pages Web Dynamiques
Côté serveur, Langage PHP

Rappels sur les notions du web

- HTML : conception de pages destinées à être publiées sur Internet
- Page html : contient le texte à afficher et des instructions de mise en page
- HTML est un langage de description de page et non pas un langage de programmation
  - pas d'instructions de calcul ou pour faire des traitements suivant des conditions
- Des sites de plus en plus riches en informations
  - Nécessité croissante d'améliorer le contenu de sites
  - Mises à jour manuelles trop complexes
    - Pourquoi ne pas automatiser les mises à jour ?

Rappels sur les notions du web

Pages web statiques : fonctionnement

- Leurs contenus ne changent ni en fonction du demandeur ni en fonction d'autres paramètres éventuellement inclus dans la requête adressée au serveur. Toujours le même résultat.
  - Rôle du serveur : localiser le fichier correspondant au document demandé et répond au navigateur en lui envoyant le contenu de ce fichier

Pages web statiques : limites

- Besoin de réponses spécifiques : passage de pages statiques à pages dynamiques

Rappels sur les notions du web

- Les langages de script-serveur : Définition
  - Un langage de script -serveur est :
    - un programme stocké sur un serveur et exécuté par celui-ci,
    - qui passe en revue les lignes d'un fichier source pour en modifier une partie du contenu,
    - avant de renvoyer à l'appelant (un navigateur par exemple) le résultat du traitement.
  - La tâche d'interprétation des ordres à exécuter est déléguée à un composant, souvent appelé moteur,
    - installé sur le serveur,
    - qui est doté d'une API et d'un fonctionnement identique quel que soit la plate-forme utilisée pour gérer le serveur

Rappels sur les notions du web

- Pages web dynamiques côté serveur ou côté client
  - **Langage côté client :** traité par la machine qui accueille le logiciel de navigation.
    - Ses résultats peuvent varier en fonction de plate-forme utilisée. Un programme en JavaScript pourra fonctionner sous Netscape et poser problème sous Internet explorer.
    - Les résultats peuvent être différents suivant la machine (PC, Mac)
    - Nécessité de tests importants
    - Ne permettent pas de masquer les sources du programme
    - Sont indépendants du serveur et donc de l'hébergement

Rappels sur les notions du web

- Les langages de création de pages web dynamiques côté serveur
  - **Les CGI**
    - Sont des composants exécutables (fichiers .exe ou .dll) qui produisent sur le serveur des contenus html à envoyer aux clients.
    - Les CGI sont compilés. Ils sont rapides mais fortement liés à la plate-forme sur laquelle ils tournent.
    - PERL
      - Surcharge rapide du serveur par la création de plusieurs processus
      - Employé sur de nombreux serveurs. Il tourne sur de nombreuses plateformes : Unix, Linux, Windows, Mac
      - Prévu à l'origine pour la manipulation de chaînes de caractères, il est rapidement devenu un véritable langage orienté objet.
      - Abord difficile et faible lisibilité.

Rappels sur les notions du web

- Les langages de création de pages web dynamiques côté serveur
  - **ASP**
    - Basé sur des scripts écrits en VBscript, Jscript ou Javascript.
    - Largement répandu,
    - Facilité de mise en œuvre
    - Plusieurs outils de développement intégrés (Macromédia Ultradev, Microsoft Visual Interdev).
    - Intimement liée à l'environnement Windows NT/2000 et au serveur IIS (Internet Information Server) de Microsoft.
      - L'environnement Microsoft est nécessaire

Rappels sur les notions du web

- Les langages de création de pages web dynamiques côté serveur
  - **JSP**
    - Constitue la réponse de Sun aux ASP de Microsoft
    - Utilisation de Java
      - Au départ simple extension du langage Java
      - Est devenu un véritable langage de développement web
    - Possède une interface de qualité
    - Lenteur relative

PHP

- Les langages de création de page web dynamiques côté serveur
  - **PHP**
    - Connaît un succès toujours croissant sur le Web et se positionne comme un rival important pour ASP
    - L'environnement Linux est sa plateforme de prédilection
    - Combiné avec le serveur Web Apache et la base de données MySQL, PHP offre une solution particulièrement robuste, stable et efficace
    - Gratuité : Tous les logiciels sont issus du monde des logiciels libres (Open Source).

Histoire et Origine

- PHP : Hypertext PreProcessor
- Première version de PHP a été mis au point au début d'automne par Rasmus Lerdorf en 1994
  - Version appelée à l'époque Personal Home Pages
  - Pour conserver la trace des utilisateurs venant consulter son CV sur son site, grâce à l'accès à une base de données par l'intermédiaire de requêtes SQL
- La version 3.0 de PHP fut disponible le 6 juin 1998
- A la fin de l'année 1999, une version bêta de PHP, baptisée PHP4 est apparue
- En 2001 cinq millions de domaines utilisent PHP
  - trois fois plus que l'année 2000

Définition

- Un langage de scripts permettant la création d'applications Web
- Indépendant de la plate-forme utilisée puisqu'il est exécuté côté serveur et non côté client.
- La syntaxe du langage provient de celles du langage C, du Perl et de Java.
- Ses principaux atouts sont:
  - La gratuité et la disponibilité du code source (PHP4 est distribué sous licence GNU GPL)
  - La simplicité d'écriture de scripts
  - La possibilité d'inclure le script PHP au sein d'une page HTML
  - La simplicité d'interfaçage avec des bases de données
  - L'intégration au sein de nombreux serveurs web (Apache, Microsoft IIS, ...)

Intégration PHP et HTML (1)

- Principe
  - Les scripts PHP sont généralement intégrés dans le code d'un document HTML
  - L'intégration nécessite l'utilisation de balises
    - avec le style xml : <? ligne de code PHP ?>
    - Avec le style php: <?php ligne de code PHP ?>
    - avec le style JavaScript :

      <script language=«php»> ligne de code PHP </script>
    - avec le style des ASP : <% ligne de code ASP %>

Intégration PHP et HTML (2)

- Forme d'une page PHP
  - Intégration directe

```
< ?php
//ligne de code PHP
?>
<html>
<head> <title> Mon script PHP
</title> </head>
<body>
//ligne de code HTML
< ?php
//ligne de code PHP
?>
//ligne de code HTML
….
</body> </html>
```

Forme d'une page PHP

- Inclure un fichier PHP dans un fichier HTML : include()

Fichier Prinipal

```
<html>
<head>
<title> Fichier d'appel </title>
</head>
<body>
<?php
$salut = " BONJOUR" ;
include "information.inc" ;
?>
</body>
</html>
```

Fichier à inclure : information

```
<?php
$chaine=$salut. " , C'est PHP "
echo "  <table border= \"3"
 <tr> <td width = " 100%\ " >
  <h2> $chaine</h2>
  </td> </tr></table> ";
?>
```

Syntaxe de base : Introduction

- Typologie
  - Toute instruction se termine par un point-virgule
  - Sensible à la casse
    - Sauf par rapport aux fonctions
- Les commentaires
  - /* Voici un commentaire! */
  - // un commentaire sur une ligne

Syntaxe de base : Les constantes

- Les constantes
  - **Define**("nom_constante", valeur_constante)
    - define (“ma_const”, “Vive PHP4”) ;
    - define (“an”, 2002) ;
  - Les constantes prédéfinies
    - NULL
    - _FILE_
    - _LINE_
    - PHP_VERSION
    - PHP_OS
    - TRUE et FALSE
    - E_ERROR

Syntaxe de base : Les variables (1)

- Principe
  - Commencent par le caractère $
  - N'ont pas besoin d'être déclarées
- Fonctions de vérifications de variables
  - Doubleval(), empty(), gettype(), intval(),
  - is_array(), is_bool(), is_double(), is_float(), is_int(), is_integer, is_long(), is_object(), is_real(), is_numeric(), is_string()
  - Isset(), settype(), strval(), unset()
- Affectation par valeur et par référence
  - Affectation par valeur : $b=$a
  - Affectation par (référence) variable : $c = &$a

Syntaxe de base : Les variables(2)

- Visibilité des variables
  - Variable locale
    - Visible uniquement à l'intérieur d'un contexte d'utilisation
  - Variable globale
    - Visible dans tout le script
    - Utilisation de l'instruction global() dans des contextes locales

```
<?
$var = 100;
function test(){
global $var;
return $var;}
$resultat = test();
if ($resultat) echo $resultat; else echo " erreur ";
?>
```

Syntaxe de base : Les variables(3)

- Les variables dynamiques
  - Permettent d'affecter un nom différent à une autre variable

```
$nom_variable = 'nom_var';
$$nom_variable = valeur; // équivaut à $nom_var = valeur;
```

  - Les variables tableaux sont également capables de supporter les noms dynamiques

```
$nom_variable = array("val0", "val1", ...,"valN");
${$nom_variable[0]} = valeur; $val0 = valeur;
$nom_variable = "nom_var";
${$nom_variable}[0] = valeur;
$nom_var[0] = valeur;
```

  - Les accolades servent aussi à éviter toute confusion lors du rendu d'une variable dynamique

```
echo "Nom : $nom_variable - Valeur : ${$nom_variable}";
// équivaut à echo "Nom : $nom_variable - Valeur :
  $nom_var";
```

Syntaxe de base : Les variables (4)

- Variables prédéfinies
  - Les variables d'environnement dépendant du client

Variable	Description
$_SERVER["HTTP_HOST"]	Nom d'hôte de la machine du client (associée à l'adresse IP)
$_SERVER["HTTP_REFERER"]	URL de la page qui a appelé le script PHP
$_SERVER["HTTP_ACCEPT_LANGUAGE"]	Langue utilisée par le serveur (par défaut en-us)
$_SERVER["HTTP_ACCEPT"]	Types MIME reconnus par le serveur (séparés par des virgules)
$_SERVER["CONTENT_TYPE"]	Type de données contenu présent dans le corps de la requête. Il s'agit du type MIME des données
$_SERVER["REMOTE_ADDR"]	L'adresse IP du client appelant le script CGI
$_SERVER["PHP_SELF"]	Nom du script PHP

Syntaxe de base : Les variables (5)

- Variables prédéfinies
  - Les variables d'environnement dépendant du serveur

Variable	Description
$_SERVER["SERVER_NAME"]	Le nom du serveur
$_SERVER["HTTP_HOST"]	Nom de domaine du serveur
$_SERVER["SERVER_ADDR"]	Adresse IP du serveur
$_SERVER["SERVER_PROTOCOL"]	Nom et version du protocole utilisé pour envoyer la requête au script PHP
$_SERVER["DATE_GMT"]	Date actuelle au format GMT
$_SERVER["DATE_LOCAL"]	Date actuelle au format local
$_SERVER["$DOCUMENT_ROOT"]	Racine des documents Web sur le serveur

Syntaxe de base : Les variables (6)

- Variables prédéfinies
  - Affichage des variables d'environnement
    - la fonction phpinfo()
      - <? phpinfo(); ?>
      - **echo phpinfo(constante);**

INFO_CONFIGURATION	affiche les informations de configuration.
INFO_CREDITS	affiche les informations sur les auteurs du module PHP
INFO_ENVIRONMENT	affiche les variables d'environnement.
INFO_GENERAL	affiche les informations sur la version de PHP.
INFO_LICENSE	affiche la licence GNU Public
INFO_MODULES	affiche les informations sur les modules associés à PHP
INFO_VARIABLES	affiche les variables PHP prédéfinies.

    - la fonction getenv()
      - <? echo getenv("HTTP_USER_AGENT");?>

Syntaxe de base : Les types de données

Principe

- Pas besoin d'affecter un type à une variable avant de l'utiliser
  - La même variable peut changer de type en cours de script
  - Les variables issues de l'envoi des données d'un formulaire sont du type string

Les différents types de données

- Les entiers : le type Integer
- Les flottants : le type Double
- Les tableaux : le type array
- Les chaînes de caractères : le type string
- Les objets

Syntaxe de base : Les types de données

Le transtypage

- La fonction settype() permet de convertir le type auquel appartient une variable

```
<?        $nbre=10;
   Settype($nbre, " double ");
   Echo " la variable $nbre est de type " , gettype($nbre); ?>
```

- Transtypage explicite : le cast
  - (int), (integer) ; (real), (double), (float); (string); (array); (object)

```
<? $var=" 100 FRF ";
   Echo " pour commencer, le type de la variable est $var, gettype($var);
   $var =(double) $var;
   Echo <br> Après le cast, le type de la variable est $var ", gettype($var);
   Echo "<br> et a la valeur $var ";   ?>
```

Détermination du type de données

- Gettype(), Is_long(), Is_double(), Is_string(), Is_array(), Is_object(), Is_bool()

Syntaxe de base : Les chaînes de caractères(1)

Principe

- Peuvent être constituées de n'importe quel caractère alphanumérique et de ponctuation, y compris les caractères spéciaux

```
\tLa nouvelle monnaie unique, l' €uro, est enfin là...\n\r
```

- Une chaîne de caractères doit être toujours entourée par des guillemets simples (')ou doubles (")

" Ceci est une chaîne de caractères valide."

'Ceci est une chaîne de caractères valide.'

"Ceci est une chaîne de caractères invalide.'

- Des caractères spéciaux à insérer directement dans le texte, permettent de créer directement certains effets comme des césures de lignes

Car	Code ASCII	Code hex	Description
\car			échappe un caractère spécifique.
" "	32	0x20	un espace simple.
\t	9	0x09	tabulation horizontale
\n	13	0x0D	nouvelle ligne
\r	10	0x0A	retour à chariot
\0	0	0x00	caractère NUL
\v	11	0x0B	tabulation verticale

Syntaxe de base : Les chaînes de caractères(2)

- Quelques fonctions de manipulation

chaîne_result = addCSlashes(chaîne, liste_caractères);
ajoute des slashs dans une chaîne

chaîne_result = addSlashes(chaîne);
ajoute un slash devant tous les caractères spéciaux.

chaîne_result = chop(chaîne);
supprime les espaces blancs en fin de chaîne.

caractère = chr(nombre);
retourne un caractère en mode ASCII

chaîne_result = crypt(chaîne [, chaîne_code])
code une chaîne avec une base de codage.

echo expression_chaîne;
affiche à l'écran une ou plusieurs chaînes de caractères.

$tableau = explode(délimiteur, chaîne);
scinde une chaîne en fragments à l'aide d'un délimiteur et retourne un tableau.

Syntaxe de base : Les opérateurs

- Les opérateurs de calcul

Opérateur	Dénomination	Effet	Exemple	Résultat
+	opérateur d'addition	Ajoute deux valeurs	$x+3	10
-	opérateur de soustraction	Soustrait deux valeurs	$x-3	4
*	opérateur de multiplication	Multiplie deux valeurs	$x*3	21
/	plus opérateur de division	Divise deux valeurs	$x/3	2.3333333
=	opérateur d'affectation	Affecte une valeur à une variable	$x=3	Met la valeur 3 dans la variable $x

Syntaxe de base : Les opérateurs

- Les opérateurs d'assignation

Opérateur	Effet
+=	addition deux valeurs et stocke le résultat dans la variable (à gauche)
-=	soustrait deux valeurs et stocke le résultat dans la variable
*=	multiplie deux valeurs et stocke le résultat dans la variable
/=	divise deux valeurs et stocke le résultat dans la variable
%=	donne le reste de la division deux valeurs et stocke le résultat dans la vari
|=	Effectue un OU logique entre deux valeurs et stocke le résultat dans la var
^=	Effectue un OU exclusif entre deux valeurs et stocke le résultat dans la var
&=	Effectue un Et logique entre deux valeurs et stocke le résultat dans la vari
.=	Concatène deux chaînes et stocke le résultat dans la variable

Syntaxe de base : Les opérateurs

- Les opérateurs d'incrémentation

Opérateur	Dénomination	Effet	Syntaxe	Résultat (avec x valant 7)
++	Incrémentation	Augmente d'une unité la variable	$x++	8
--	Décrémentation	Diminue d'une unité la variable	$x--	6

- Les opérateurs de comparaison

Opérateur	Dénomination	Effet	Exemple	Résultat
==	opérateur d'égalité	Compare deux valeurs et vérifie leur égalité	$x==3	Retourne 1 si $X est égal à 3, sinon 0
<	opérateur d'infériorité stricte	Vérifie qu'une variable est strictement inférieure à une valeur	$x<3	Retourne 1 si $X est inférieur à 3, sinon 0
<=	opérateur d'infériorité	Vérifie qu'une variable est inférieure ou égale à une valeur	$x<=3	Retourne 1 si $X est inférieur à 3, sinon 0
>	opérateur de supériorité stricte	Vérifie qu'une variable est strictement supérieure à une valeur	$x>3	Retourne 1 si $X est supérieur à 3, sinon 0
>=	opérateur de supériorité	Vérifie qu'une variable est supérieure ou égale à une valeur	$x>=3	Retourne 1 si $X est supérieur ou égal à 3, sinon 0
!=	opérateur de différence	Vérifie qu'une variable est différente d'une valeur	$x!=3	Retourne 1 si $X est différent de 3, sinon 0