

PHP - Ateliers Web

Présentation

L'Internet a évolué ces dernières années. Les internautes sont de plus en plus exigeants : ils veulent des sites attractifs animés, accessibles, bien présentés, de navigation facile et mis à jour sur l'heure. Des normes apparaissent pour que ces exigences soient de mieux en mieux prises en compte, notamment celle de l'accessibilité aux personnes handicapées et c'est une bonne chose. On parle de Web 2 pour désigner les techniques qui se sont fait jour pour accompagner cette évolution.

Parmi ces techniques, nous citerons :

- XHTML, langage de balises, très peu différent de HTML qu'il perfectionne en étant beaucoup plus rigoureux ;
- les CSS qui permettent toutes les spécifications de mise en forme des pages Web ;
- AJAX, technique permettant de rendre les pages réactives à des demandes de l'internaute ;
- les CMS (Content Management System), systèmes permettant de créer et de mettre à jour un site en ligne, ce qui évite les délais d'envoi des fichiers.

Pour cela, les langages de spécification HTML/XHTML et CSS doivent être épaulés par de la véritable programmation. Du côté visiteur, on utilise principalement JavaScript ; du côté serveur, nous nous focaliserons sur PHP qui est le plus universellement offert par les hébergeurs. Un aspect très important abordé dans ce livre sera la question de l'équilibre entre ce qui est traité sur la machine du visiteur et ce qui l'est sur le serveur.

Ce livre décrit ces techniques de façon pratique, par l'exemple. Les principaux chapitres sont intitulés « atelier » ; ils traitent complètement un exemple d'application, mettant en jeu une des méthodes de résolution à connaître et correspondant aux besoins les plus fréquents des webmasters. Dans certains cas, différentes variantes sont confrontées et comparées.

Quelques autres chapitres sont intitulés « cours » : ils servent à l'indispensable apprentissage des éléments à connaître. Nous avons essayé de les réduire au « minimum vital ». De fait, le principal sujet de ce livre est l'apprentissage du

langage PHP, accompagné des techniques de mise en œuvre dans les contextes où il est le plus couramment appliqué.

Les derniers ateliers montrent la construction d'un CMS rudimentaire : il n'est pas question de concurrencer les CMS élaborés comme SPIP, mais simplement de montrer un exemple de bonne programmation PHP.

Ce livre suppose connus les éléments de HTML et JavaScript, tels qu'ils sont exposés, entre autres, dans D-J David : *Développer un site Web*, Editions Ellipses.

Conventions typographiques

Les citations d'instructions (X)HTML, JavaScript ou PHP dans le texte sont ainsi :

<html>.

Les extraits de programme sont sur fond grisé :

```
$x='Bienvenue sur le site' ;  
echo "<h1>".$x."</h1>\n" ;
```

Dans les chapitres de cours, les marches à suivre, exercices ou manipulations sont

signalés par  . Les exemples sont signalés par  ou par  .

Dans les ateliers, le but poursuivi est indiqué par :  **Mission :**

Les compléments d'information sont signalés par  tandis que le travail proprement-dit de l'atelier est signalé par  .

Introduction

► Les différents langages du Web

Dans la tendance actuelle de la conception des sites Web, on cherche à séparer le mieux possible le rôle joué par les différents composants. Mais de quoi est faite une page Web ? Eh bien un fichier page véhicule trois éléments :

- une information ; c'est l'essentiel du contenu que le visiteur est venu chercher sur Internet ; on doit assurer qu'il le trouve dans tous les cas ;
- la mise en forme (ou la mise en page) de cette information ;
- des comportements de la page Web comme des animations, la réponse à des actions du visiteur, ou la réaction à des réponses qu'il donne à un formulaire etc.

La fourniture de l'information est dévolue au langage HTML ou XHTML. Ce langage offre aussi des commandes de mise en forme, mais on conseille d'essayer de les éviter et de grouper les spécifications de présentation dans un fichier à part, rédigé le langage dit « des CSS » (Cascaded Style Sheets, feuilles de style en cascade).

Les comportements, eux, nécessitent de la programmation dès qu'ils sont un peu élaborés. Mais il y a deux lieux où un programme peut s'exécuter : sur l'ordinateur de l'internaute ou sur le serveur qui héberge le site. Il y a des arguments en faveur et en défaveur des deux hypothèses.

1. Après tout, il est normal que celui qui profite de quelque-chose fasse les efforts nécessaires, donc que le programme s'exécute chez le visiteur, puisque c'est pour lui qu'il est appelé.
2. Comme le serveur est, en moyenne, beaucoup plus performant que l'ordinateur de l'internaute, il est peut-être plus efficace qu'il s'exécute chez l'hébergeur. Cet argument est à modérer par le raisonnement suivant : supposons que le serveur soit k fois plus puissant en moyenne que l'ordinateur d'un visiteur ; mais le serveur doit répartir son temps entre, disons, n visiteurs simultanés ; alors selon que k/n est supérieur ou inférieur à 1, la balance penchera en faveur du serveur ou du visiteur.
3. Selon l'action à effectuer, le choix est parfois imposé. Une animation sera programmée sur la machine du visiteur, on dit « en local ». Au contraire, certaines opérations ne peuvent se faire que sur le serveur : tout ce qui est gestion d'une base de données (par exemple pour du e-commerce) ne peut être traité que sur le serveur ; la prise en compte finale des réponses à des

formulaires s'impose sur le serveur, de même que la mise à jour en ligne du site.

4. De fait, le plus efficace est de trouver un équilibre entre les deux solutions. Le critère à observer est de minimiser le trafic sur le réseau, donc d'éviter les échanges d'informations inutiles et de proscrire les traitements de données erronées sur le serveur. Par exemple, pour les réponses à un formulaire, on sait que le traitement final aura lieu sur le serveur, mais il est judicieux d'implanter un prétraitement chez l'internaute pour ne pas envoyer des réponses incomplètes ou incohérentes qui seraient de toutes façons rejetées.

► Les langages en local

Sur l'ordinateur du visiteur, les langages qui peuvent être utilisés sont JavaScript, VBScript et les applets Java. VBScript a l'inconvénient de n'être utilisable qu'avec le navigateur Internet Explorer. JavaScript est compris par tous les navigateurs, même s'il y a des différences d'interprétation que nous signalerons au fur et à mesure. C'est JavaScript qui est utilisé côté serveur pour Ajax, raison de plus pour que nous le considérons dans ce livre.

► Les langages côté serveur

Côté serveur, les langages sont nombreux. A côté de C qui a l'inconvénient d'exiger de résider dans un répertoire imposé CGI-BIN, on a développé, entre autres, ASP (qui ressemble à Visual Basic), Pearl et surtout PHP qui est l'objet principal de ce livre. Nous le choisissons pour sa simplicité qui a fait son succès universel et aussi parce-que, à la différence d'autres, il est le plus facilement admis par les hébergeurs. Il faut que vous vérifiez, au moment de prendre un contrat d'hébergement, que le langage que vous souhaitez utiliser sur votre site est autorisé par votre hébergeur. En effet un hébergeur peut interdire l'exécution de programmes sur ses ordinateurs pour raison de sécurité. De ce point de vue, c'est PHP qui a le plus de chances d'être accepté. Après avoir signifié Personal Home Page, le sigle PHP a pris la signification récursive PHP Hypertext Preprocessor.

► Fichier .php - Interaction avec HTML

► Appel d'un fichier PHP

Les instructions PHP (on dit qu'elles forment un script) sont contenues dans un fichier d'extension .php. Un tel fichier est appelé depuis un fichier HTML exactement comme s'il était un fichier .htm donc par un lien (), par une instruction JavaScript (comme location.href='xxx.php') ou comme valeur du paramètre action d'un formulaire (<form action="xxx.php" ...>).

Cette dernière forme est très souvent utilisée puisqu'un programme PHP est très bien adapté à la prise en compte des réponses à un formulaire. Il y a encore d'autres formes que nous verrons à l'occasion : partout où on peut spécifier une URL, celle-ci peut indiquer un fichier .php.

Vous pouvez aussi appeler directement un fichier .php en tapant son URL dans la zone adresse de votre navigateur.

► Structure d'un fichier PHP

Tout programme PHP doit être compris entre deux balises, une balise de départ et une balise de fin. Quatre couples différents sont utilisables, mais, dans ce livre nous n'utiliserons que le couple `<?php` (ouvrante), `?>` (fermante). Il est inutile de vous surcharger la mémoire avec les autres : deux ne sont pas toujours utilisables et le 3^e couple est plus long à écrire.

Voici donc un premier programme PHP :

```
<?php
echo "<h1>Bonjour !</h1>";
?>
```

L'instruction `echo` a pour rôle d'envoyer le texte `<h1>Bonjour !</h1>` sur la sortie standard. `echo` peut aussi se comporter comme une fonction auquel cas l'argument serait entre parenthèses : `echo (".....") ;`

► Mécanisme fondamental de PHP

Lorsque le navigateur de l'internaute appelle un fichier .php, il s'attend à recevoir une page Web comme s'il avait appelé un fichier .htm. Mais la réaction du serveur est différente : s'il s'agit d'un fichier .htm, le serveur l'envoie tout simplement. Pour un fichier .php, et justement au vu de l'extension .php, le serveur va interpréter et exécuter le fichier.

Les instructions du programme PHP vont être de deux sortes :

- des actions sur des fichiers sur un disque de l'hébergeur (ce sont précisément ces actions que JavaScript ne permet pas et qui nécessitent un programme sur le serveur) ;
- des écritures sur la sortie standard (c'est le cas de `echo` vue ci-dessus). Ces données vont être envoyées vers l'ordinateur de l'internaute comme si elles étaient un fichier .htm et le navigateur va les afficher comme page Web en réponse à sa demande.

Dans notre exemple ci-dessus, on voit que le texte créé contient une balise HTML : elle sera obéie lorsque le navigateur interprètera le texte.

PHP a deux manières de « fabriquer » du HTML :

- à la volée en écrivant du texte par `echo` ;
- de façon plus durable en créant un fichier `.htm` sur le serveur. Ceci est la source de la mise à jour en ligne des sites. Donc PHP est utilisable pour créer des CMS et il est utilisé effectivement par des CMS célèbres comme SPIP.

■ *Mélange avec HTML*

Maintenant, il y a une 3^e manière de « fabriquer » du HTML dans un fichier `.php` : si un fichier `.php` contient du texte hors de l'intervalle `<?php ?>`, ce texte est envoyé sans être interprété. Le programme :

```
texte 1
<?php echo "texte 2" ?>
texte 3
```

envoie au navigateur le texte :

texte 1 texte 2 texte 3.

Il peut y avoir plusieurs sections PHP ; tout se passe comme si elles s'enchaînaient :

```
texte 1
<?php echo "texte 2" ?>
texte 3
<?php echo "texte 4" ?>
texte 5
```

crée la séquence :

texte 1 texte 2 texte 3 texte 4 texte 5.

Nous ne sommes pas très partisan de ce procédé. Nous préférons avoir des instructions PHP véritables avec `echo`, mais il faut noter que ce procédé économise un niveau de guillemets et on sait qu'insérer des guillemets entre guillemets pose toujours un problème. En tous cas, nous utiliserons rarement la structure suivante :

```
<html>
<body>
Bonjour
<?php
    $x=1;
    if ($x>0) { ?>
<br>x positif<br>
<?php
    }else{ ?>
<br>x négatif<br>
<?php } ?>
Au revoir
```

```
</body>  
</html>
```

Elle est autorisée et, avec $x=1$ donc positif, elle affiche la page Web :

```
Bonjour  
x positif  
Au revoir
```

La structure PHP `if {...}else{...}` sera étudiée en son temps. Insistons encore sur le fait qu'un fichier qui mélange PHP et HTML doit avoir l'extension `.php` et non `.htm` afin de déclencher l'interpréteur PHP sur le serveur.

Chapitre 1

Atelier 1

Notre laboratoire de PHP



Mission : Préparer un « terrain de jeu » où vous pourrez tester des instructions et des programmes PHP dans la configuration exacte de votre hébergeur.

Pour tout apprentissage d'un langage, il faut pouvoir essayer les programmes qu'on écrit et les corriger jusqu'à ce que le fonctionnement soit satisfaisant. Mais il faut même pouvoir essayer, sinon des instructions individuelles, du moins de petits morceaux de programmes.



Pour PHP, les choses sont compliquées par le fait que les programmes ne sont pas censés s'exécuter sur la machine de l'utilisateur, là où ils affichent leurs résultats, mais sur un serveur lointain chez l'hébergeur du site. De fait, un programme PHP produit deux sortes de résultats :

- il effectue des actions comme créer ou modifier des fichiers ou des bases de données chez l'hébergeur ;
- il produit des sorties qui seront affichées en tant que pages Web chez l'internaute.

D'ailleurs, l'interpréteur de PHP n'est pas chez le visiteur ou chez le Webmaster, il est chez l'hébergeur et il doit fonctionner en tant que serveur.

► EasyPHP

EasyPHP est un package téléchargeable gratuitement pour vous permettre d'essayer vos développements sur votre ordinateur. Il conduit votre ordinateur à faire un « dédoublement de personnalité » : d'une part son comportement habituel de machine-visiteur, d'autre part un serveur appelé « localhost ». Pour cela EasyPHP contient :

- une version d'Apache (logiciel serveur)
- un interpréteur PHP
- le SGBD MySQL (le logiciel de BD le plus souvent associé à PHP)
- PHPMyAdmin (outil d'administration des bases MySQL).



Dans l'unique but de rendre service aux lecteurs qui voudraient l'utiliser, nous résumons ci-dessous la marche à suivre pour le téléchargement d'EasyPHP, sachant que **ce n'est pas le système que nous utiliserons dans ce livre**.

■ Téléchargement d'EasyPHP

Allez sur le site <http://www.easyphp.org> :

The screenshot shows the EasyPHP website interface. At the top, there's a navigation bar with the EasyPHP logo and a 'soon...' indicator. Below the navigation bar, there's a main heading and a paragraph describing EasyPHP's purpose. A news section dated 12/23/2006 announces the release of EasyPHP 2.0 beta 1. The page is organized into three columns of links to external resources: loogaroo.net, www.phpindex.com, and www.phpbuilder.com. At the bottom, there's a 'Support & développement' section with links to forums and discussion lists. A large 'E' logo is visible in the bottom right corner.

Figure 1-1 : Le site d'EasyPHP.

Cliquez sur le lien **téléchargements** sur le bandeau supérieur. Dans l'écran des propositions de téléchargements proposés, prenez la version la plus récente qui ne soit pas une bêta. A la date de la figure 1-2, c'est la 1.8 ; la 2.0 est en bêta, donc elle risque de présenter quelques problèmes.

Participez à la traduction du nouveau site et de la nouvelle version 2.0

Vous pouvez traduire quelques mots ou tout un fichier. C'est selon votre temps et vos connaissances. Le but est de pouvoir compter sur 7 ou 8 langues (dont l'arabe et le chinois) pour pouvoir valider les nouveaux développements.

Centre de Traductions

[\[avancement des traductions\]](#)

Les versions de EasyPHP ont été testées sous Win9x/Me/NT/2000/XP. En cas de problèmes, reportez vous à la FAQ, puis au forum ou à la liste de discussion.

Version courante

	EasyPHP 2.0 beta1	date [23/12/2006]	taille [13 496 ko]	nb de téléchargts [0626458]	détails*
	EasyPHP 1.8	date [01/03/2005]	taille [07 896 ko]	nb de téléchargts [9932213]	détails*
	langues.bt (serial 1.8.0.3)	date [23/01/2007]	taille [00 152 ko]	nb de téléchargts [0970258]	détails*

Anciennes versions

	EasyPHP 1.7	date [08/10/2003]	taille [10 900 ko]	nb de téléchargts [2242542]	détails*
	Pages locales en anglais	date [12/10/2003]	taille [00 047 ko]	nb de téléchargts [0156543]	détails*
	Pages locales en polonais	date [14/11/2003]	taille [00 079 ko]	nb de téléchargts [0019980]	détails*
	Pages locales en espagnol	date [29/01/2004]	taille [00 049 ko]	nb de téléchargts [0042041]	détails*
	Pages locales en tcheque	date [16/02/2004]	taille [00 054 ko]	nb de téléchargts [0017639]	détails*
	EasyPHP 1.6	date [24/04/2002]	taille [10 584 ko]	nb de téléchargts [0857806]	détails*
	Pages locales en anglais	date [01/05/2002]	taille [00 006 ko]	nb de téléchargts [0074788]	détails*

Figure 1-2 : Téléchargements proposés.

Vous obtenez un fichier .exe dont vous demandez l'exécution. Le mieux est d'accepter toutes les propositions qui vous sont faites et, à chaque fois, de cliquer sur **(Suivant)**. A ce moment, l'interpréteur PHP (*php.exe*) sera dans *C:\Program files\EasyPHP\php*, MySQL (*mysql.exe*) sera dans *C:\Program files\EasyPHP\MySQL\bin* ; les programmes PHP ainsi que vos pages Web devront être écrits dans *C:\Program files\EasyPHP\php\www*.

Après installation, le logo **e** sera présent sur la barre de démarrage et vous devrez cliquer dessus pour activer le système. Pour utiliser PHPMyAdmin, vous appelez *localhost/home/* dans la ligne adresse de votre navigateur.



Pourquoi nous n'utilisons pas EasyPHP dans ce livre

EasyPHP est un système remarquable. Sans lui, l'essai en local de pages Web qui font appel à PHP n'est pas possible car, à la différence de JavaScript, l'interpréteur PHP ne peut pas faire partie d'un navigateur. Mais il y a deux éléments qui nous amènent à considérer que l'essai de vos pages avec PHP sous EasyPHP sera **incomplet**. Bien sûr, un tel essai permettra de déceler certaines erreurs, principalement de syntaxe, mais il faudra toujours, après cela, envoyer les pages chez votre hébergeur et les tester en situation. Voici les deux points litigieux :

1. Selon les options d'installation et d'initialisation de PHP, certains éléments du langage sont utilisables ou non, et ce, pour des raisons de sécurité. Ces problèmes de sécurité étant moins cruciaux dans une installation locale d'EasyPHP, il y a un grand risque que le fonctionnement chez votre hébergeur soit plus restrictif que sur votre machine locale. Donc vous avez intérêt à vous habituer à tenir compte des particularités de la configuration de votre hébergeur.
2. Une fois les erreurs de syntaxe ou de frappe corrigées, la plus grande partie des échecs de fonctionnement des scripts PHP vient de ce qu'ils essaient d'effectuer des écritures sur des fichiers alors qu'ils n'ont pas les droits voulus. Ce problème est fondamental : on a précisément recours à PHP (ou à d'autres langages s'exécutant sur le serveur) parce-que c'est le seul moyen d'agir sur des fichiers sur le serveur, notamment pour la mise à jour en ligne des pages ; mais les serveurs fonctionnent le plus souvent sous Unix ou Linux, systèmes très stricts sur les droits d'accès aux fichiers, ce en quoi ils ont une meilleure sécurité que Windows. Bien sûr avec EasyPHP sous Windows sur **votre** machine, vous avez tous les droits, donc un problème dans ce domaine ne sera jamais décelé et c'est une fois que le site sera dans sa situation finale que vous aurez une mauvaise surprise.

Note

Il existe un autre système appelé XAMPP, qui comme EasyPHP installe un serveur sur votre machine ; nous ne l'utiliserons donc pas, pour les mêmes raisons. Il est téléchargeable (gratuitement) sur

<http://www.apachefriends.org/fr/index.html>.

► Fixation des droits sur les fichiers

Lorsque vous exécutez un script PHP sur votre site chez votre hébergeur, vous êtes le propriétaire des fichiers que vous créez, donc vous avez les droits d'écriture. Mais vous devez donner ces droits à toutes les catégories d'utilisateurs, pour que les visiteurs de votre site aient ces droits.