

Introduction au langage PHP



Article rédigé par Emacs

Le langage PHP a été inventé par Rasmus LERDORF en 1995 pour son usage personnel (mise en ligne de son CV en l'occurrence). Autrefois abbréviation de Personal HomePage devenue aujourd'hui Hypertext Preprocessor, PHP s'impose comme un standard dans le monde de la programmation web par ses performances, sa fiabilité, sa souplesse et sa rapidité.

Préambule

PHP a été inventé à l'origine pour le développement d'applications web dynamiques qui constituent encore le cas d'utilisation le plus courant et son point fort. Cependant, les évolutions qui lui ont été apportées jusqu'à aujourd'hui assurent à PHP une polyvalence non négligeable. PHP est par exemple capable d'interagir avec Java, de générer des fichiers PDF, d'exécuter des commandes Shell, de gérer des objets (au sens programmation orientée objet), de créer des images ou bien de fournir des interfaces graphiques au moyen de PHP GTK.

Dans cette présentation du langage, nous introduirons tout d'abord les caractéristiques de PHP, puis nous verrons en quoi il est particulièrement adapté aux développements d'applications web. Nous synthétiserons ensuite les autres types d'applications possibles avec PHP avant de terminer sur les limites que l'on peut lui reprocher.

Les caractéristiques du langage PHP

License

PHP est tout d'abord un langage de script interprété (en réalité précompilé en Opcode), gratuit, OpenSource et distribué sous une license autorisant la modification et la redistribution.

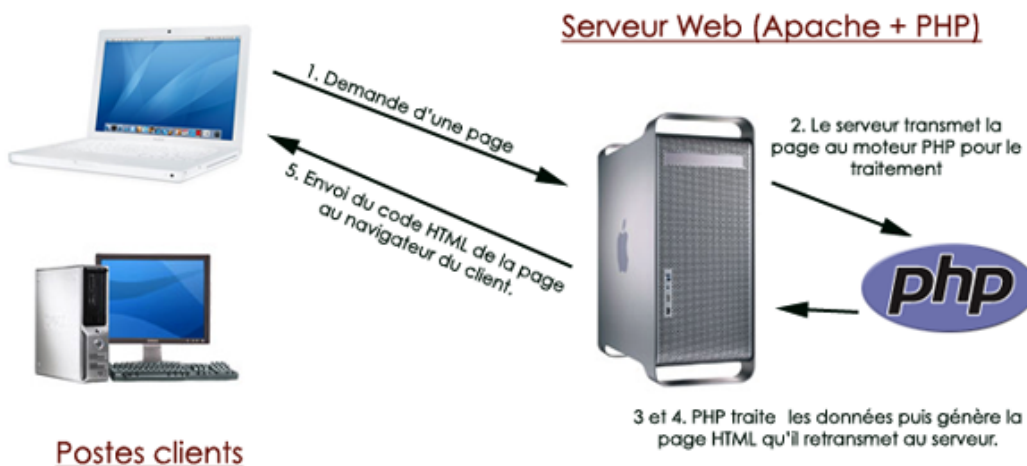
Portabilité

PHP est supporté sur plusieurs systèmes d'exploitation. C'est par exemple le cas des versions Microsoft Windows™, mais aussi des systèmes reposant sur une base UNIX (Apple MAC OS X™, distributions Linux ou encore Sun Solaris).

Il sera alors très facile de déplacer une application écrite en PHP d'un serveur Windows d'origine vers un serveur Linux sans avoir à la modifier (ou très peu).

Exécution

D'un point de vue exécution, PHP a besoin d'un serveur Web pour fonctionner. Toutes les pages demandées par un client seront construites par le serveur Web, en fonction des paramètres transmis, avant d'être retournées au client. Le schéma ci-dessous illustre le principe de fonctionnement de PHP.



Note : il aurait été possible d'ajouter un serveur de bases de données (local ou distant) sur cette illustration. Nous aurions eu alors deux étapes supplémentaires qui sont l'interrogation de la base de données par PHP et la récupération des résultats en provenance du serveur SQL.

Apprentissage de PHP

PHP est un langage dit de « haut niveau » dont la syntaxe est très proche du langage C. Cette syntaxe proche du langage naturel lui assure un apprentissage rapide et peu de contraintes d'écriture de code. Néanmoins, la maîtrise rapide de sa syntaxe ne signifie pas la maîtrise de ses fonctionnalités et de ses concepts. Une bonne connaissance et une utilisation avancée de la programmation PHP nécessite un temps d'apprentissage relativement long.

Richesse du langage PHP

Une des forces du langage PHP est sa richesse en terme de fonctionnalités. En effet, il dispose à l'origine de plus de 3 000 fonctions natives prêtes à l'emploi garantissant aux développeurs de s'affranchir de temps de développement supplémentaires et parfois fastidieux. Ces fonctions permettent entre autre de traiter les chaînes de caractères, d'opérer mathématiquement sur des nombres, de convertir des dates, de se connecter à un système de base de données, de manipuler des fichiers présents sur le serveur...

PHP puise aussi sa richesse dans le dynamisme de sa communauté de développeurs. Celle-ci était estimée à 500 000 personnes environ en 2003 mais il est très probable qu'elle ait dépassé le million maintenant. Les profils de développeurs de la communauté sont très divers. Il y'a ceux qui apportent de nouvelles fonctionnalités et bibliothèques de version en version, ceux qui traduisent la documentation en plusieurs langues ou encore les programmeurs ayant des compétences plus modestes qui réalisent des applications Opensources prêtes à l'emploi. Parmi les plus connues, nous pouvons citer les CMS (Joomla, SPIP, Dotclear, Wordpress...), les systèmes de ventes en ligne (OSCommerce), les forums (PHPBB, IPB, VBulletin), les frameworks (Zend Framework, Symfony, CakePHP, Jelix)...

PHP, un langage fiable et performant

Le langage est maintenant devenu un langage fiable, performant et viable. Il est capable de supporter des sites qui sollicitent des millions de requêtes journalières. De nombreuses entreprises de renommée nationale et internationale lui font confiance pour le développement de leur site Internet. Nous pouvons parmi elles citer TF1, IBM, Le Monde, Le Figaro, Club-Internet, Orange, Pages Jaunes... Un récent rapport daté de novembre 2006 indique que 87% des entreprises du CAC40 utilisent PHP. [Lire l'interview de Perrick Penet \(AFUP\)](#).

PHP, un langage conçu pour les applications web dynamiques

Le langage PHP a la principale fonction d'être spécialement conçu pour la réalisation d'applications web dynamiques. Par définition, une « application (ou page) dynamique » est un programme capable de générer une page unique en fonction de paramètres qui lui sont transmis.

Un script PHP peut donc être intégré directement à l'intérieur d'un code html. Petit exemple pratique ci-dessous :

```
Premier programme PHP : "Hello World"
<html>
  <head>
    <title>Hello World en PHP</title>
  </head>
  <body>
    <p>
      <?php echo 'Hello World !'; ?>
    </p>
  </body>
```

Le script PHP, clairement identifié par les deux balises <?php (ouverture) et ?> (fermeture), provoquera l'écriture de la chaîne de caractères Hello World entre les balises html après son exécution sur le serveur.

Nous aurions également pu obtenir le même résultat en utilisant le script php suivant :

```
Autre version du "Hello World"
<html>
  <head>
    <title>Hello World en PHP</title>
  </head>
  <body>
    <?php echo '<p>Hello World !</p>'; ?>
  </body>
```

De ce fait, on en déduit que PHP est capable de générer du code HTML (ainsi que d'autres formats), ce qui fait tout son intérêt. La présentation du document généré est alors complètement dépendante des conditions passées et des paramètres initiaux fournis. Prenons l'exemple du site de vente par correspondance [Amazon.fr](#). Les utilisateurs inscrits au site qui ont déjà passé plusieurs commandes auront la surprise de voir à chaque nouvelle visite sur leur page personnalisée, une liste de produits sélectionnés automatiquement qui correspondent aux critères de ses précédents achats. Chacune de ces actions de marketing direct ciblé est unique et générée en fonction des intérêts du consommateur.

PHP pour d'autres domaines d'application

Fort de sa richesse, PHP ne se limite pas forcément à l'édition de pages web dynamiques. Il peut par exemple être utilisé en ligne de commande via l'utilisation de l'exécutable *php*. Ce cas d'utilisation permet alors d'exécuter des scripts directement sur les machines. Un script PHP serait alors mis au profit de la machine. Nous pouvons très bien imaginer un programme PHP capable de supprimer un certain nombre de

fichiers présents dans un dossier. Il est même possible de coupler l'utilisation de PHP avec un gestionnaire de tâches tel qu'un serveur *cron* sous Linux.

PHP c'est également la possibilité de créer des applications lourdes fonctionnant sans serveur ni navigateur. Autrement dit des applications traditionnelles, autonomes et munies de fenêtres. Tout cela se réalise au moyen de la librairie PHP GTK disponible à l'adresse : <http://gtk.php.net>

Un autre point fort de PHP est sa capacité à s'interfacer très facilement avec de nombreux systèmes de gestion de bases de données relationnelles (SGBDR). Parmi eux, nous pouvons retrouver *MySQL*, *Oracle*, *SQLite*, *MSSQL*, *PostgreSQL*... Grâce à ces systèmes couplés au langage PHP, il devient possible de distribuer les applications sur plusieurs serveurs (serveur Web + serveur de bases de données). Le second intérêt à cela est de pouvoir rendre une application encore plus dynamique. En effet, les données (contenu) de l'application se trouve à présent dans la base de données et PHP se charge de les récupérer puis de les manipuler (traitement des chaînes de caractères, enregistrement dans des fichiers, génération de flux RSS...).

Les possibilités offertes par PHP sont donc très nombreuses et nous n'allons pas les détailler toutes car nous y resterions des heures. Retenons néanmoins une liste de ses principales capacités :

- Manipulation d'un système de fichiers (création, édition, suppression, droits d'accès...)
- Gestion des sessions utilisateurs
- Génération et parsing de documents XML grâce à la librairie SimpleXML
- Génération d'images avec GD2
- Génération de fichiers PDF
- Accès simplifié aux bases de données avec la librairie PDO
- Exécution de commandes Shell
- Gestion des e-mails en POP et IMAP
- Compression et décompression d'archives ZIP
- Cryptage MD5 et SHA1
- Gestion d'annuaires LDAP
- Manipulation des dates
- Manipulation des URL
- Envoi et lecture de cookies
- Dialogue avec Java
- Utilisation d'Ajax
- ...

La version actuelle de PHP apporte un grand vent de fraîcheur au langage et aux professionnels. Le principal manque jusque là était son modèle objet trop succinct. Les développeurs se sont alors penchés sur cette problématique et ont finalement implémenté un modèle de programmation objet proche du langage Java. Grâce à un tel modèle, les professionnels encore hésitants à utiliser PHP deviennent de plus en plus nombreux. Des applications complètement objet voient également le jour et intègrent un design pattern (motif de conception) MVC. Les plus connues aujourd'hui sont les frameworks [Zend](#) et [Symfony](#) des sociétés respectives [Zend Technologies](#) et [Sensio](#) (agence française).

Les limites de PHP

Malgré toutes les qualités que nous pouvons attribuer au langage PHP, subsistent quelques défauts. Par exemple, dans le cas de très grosses applications, il peut présenter quelques faiblesses et devenir inadapté. Un langage tel que PERL deviendrait alors plus adéquat. Cet argument reste toutefois très subjectif dans la mesure où les développeurs de PHP améliorent la qualité et la robustesse du langage.

Le second défaut (mais qui paradoxalement fait son succès et sa qualité) que nous pouvons lui reprocher est sa grande simplicité d'utilisation. Cela a beaucoup terni l'image de PHP parceque tout webmaster (même très peu expérimenté) devient capable de créer du code et des applications facilement. Cependant, la plupart des codes produits par des développeurs amateurs n'est pas forcément "propre" ou bien conçu, souvent peu sécurisé, peu maintenable et même non optimisé. Les autres langages comme C++, .Net, Java, ASP, Perl, Python ou Ruby ne subissent pas cette mauvaise image dans la mesure où leur apprentissage n'est pas forcément très aisé.

Enfin, le dernier défaut reprochable à PHP est son manque de rigueur dans la nomenclature des fonctions et de la syntaxe (voir tutoriel sur [l'utilisation des balises courtes](#)). Tout d'abord, les fonctions ne sont pas sensibles à la casse, ce qui signifie par exemple que `str_replace()` et `STR_REplAcE()` sont identiques pour l'interpréteur PHP. Heureusement la nouvelle version en cours (PHP6) résoudra ce défaut. Par ailleurs, nous constatons un manque de standardisation des noms des fonctions :

- Utilisation du séparateur underscore : `str_replace()`, `preg_match()`, `mysql_real_escape_string()` ...
- Fonctions composées de plusieurs mots écrite en un seul : `wordwrap()`, `htmlspecialchars()` ...
- Traduction du *to* en littéral ou numéraire : `bin2hex()`, `strtotime()`...

Conclusion

Nous concluons que PHP a encore de beaux jours devant lui et que son avenir sera encore très prometteur avec l'arrivée de sa nouvelle version. Celle-ci lui apportera d'ailleurs une touche plus professionnelle car elle se destine plus particulièrement aux utilisateurs confirmés et professionnels.