PHP pour le Web dynamique

Mikhaël Janson

26 janvier 2000

Introduction

- Le Web dynamique : qu’est-ce que c’est ?
- Différentes solutions
- Comparaison des outils disponibles

Pages HTML statiques

- Temps de développement initial court
- Modification du contenu a la main
- Contenu important : mises à jour longues
- Aucune interaction avec le visiteur
- Résultat “pauvre”

Pages Web dynamiques

- Travail de programmation requis au départ
- Mise à jour du contenu automatique
- Interaction avec le visiteur
- Interfaçage avec SGBD

Solutions pour le Web dynamique

- Exécutables CGI
- Active Server Pages
- Java Server Pages
- Allaire Cold Fusion
- Everyware Tango Entrerprise
- PHP
- etc.

Exécutables CGI

- Programmes externes aux pages HTML appelés par un lien
- Choix du langage : Perl, C, C++, Shell Script, VB
- Exécutables autonomes ou
  scripts dépendant d'un interpréteur
- Risque de sécurité
- Plusieurs étapes nécessaires pour tester
- Pas de modification directe dans la page Web

Active Server Pages

Avantages :

- Variables de session
- Langage relativement souple, variables sans dćlaration préalable
- Composants ActiveX très nombreux
- Interaction ActiveX - système

Inconvénients :

- Syntaxe VBScript parfois obscure
- Problèmes de sécurité sur le serveur IIS de WinNT

ColdFusion

Avantages :

- Variables de session
- Très pratique pour les débutants
- Code extrêmement simple
- Environnement de développement ColdFusion Studio : gestion visuelle du dynamique

Inconvénients :

- Déclaration des variables avant utilisation : vérifications fastidieuses
- Limitations dûes à la simplicité : beaucoup de code nécessaire pour les contourner

PHP3

Avantages :

- Variables sans déclaration préalable
- Relativement souple
- Syntaxe se rapprochant du C (lisibilité)
- Interfaces avec une mulititude de SGBD
- Multiplateformes – portabilité
- Open Source !!
- Création d'images et graphiques à la volée
- Génération de fichiers PDF
- Rapidité – etc. etc.

Inconvénients :

- Pas de variables de session (dans la v. 3)

Caractéristiques

- Syntaxe facilement accessible aux programmeurs initiés à Perl, C ou Java
- Accès facile à Perl, C ou Java après avoir appris PHP
- SGBD supportés :
  dBase, DBM, FilePro, IMAP, LDAP, mSQL, MySQL, ODBC, Oracle, Postgres, SNMP, Sybase, Velocis
- Plate-formes : UNIX, Win98, WinNT, Macintosh
- Serveurs : Apache, Xitami – IIS, PWS, Netscape (avec ISAPI et NSAPI), etc.
- Open Source, évolutif, ajout de fonctionnalités, support
- Orientation Web :
  Communication avec le serveur Web facilitée

Fonctionnement :

- Le serveur Web reçoit la demande HTTP
- Il parcourt le fichier appelé et en extrait les instructions PHP
- Le serveur transmet les instructions PHP à l'interpréteur php
- Traitement des instructions par l'interpréteur
- Renvoi du résultat au serveur Web qui l'insère dans la page envoyée au visiteur

Les programmes php3 ne sont pas transmis au client, mais exécutés sur le serveur.

Pas de configuration particulière requise pour le navigateur : il ne reçoit que du code HTML.

Inclure des instructions php :

```
<HTML>
<HEAD>
<TITLE>Quel jour sommes-nous ?</TITLE>
</HEAD>

<BODY>
<H1>Quel jour sommes-nous ?</H1>

<?

/* La balise "<?" ci-dessus indique
qu'on ouvre le script PHP */
$aujourdhui = date("Y-m-d");
print "<CENTER>La date du jour est:  ";
print "$aujourdhui.</CENTER>";

# pour refermer le script
?>
</BODY>
</HTML>
```

Inclure un fichier externe :

```
<HTML>
<HEAD>
<TITLE>Quel jour sommes-nous ?</TITLE>
</HEAD>

<BODY>
<H1>Quel jour sommes-nous ?</H1>

<?
$aujourdhui = date("Y-m-d");
print "<CENTER>La date du jour est:  ";
print "$aujourdhui.</CENTER>";

?>
Merci d'avoir lu ce texte.

<?  include("pied.txt")?>



</BODY>
</HTML>
```

Formulaires HTML et variables :

Un formulaire pour recueillir les données

```
<HTML> <HEAD>
<TITLE>Demande d'informations</TITLE>
</HEAD> <BODY>
Souhaitez-vous plus d'informations sur notre équipe ?
<P>
<TABLE WIDTH=400><TR><TD align = right>
<FORM ACTION="email.php3" METHOD="POST">
  Votre nom :<BR>
  <INPUT TYPE="text" NAME="nom" SIZE="20" MAXLENGTH="30">
  <P>
  Votre adresse e-mail :<BR>
  <INPUT TYPE="text" NAME="nom" SIZE="20" MAXLENGTH="30">
  <P>
  Je trouve votre site :
  <SELECT NAME="opinion">
    <OPTION value=excellent>excellent
    <OPTION value=arevoir>à revoir
  </SELECT>
  <P>
  <INPUT TYPE="submit" VALUE="Envoyer">
</FORM>
</TD></TR></TABLE></CENTER>
</BODY>
</HTML>
```

Formulaires HTML et variables

Le script qui récupère les donnees saisies

Les arguments reçus du formulaire sont parcourus et pré-affectés à des variables automatiquement accessibles ici.
On a donc :

- $nom=[le nom fourni par le visiteur]
- $email=[l'adresse fournie par le visiteur]
- $opinion=[le choix sélectionne par le visiteur]

```
<?  /* ce script sert  traiter les variables transmises
par le fichier demandeinfo.html */
PRINT "<CENTER>";
PRINT "Bonjour, $nom.";
PRINT "<BR>";
PRINT "Merci d'avoir visité notre site.<BR>";
PRINT "Un e-mail est envoyé à $email, pour dire que vous
trouvez ce site $opinion.";
PRINT "</CENTER>";
?>
```

On informe le visiteur qu'un courrier lui a été envoyé.
Les informations saisies dans le formulaires n'ont pas été sauvegardés. Les variables qui les contiennent disparaitront dès la fin de l'exécution de ce script.

Formulaires HTML et variables

Le script qui exécute les actions

Le script ne se contentera pas d'envoyer des informations au navigateur du visiteur.
Il va nous permettre ici d'exécuter automatiquement des commandes telles que la commande MAIL.

```
<?  /* ce script sert  traiter les variables transmises
par le fichier demandeinfo.html */
print "<CENTER>";
print "Bonjour, $nom.";
print "<BR>";
print "Merci d'avoir visité notre site.<BR>";
print "Un e-mail est envoyé à $email, pour dire que vous
trouvez ce site $opinion.";
print "</CENTER>";

mail( "$email",
      "Votre visite sur linux-gull.ch",
      "Bonjour $nom.  Merci de votre visite sur notre site.
      Nous allons rendre ce site encore plus $opinion.
      Vous trouverez toujours nos informationss sur :
      http://www.linux-gull.ch");

mail( "admin@linux-gull.ch",
      "opinion sur le site",
      "$name a visite le site et l'a trouve $opinion.
      son adresse est $email.");
?>
```

Listes et itération

Initialisation d'une liste

```
<?  $cantons = array(
    "Valais",
    "Genève",
    "Vaud",
    "Fribourg",
    "Neuchâtel",
    );
print("Cantons romands avant 1979 :<br>");
print($cantons);
print("<br>");
```

Ajout à une liste

```
$cantons[] = "Jura";
print("Cantons romands après 1979 :<br>");
print($cantons);
print("<br>");
```

Récupération d'elements par index

```
print("Le 5e canton de ma liste est" .
$cantons[4]);
?>
```

Listes et itération

Liste indexée par chaines de caractères

```
<?  $cantons = array(
    "VS"=>"Valais",
    "GE"=>"Genève",
    "VD"=>"Vaud",
    "FR"=>"Fribourg",
    "NE"=>"Neuchâtel",
    "JU"=>"Jura"
    );
print("Le code VS correspond à" .  $cantons["VS"]);
```

Boucle conditionnelle

```
print("<table>");

while( $uncanton = next($cantons) ){
   $code = key($uncanton);
   $nom = $cantons[$code];
   print("<tr><td>" .  $code
        .  "</td><td>" .  $nom
        .  "</td></tr>");

}

   print("</table>");

?>
```

Fonctions

Utilisation de sous-routines

```
<?  $authentification = array (
    "gilbert"=> "coucou",
    "erik"=> "hello",
    "cedric"=> "bouh",
    );
function authentifier($qui, $secret) {
   global $authentification;
   if ($secret == $authentification[$qui]) {
      return 1;
   } else {
      return 0;
}

function saluer($nom) {
   print "Bonjour $nom et bienvenue sur notre site";
}

if ( authentifier($nom, $password) == 0 ) {
   print "Mot de passe erroné";
} else {
    saluer($nom);
}

?>
```

Interface avec SGBD

Extraire des données d'une base

```
<?
```

Identifiants de connexion : possibilité de les stocker dans un fichier protégé.

```
include("../include/database.inc");
```

Etablir une connexion avec le serveur de BDD

```
mysql_connect($machine, $utilisateur, $secret);
```

Choisir une base sur le serveur

```
mysql_select_db("catalogue");
```

Préparer une requête SQL

```
$query = "SELECT * FROM articles";
```

Exécuter la requête et récupérer le résultat dans un tableau

```
$resultat = mysql_query($query);
```

Compter le nombre de lignes du tableau

```
$lignes = mysql_num_rows($resultat);
```

Pour chaque ligne, afficher les informations qu'elle contient – stopper a la dernière ligne

```
for($compteur = 0; $compteur < $lignes; $compteur++) {
   $NomArt = mysql_result($resultat, $compteur, "nom");
   $DescrArt = mysql_result($resultat, $compteur, "descr");

   print("<b>$NomArt</b><br>$DescArt<br><br>");
}
```

N.B. : les valeurs des champs peuvent être récupérées plus efficacement avec
mysql_fetch_row *ou* mysql_fetch_array*. Voir documentation.*

Terminer la connexion

```
mysql_close();
```

```
?>
```

Aperçu de la syntaxe pour programmeurs

Quelques aspects similaires à Perl

- **conversion automatique de type de données**
- **expressions rationnelles (regexp)**

Quelques déclarations similaires à C

- **if {} elseif {} else {}**
- **switch**
- **while(debut, condition, FinBoucle) {}**
- **break**
- **do {...} while(debut, condition, FinBoucle)**

Références

Internet

http://passtech.org/php/php.html

http://www.commentcamarche.net/php/phpintro.htm

http://www.phpindex.com/

http://tecfa.unige.ch/guides/php/welcome.html

http://doc.ctrlaltdel.ch/html/php/francais/manual.html

http://www.php.net

http://www.phpbuilder.com

http://www.devshed.com

Papier

Programmation PHP (en français)
Leon Atkinson, 1999 – ed. Campus Presse

Core PHP Programming
Leon Atkinson, 1999

Professional PHP Programming
Castagnetto, et al, 1999

Php Pocket Reference
Rasmus Lerdorf, 2000

Practical Php
Chad Robinson, Grame Merrall

Remerciements

Pour m'avoir fait découvrir et aidé a progresser en php, je tiens à remercier particulièrement :

François Deppierraz (francois@ctrlaltdel.ch)
Daniel Muller (daniel@ctrlaltdel.ch)
Alexandre Kraft (safj@infomaniak.ch)