
```
<?
    if($Besoin==TRUE){
        require(connexion.php);
        include(fonctions.php);
    }
?>
```

- Le fichier connexion.php est toujours inclus
- Le fichier fonctions.php est inclus si la condition est vérifiiée

Créer un site en PHP - MySQL

- Concevoir le schéma relationnel
  - Penser la structure de la base de données
  - Outil de modèlisation : MERISE, UML, modèle Entité – Association
- Créer la base sur le serveur MySQL
  - Commandes SQL au prompt ou via PHP
  - phpMyAdmin
- Créer l'interface Homme-machine
  - PHP pour récupérer et mettre à jour les données stockées dans la base de données MySQL
  - HTML pour mettre en forme les résultats d'une requête ou pour créer des formulaires de remplissage adaptés

Rappels



Connexion à une base de données

- Pour se connecter à un serveur MySQL il faut un nom de serveur, un utilisateur, un mot de passe, un nom de base de données

Int *mysql_connect*(string *server*, string *username*, string *password*)

- Permet à PHP d'utiliser MySQL avec les privilèges en lecture / écriture de l'utilisateur username
- L'entier que renvoie cette fonction est un identifieur de connexion

Sélection d'une base de données

- A priori le serveur MySQL peut gérer plusieurs bases de données, il faut donc en sélectionner une

Int *mysql_select_db*(string *database* [, id resource *identifier*])

- Sélectionne la base de données au niveau de PHP

Boolean *mysql_close*([id ressource *identifier*])

- Ferme la connexion au seveur MySQL

Code d'un connexion type

```
$server=''localhost'';
$user=''root'';
$pass=''s3cr3t'';
$base=''MaBase'';
If(mysql_connect($server,$user,$pass)){
   if(mysql_select_db($base)){
        echo « connexion réussie »;
        mysql_close();
   }
   else{die("Cette Base de données n'existe pas");}
}
else{ die("Echec de connexion au serveur!");}
```

Personnaliser ses messages d'erreur

- Le @ devant les fonctions mysql_connect() et mysql_select_db() empêche l'affichage des erreurs MySQL

@mysql_connnect($server,$user,$pass) or die (« connexion échouée »);

- Permet de cacher les messages d'erreurs MySQL en cas d'échec et de les remplacer par des messages personnalisés
- Après le die le script est interrompu

Connexion persistante

- Une connexion est dite persistante quand elle ne se ferme pas à la fin de chaque commande de PHP (économie de l'authentification)

Int *mysql_pconnect*(string *server*, string *username*, string *password*)

- Les connexions persistantes sont fermées à la fin de l'exécution du script
- L'entier renvoyé par cette fonction est aussi un identifiant de connexion

Envoyer un requête SQL au serveur

resource *mysql_query*(string *query* [, resource *link_identifier*])

- Envoie une requête SQL au serveur
- Renvoie un identificateur de résultat

```
$Requete=''SELECT * FROM Users
            WHERE Login=''\  $Login \'' '';
$Result= mysql_query($Requete) or die (''requete
  non valide! '');
```

Traiter le résultat avec PHP

Array *mysql_fetch_row* (resource *result_identifier*)

- Retourne une ligne du résultat de la requête sous la forme d'un tableau indexé par des entiers
- Prend en paramètre un identificateur de resultat

Traiter le résultat avec PHP

```
$requete=''SELECT Nom,Prenom FROM
  Personne'';
if($result=mysql_query($requete)){
      while($ligne=mysql_fetch_row($result)){
        echo ''Nom : ''.$ligne[0].''<br>'';
        echo ''Prénom : ''.$ligne[1] .''<br>'';
      }
}
```

Traiter le résultat avec PHP

Array *mysql_fetch_array* (ressource *result_identifier*)

- Retourne une ligne du résultat de la requête sous la forme d'un tableau indexé par les champs de la sélectionnés
- Prend en paramètre un identificateur de resultat

Traiter le résultat avec PHP

```
$requete=''SELECT * FROM Personne'';
if($result=mysql_query($requete)){
   while($ligne=mysql_fetch_array($result)){
      foreach($ligne as $champs => $valeur){
         Echo ''<p>$champs : $valeur</p>''
      }
   }
}
```

Fonctions additionnelles

Int *mysql_num_fields*(resource *result_identifier*)

- Retourne les nombres de champs du résultat

Int *mysql_num_rows*(resource *result_identifier*)

- Retourne les nombre de lignes du résultat

```
$result=mysql_query(SELECT * FROM Users);
$Nombredutilistateur=mysql_num_rows($result);
Echo ‘’Déjà $Nombredutilisateur utilisateurs!!’’;
```

Fonctions additionnelles

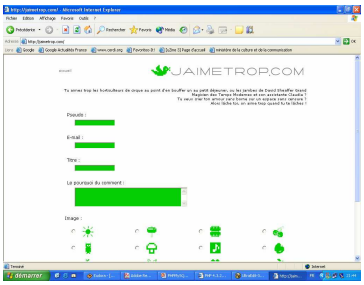
Int *mysql_affected_row*(resource *link_identifier*)

- Retourne le nombre de lignes affectées lors de la dernière commande INSERT, UPDATE, ou DELETE.
- agit comme mysql_num_rows pour un SELECT

Cas simples

- http://jaimetrop.com, http://jaimepas.com, http://bonnes.resolutions.free.fr

Cas simples



Bonnes.resolutions.free.fr

- Le Principe :
  - Une page qui liste toutes les bonnes résolutions de l’année courante
  - Une page qui permet de saisir une bonne résolution personnalisée
  - Une base de données simple qui stocke les bonnes résolutions et permet de les consulter

Création de la base

- Via phpMyAdmin, via mysql_query() au sein d’un script PHP, ou Via le prompt de commandes du serveur MySQL

```
CREATE TABLE Resolutions (
   Resolution text NOT NULL,
   Pseudo tinytext NOT NULL,
   Date date NOT NULL default '0000-00-00',
   Heure time NOT NULL default '00:00:00',
   Img tinytext NOT NULL );
```

Structure du script

```
<?
//Initialisations diverses
if(!isset($cmd)){?>//si aucune commande
    <a href= "index".php?cmd=FRM">prendre une bonne résolution</a>
    //Affiche la liste des bonnes résolutions de l'année
    <?if(isset($WRT)){//s' il faut insérer une nouvelle résolution
        //Vérifie que cette bonne résolution
        //ne contient pas de code malicieux et
        //n'a pas déjà été enregistré (refresh)
        //Sinon écrit le nouvelle résolution
    }
elseif($cmd == "FRM"){?>
    //Afficher le formulaire pour prendre une
    //bonne résolution personnalisée
<?}?>
```

Affichage de toutes les bonnes resolutions

- $anne_aff, si elle n'est pas définie, est initialisée à l'année courante

```
$sql= "SELECT
  Resolution,Pseudo,Img,Date,Heure
  FROM Resolutions
  WHERE YEAR(Date) = $annee_aff
  ORDER BY Date DESC,Heure DESC;";
```

Affichage des archives

```
<?
$sql= "SELECT DISTINCT YEAR(Date) FROM
   Resolutions ORDER by Date;";
$result = mysql_query ("$sql");
while($row = mysql_fetch_row ($result)){?>
<a href ="index.php?annee_aff=<? echo row[0];?>">
Les bonnes résolutions <? echo $row[0];?>
</a><br>
<?}?>
```

Le formulaire

```
<FORM action="index.php" method=post>
Pseudo:<br>
<INPUT type="text" name="Pseudo"><BR><br>
Ce qui va changer cette année:<br>
<TEXTAREA name="Resolution" colspan = "35" rowspan = "20"></TEXTAREA><BR>
<INPUT type="hidden" name="Date" value="<?print $date;?>"><BR>
<TABLE width=100%>
<TR>
    <TD width=33%>
    <input TYPE=radio name=Img value="1.gif"><img src="img/1.gif"><BR>
    <input TYPE=radio name=Img value="4.gif"><img src="img/4.gif"><BR>
    <input TYPE=radio name=Img value="7.gif"><img src="img/7.gif"><BR>
    <input TYPE=radio name=Img value="10.gif"><img src="img/10.gif"><BR>
    <input TYPE=radio name=Img value="13.gif"><img src="img/13.gif"><BR>
    </TD>
</TR>
<input type="submit" name="WRT" value="Je ne le referai plus :-)">
</TABLE>
</FORM>
```

Contrôle Des données

```
if (isset($WRT)){
    if(!isset($Img)||$Img==" "||$Img==""||
        !isset($Resolution)||$Resolution==" "||$Resolution==""||
        !isset($Pseudo)||$Pseudo==" "||$Pseudo==""){
        $cmd="FRM";
        $flag=1;
    }
    if(preg_match("/.*<.+>.*/",$Img,$parts)||
        preg_match("/.*>.+<.*/",$Img,$parts)||
        preg_match("/.*<.+>.*/",$Resolution,$parts)||
        preg_match("/.*>.+<.*/",$Resolution,$parts)||
        preg_match("/.*<.+>.*/",$Pseudo,$parts)||
        preg_match("/.*>.+<.*/",$Pseudo,$parts)){
        $cmd="FRM";
        $flag_=1;
    }
}
```

Contrôle dans le cas d'un refresh

```
$sql= "SELECT Resolution
      FROM Resolutions
      WHERE Resolution = \"$Resolution\"
      AND Pseudo = \"$Pseudo\"
      AND Img = \"$Img\";";
$result = mysql_query ("$sql");
If(mysql_num_rows($result)<1){
  //Insertion de la nouvelle résolution
}
```

Insertion de la bonne résolution dans la base de données

```
if(mysql_num_rows($result)<1){
    $sql= "INSERT INTO Resolutions SET Resolution =
    \"$Resolution\",Pseudo = \"$Pseudo\",Img =
    \"$Img\",Date = \"$Date\",Heure  = \"$Heure\" ;";
    $result = mysql_query ("$sql");
    if ($result){
          echo "Merci";
    }
    else{
          echo "Réessayez";
    }
}?>
<a href= "index.php"  >Voir votre bonne résolution</a>
```

Les stats

```
SELECT Annee, Pseudo,
  COUNT(Resolution) AS nbr
FROM citation
GROUP BY Annee, Pseudo
ORDER BY Annee DESC, Pseudo,
  Resoltution
```

Authentification

•Processus consistant à établir la légitimité ou la validité d'un utilisateur avant de permettre son accès à l'information demandée

•Authentification simple au sens informatique = 1 nom d'utilisateur unique + 1 mot de passe associé

•L'utilisateur authentifié ouvre une session

Authentification Web

- Beaucoup de protocole réseau font appel à une authentification (pop, ftp, ssh, …)
- Pour un site web l'authentification se fait
  - Au niveau du serveur web (fichiers .htacces dans apache)
  - Avec PHP (script de validation de l'authentification, ouverture d'une session, formulaire HTML de soumission login/mot de passe)

Script d'authentification

Soumission.htm

```
<FORM action="Authentification.php“ method= "POST “>
<INPUT Type=“text” name=“login”>
<INPUT Type=“Password” name=“pass”>
<INPUT Type=“submit”>
</FORM>
```

Authentifiaction.php

```
<?
if($login=“oim” && $pass=“B1gS3cR3t”){
   echo “Vous êtes authentifiés”;
}
else{
   echo “Vous n'êtes pas autorisés à voir la suite”;
}?>
```

Pourquoi?

- Pour que l'utilisateur puisse accéder à un service personnalisé (mail, banque en ligne, skin préférée, Panier d'achat …)
- Pour gérer un intranet (infos sur le web, mais réservé à un groupe)
- Pour permettre aux webmestres d'avoir accés à des pages de maintenance

Ou stocker les mots de passe?

- Information sensibles
  - Crypter les mots de passe
  - Contrôler tous les formulaires de saisies (code malicieux : Cross script)
  - Gestion par fichier texte ☹
    - Attention aux permissions système en lecture/écriture pour l'utilisateur web
    - Performance limitée pour des gros fichiers
  - Gestion via MySQL ☺
    - Attention au permission de l'utilisateur MySQL utilisé dans le script PHP

Cryptologie

- Crypter les données signifie les modifier selon un algorithme en vue de les décrypter selon un autre algorithme.
- Cryptage Basique dit de César
  - substituer les lettres du messages de la manière suivante:
    - A -> T
    - B -> U
    - C -> V
    - …

Cryptologie

- Cryptage a clé symétrique : une clé
  - Le message est brouillé en utilisant un mot que seul l'émetteur et le récepteur connaissent
- Cryptage à clé publique : Deux clés
- Chaque utilisateur possède une clé publique que tout le monde connaît
  - Elle sert à encrypter les données
- Chaque utilisateur possède une clé privée que lui seule connaît
  - Elle sert à décrypter les données encodées avec la clé publique.

Le Hachage

- Brouiller les données de manière irreversible (signature)
- Algos de Hachage basiques

Lettre	Code ASCII
S	83
E	69
C	67
R	82
E	69
T	84
	454

Lettre	Code ASCII	Position	Produit
S	83	1	83
E	69	2	138
C	67	3	201
R	82	4	328
E	69	5	345
T	84	6	504
			1569

Le Hachage

- Signatures toutes distinctes
- MD5 est algorithme de hachage
- *string* md5(*string* text) renvoie le hachage de text
- On compare les signatures:

```
<?mysql_query("select * from user where login=\"$login\";");
If($row=mysql_fetch_array){
    if($row['Pass']=md5($Pass)){
        echo "authentification réussi";
    }
    else{echo "Mauvais mot de passe";}
}
else{echo "Mauvais login";}?>
```

HTTP & PHP

- http://www.delorie.com/web/headers.html permet de visionner l'en tête HTTP de n'importe quelle page web

```
HTTP/1.0 302 Found
Location: http://www.google.fr/
Set-Cookie:
    PREF=ID=3c6634aa427f3a0a:TM=1069101214:LM=1069101214:S
    =sYXwFZOnnWC6D8EF;
expires=Sun, 17-Jan-2038 19:14:07 GMT;
path=/; domain=.google.com
Date: Mon, 17 Nov 2003 20:33:34 GMT
Content-Type: text/html
Server: GWS/2.1
 Connection: Keep-Alive
Content-Length: 151
```

HTTP & PHP

int header(*string* Entete)

- Permet d'envoyer des entêtes HTTP spécifiques

```
<?header("Location: http://cerdi.org");?>
```

Envoie un statut de redirection à Apache

boolean headers_sent()

- Retourne TRUE si les en-têtes ont été envoyé.

Session & PHP

- Une session dans le contexte de PHP est une collection de variables qui doivent se transmettre de page en page.
- On connaît déjà deux manières de faire passer des variables à un script PHP
  - La méthode GET nécessite que chaque lien mentionne les variables

```
<a href= "protect.php?login=maze&pass="s3cR3t"&article 1="CD"&….>
```

  - La méthode POST nécessite de créer un formulaire metnionnant toutes les variables pour chaque lien.

Les cookies

- Les cookies sont des fichiers textes simples stockés sur le client, où le serveur peut stocker des informations sur l'utilisateur et les retrouver d'un appel de page à l'autre.
- Mécanisme conçu par Netscape pour pallier à l' « amnésie » du protocole HTTP
- Chaîne de caractère de 4Ko maximum
- Maximum de 20 cookies par site
- Maximum de 300 cookies par navigateur

Envoyer un cookie via PHP

boolean setcookie(*string* nom [, *string* valeur, *int* expiration, *string* chemin, *string* domaine, *int* securise])

- nom : nom du cookie
- valeur : valeur du cookie
- expiration : date de suppression du cookie (format UNIX : temps écoulé depuis 01/01/70). Si non défini, supprimer quand on sort du site.
- chemin : répertoire du site ou le cookie est valide ('/' par défaut)
- domaine : domaine de validité du cookie (monsite.com)
- securise : valeur à 1 si le cookie est envoyé par HTTPS

Accéder à un cookie via PHP

- Les cookies font partie des en-têtes HTTP , ce qui impose que setcookie soit appelée avant tout affichage de texte. Ce sont les mêmes limitations que pour header .
- Directement par $NomDuCookie si le register_global=On dans le php.ini
- Par le tableau associatif $_COOKIE["NomDuCookie"] (PHP 4) ou $_HTTP_COOKIE_VARS["NomDuCookie"] (versions antérieures)

Cookie en PHP

```
<?
if(!isset($_COOKIE["BgColor"]){
   setcookie("BgColor","#000000",time()+3600*24)
}
//Cookie stocke la couleur préférée pendant 24h00
?>
<HTML>
<BODY>
   <TABLE bgcolor= "<?echo $_COOKIE["BgColor"];?> ">
        Mon texte
   </TABLE>
</BODY>
</HTML>
```

Session, PHP & Cookie

- Permet de passer des variables d'une page à une autre de manière simple
- Les navigateurs peuvent être configurés pour ne pas accepter les cookies
- «Voler» un cookie via javascript est peu complexe, donc problèmes de sécurités…
  - http://www.securityhack.net/?page=recup_cookie

Les sessions en PHP

- PHP permet de gérer des sessions au niveau du serveur pour une sécurité optimale
- Une Session en PHP c'est
  - Un identifiant de session (unique!)
  - Une collection de variable associée à cet identifiant de session
- PHP n'a qu'à faire passer l'identifiant de session d'un page à l'autre pour retrouver la collection de variables associée

Les sessions en PHP

- session.use_trans_sid=1 permet d'accoler automatiquement l'identifiant de session à l'URL d'un lien (solution peu sécurisée)
- Sinon les liens devront être de la forme
  <a href="test.php? <?echo SID;?>">
  Passe toutes les variables de la session à la page test.php
  - SID est une constante ayant pour valeur l'identifiant de sesion
- Une session fait partie de l'en-tête HTTP donc doit être appelé avant tout affichage de texte ….

Les sessions en PHP

- Register_global=On : (déconseillé)

boolean session_start()
- Crée une session ou reprend la session courante.

boolean session_register(Var1, Var2,…)
- Associe des variables à la session courante (illimité)

boolean session_unregister(Var1, Var2,…)
- Supprime des variables associées à la session courante

boolean session_is_registered(Var)
- Supprime des variables associées à la session courante

Les sessions en PHP

boolean session_unset()
- Supprime toutes les variables associées à la session courante

boolean session_destroy()
- Détruit la session courante. A NE PAS OUBLIER

- Les variables de session sont accessibles via leur nom $NomDeVariableDeSession

Les sessions en PHP

- Register_global=Off : (solution sécurisée)

session_start(), session_unset(), session_destroy() sont toujours valables

- Les variables globales sont accessibles depuis le tableau associatif $_SESSION["Var"] (PHP 4) $HTTP_SESSION_VARS ["Var"] (versions antérieures)

Formulaire d'authentification

```
<HTML>
<BODY>
<FORM action= "login.php" method="POST">
<INPUT type="text" name= "login">
<INPUT type="password" name= "pass">
<INPUT type="submit">
</BODY>
</HTML>
```

Login.php

```
<?
mysql_connect('localhost','root','s3cR3t') ;
Mysql_select_db('Admin');
SQL="select Password FROM Users Where Login=\"$_POST[Login]\";";
$result=mysql_query($SQL);
if($row=mysql_fetch_array($result)){
    if($row["Password"]==$pass){
            session_start();
            $_SESSION["login"]=$_POST["login"];
            $_SESSION["pass"]=$_POST["pass"];
            header('Location: ./membre.php?'.SID);
    }
    else{header('Location: ./index.html');}
}
else{header('Location: ./index.html');}
?>
```

Membre.php

```
<?session_start();?>
<HTML>
<HEAD><TITLE>page privée</TITLE></HEAD>
<BODY>
<?echo ”Bienvenue ”.$_SESSION[”login”]; ?>
<a href=”logout.php”>Se déconnecter</a>
</BODY>
</HTML>
```

Logout.php

```
<?
session_start();
session_unset();
session_destroy();
Header(‘Location: ./index.html’);
?>
```

Le e-commerce

- Le commerce électronique
  - Vente ou achat de biens ou de services, effectué par une entreprise, un particulier, une administration ou toute autre entité publique ou privée, et réalisé au moyen d'un réseau électronique
- Objectifs
  - L'entreprise peut conquérir de nouveaux marchés
  - Elle peut également élargir sa clientèle

Le e-commerce

- Les Produits
  - Permettre le parcours d'un catalogue d'articles
  - Classer les produits de manière à ce qu'ils puissent être retrouvés rapidement (ergonomie)
  - Proposer un détail pour chaque produit
  - Gérer les stocks, et les délais de livraisons associés pour chaque produit

Le e-commerce

- Les utilisateurs
  - Doivent être authentifiés pour personnaliser leur session
  - Doivent pouvoir créer et modifier un panier virtuel
  - Doivent être suivis d'une session à l'autre

Le e-commerce

- Système d'information liés aux sites de e-commerce
  - Système de gestion de l'inventaire
  - Système de gestion de commandes
  - Système de gestion de profils utilisateurs
  - Système de génération de rapport
  - Système de génération des livraisons

Système de gestion de l'inventaire

- Base de données produit
  - Nom
  - Description, détail
  - Prix
  - Quantité en stock
  - Quantité vendue
- Fonctionnalités
  - Ajout / modification de produits
  - Gestion des stocks

Système de gestion de profils utilisateurs

- Base de données client
  - Nom
  - Contact (adresse postale, e-mail, téléphone Fax)
  - Mode de paiement (chèque carte de crédit)
  - Historique des commandes

Système de gestion de profils utilisateurs

- Fonctionnalités
  - Ajout / modification d'un client (faite par le client lui-même)
  - Gestion des commandes antérieures, pour publicité ciblée, éventuellement réduction de fidélisation.
  - Système de « mot de passe oublié »

Système de gestion de commandes

- Une interface de commande/achat pour le client
- Un traitement automatisé de ces commandes
- Lié au système de gestion de l'inventaire pour le réapprovisionnement.
- Principe
  - Un utilisateur = 1 caddie
  - À la fin de la session l'utilisateur paie ce que contient son caddie
- Problème
  - Gestion simultanée de plusieurs caddies
  - Le caddie doit suivre l'utilisateur sur chaque page du site parcouru.

Système de gestion de commandes

- Le caddie
  - Nécessite l'utilisation des cookies ou des sessions
  - Doit pouvoir être incrémenté à tout moment
- Base de données
  - Id client
  - Date commande
  - Contenu de la commande
  - Date de livraison
  - Etat de la commande
  - Montant totale de la commande

Système de génération de rapport

- Analyse et présente les données de façon intelligente:
  - Totale des ventes sur une période donnée
  - Rentabilité de l'entreprise
  - Etude des comportements clients
  - Articles le plus / le moins vendu
  - Gestion de l'inventaire

Paiement

- Le Paiement sécurisé ne l'est pas toujours
- Toujours penser à une alternative (paiement par chèque, creditmail, minutepaid)
- Correspond à la fin de la session de l'internaute
- Stocker la commande, envoyer la commande à l'inventaire, générer une facture pour le client.

Paiement sécurisé

- Le paiement sécurisé est géré le plus souvent par un autre organisme, le plus souvent une banque.
- L'utilisateur ne saisi pas son numéro de carte bleue sur le site de e-commerce
- Possibilité d'assurer le système de paiement sécurisé en cas de fraude

PHP et JavaScript

- N'oubliez pas :
  - PHP est Server-Side
  - Javascript est Client-Side
- PHP génère peut donc générer du code Javascript au même titre que du code HTML!!
- La combinaison permet d'accroître l'ergonomie

JavaScript le debugger PHP ;-)

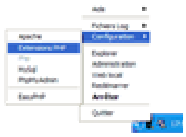
```
…
$SQL=«SELECT * FROM Article WHERE Id
   = $Article»;?>
<SCRIPT langage=« Javascript1.0 »>
<!—
alert(« <? echo $SQL;?>);
-->
</SCRIPT>
<?mysql_query($SQL);?>
…
```

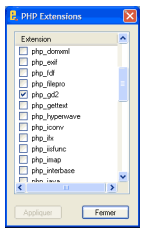
Remplir des tableaux JavaScript avec PHP

```
<SCRIPT LANGUAGE="Javascript">
<!--
<?
$LG_sql= "SELECT Id, Nom FROM groupe ORDER BY Nom;";
$result_LG = mysql_query ("$LG_sql");
$i=0;
while($row_LG = mysql_fetch_array ($result_LG)){
    $tmp_Nom = ereg_replace("'","\'",$row_LG["Nom"]);
    $i++;
?>
Groupe[<?echo $row_LG["Id"];?>]='<?echo $tmp_Nom;?>';
OrdreGroupe[<?echo $i;?>]=<?echo $row_LG["Id"];?>;
<?}?>
-->
</SCRIPT>
```

Modules supplémentaires

- A la main :
  - Enlever le commentaire sur les lignes correspondant au module dans php.ini
  - Sous windows 1module = 1 DLL
  - Apache doit être relancé
- Avec EasyPHP :





GD : Génération dynamique d'images

- Module permettant de créer des images à base de commandes PHP
- L'image est envoyée au navigateur
- Cet envoie doit être précédé d'un Header annonçant le type MIME de l'image:
  - header('Content-type: image/png');
  - header('Content-type: image/jpg');
  - …
- Le fichier php ne fait que générer l'image!

Exemple avec GD

```
// Création de l'image 100 px * 100 px
$image = imagecreate(100, 100);

// Définition des couleurs

$white    = imagecolorallocate($image, 0xFF, 0xFF, 0xFF);
$gray     = imagecolorallocate($image, 0xC0, 0xC0, 0xC0);
$darkgray = imagecolorallocate($image, 0x90, 0x90, 0x90);
$navy     = imagecolorallocate($image, 0x00, 0x00, 0x80);
$darknavy = imagecolorallocate($image, 0x00, 0x00, 0x50);
$red      = imagecolorallocate($image, 0xFF, 0x00, 0x00);
$darkred  = imagecolorallocate($image, 0x90, 0x00, 0x00);
```

Exemple avec GD

```
// Superpostion de 10 ellipses ombrées

for ($i = 60; $i > 50; $i--) {
  imagefilledarc($image, 50, $i, 100, 50, 0, 45, $darknavy,
     IMG_ARC_PIE);
  imagefilledarc($image, 50, $i, 100, 50, 45, 75 , $darkgray,
     IMG_ARC_PIE);
  imagefilledarc($image, 50, $i, 100, 50, 75, 360 , $darkred,
     IMG_ARC_PIE);
}

// Ellipse avec couleurs non grisées

imagefilledarc($image, 50, 50, 100, 50, 0, 45, $navy, IMG_ARC_PIE);
imagefilledarc($image, 50, 50, 100, 50, 45, 75 , $gray, IMG_ARC_PIE);
imagefilledarc($image, 50, 50, 100, 50, 75, 360 , $red, IMG_ARC_PIE);
```

Exemple avec GD

```
// Envoie de l'image

header('Content-type: image/png');
imagepng($image);
imagedestroy($image);
```

Le Résultat :-o

PDF : génération de documents PDF à la volée

- Permet d'écrire du texte ou de dessiner dans un fichier PDF
- Le fichier créé peut être stocké sur le serveur et/ou envoyé au client
- Intéressant pour toute forme de rapport imprimable : PDF est un standard
- Module CPDF propriétaire

Exemple avec PDF

- Créer un document et l'ouvrir

```
$pdf = pdf_new();
pdf_open_file($pdf, "");
```

- Quelques paramètres sur le fichier

```
pdf_set_parameter($pdf, "warning", "true");
pdf_set_info($pdf, "Creator", "index.php");
pdf_set_info($pdf, "Author", "MaZe->");
pdf_set_info($pdf, "Title", "My First PDF");
```

- Commencer une page

```
pdf_begin_page($pdf, 595, 842); //Format A4
```

Exemple avec PDF

- Ecrire un texte :

```
pdf_set_font($pdf, "Times-Roman", 30, "host");
pdf_set_value($pdf, "textrendering", 1);
pdf_show_xy($pdf, "Le permier PDF de : ", 50, 750);
```

- Tracer une ligne :

```
pdf_moveto($pdf, 330, 740);
pdf_lineto($pdf, 550, 740);
pdf_stroke($pdf);
```

- Envoyer le fichier au client :

```
pdf_close($pdf);
$buf = pdf_get_buffer($pdf);
$len = strlen($buf);
header("Content-type: application/pdf");
header("Content-Length: $len");
header("Content-Disposition: inline; filename=foo.pdf");
```

Et une fois de plus …

C bo :-o