

PHP



Introduction	3
Présentation du concept.....	3
Historique	4
Fonctionnement	4
Programmation orientée objet	5
L'utilisation	6
Versions	6
La communauté	6
Framework.....	8
Bibliothèque.....	10
Avantage et Inconvénients.....	10
Avantages :.....	10
Inconvénients :.....	10
Conclusion	11
Bibliographie :.....	11

Introduction

Le PHP est un langage de script utilisé le plus souvent côté serveur. Cette architecture permet au serveur d'interpréter le code PHP des pages web demandées et génère du code (HTML, XHTML, CSS) et des données (JPEG, GIF, PNG) pouvant être interprétés et rendus par un Navigateur web.

Il peut également générer d'autres formats comme le WML, le SVG, le PDF. Il a été conçu pour permettre la création d'applications dynamiques, le plus souvent développées pour le Web. PHP est le plus souvent couplé à un serveur Apache bien qu'il puisse être installé sur la plupart des serveurs HTTP.

Ce couplage permet de récupérer des informations issues d'une base de données, d'un système de fichiers ou plus simplement des données envoyées par le navigateur afin d'être interprétées ou stockées. C'est un langage peu typé et souple et donc facile à apprendre par un débutant.

Présentation du concept

L'utilisation commence avec le traitement des formulaires puis par l'accès à la base de données. La force la plus évidente de ce langage est qu'il a permis au fil du temps la résolution aisée de problèmes autrefois compliqués et est devenu par conséquent un composant incontournable des offres d'hébergements. Il est multiplateforme autant sur Linux qu'avec Windows il permet aisément de reconduire le même code sur un environnement à peu près semblable.

Libre, gratuit, simple d'utilisation et d'installation, ce langage nécessite comme tout langage de programmation une bonne compréhension des principales fonctions usuelles ainsi qu'une connaissance aigüe des problèmes de sécurité liés à ce langage. La version 5.3 a introduit de nombreuses fonctions nouvelles comme les espaces de noms, élément important pour créer des bibliothèques et des Framework.

En 2018, près de 80 % des sites web utilisent le langage PHP sous ses différentes versions. Le langage PHP fait l'objet, depuis plusieurs années maintenant, de rassemblements nationaux organisés par l'AFUP (l'Association Française des Utilisateurs de PHP), où experts de la programmation et du milieu se retrouvent pour échanger autour du PHP et de ses développeurs.

Historique

Le langage PHP a été créé en 1994 par Rasmus LERDORF pour son site web. C'était à l'origine une bibliothèque pour un logiciel, il servait pour conserver une trace des visiteurs qui venaient consulter son CV.

Au fur et à mesure Rasmus LERDORF ajoutait de nouvelles fonctionnalités, la bibliothèque a évolué en une implémentation capable de communiquer avec des bases de données et de créer des applications pour le Web. En 1995, le PHP devient libre pour que tout le monde puisse l'utiliser et en profiter. Le PHP s'appelait alors PHP/FI (pour Personal Home Page Tools/Form Interpreter).

En 1997, deux étudiants, Andi GUTMANS et Zeev SURASKI, ont redéveloppé le cœur de PHP/FI. Ce travail a abouti un an plus tard à la version 3 de PHP. Peu de temps après, Andi GUTMANS et Zeev SURASKI ont commencé la réécriture du moteur interne de PHP. C'est ce nouveau moteur, appelé Zend Engine, le mot Zend est la contraction de Zeev et Andi qui a servi de base à la version 4 de PHP. Actuellement le PHP est à la version 7.3.1 .

Fonctionnement

PHP appartient à la grande famille des descendants du C, la syntaxe est très proche. En particulier, la construction ressemble à celles des langages Java et Perl, à ceci près que du code PHP peut facilement être mélangé avec du code HTML au sein d'un fichier PHP.

Dans une utilisation destinée à internet, l'exécution du code PHP se déroule ainsi : lorsqu'un visiteur demande à consulter une page de site web, son navigateur envoie une requête au serveur HTTP correspondant. Si la page est identifiée comme un script PHP (grâce à l'extension .php), le serveur interprète le PHP qui va traiter et générer le code final de la page (constitué généralement d'HTML ou de XHTML, mais aussi souvent de CSS et de JS). Ce contenu est renvoyé au serveur HTTP, qui l'envoie finalement au client.

Ce schéma explique ce fonctionnement :



Programmation orientée objet

Le PHP permet de programmer en orienter objet, il nous permet de crée des classes contenant des attributs ainsi que des méthodes, elles peuvent être instanciées ou utilisées en statique.

Toutefois, le PHP est un langage à héritage simple, c'est-à-dire qu'une classe ne peut hériter que d'une seule autre classe. Cependant les interfaces peuvent en étendre plusieurs autres.

Exemple de création d'une classe :

```
<?php

class Perso {
    const PV_INITIAL = 2000;
    private $pv;

    public function __construct($pv = false) { // Paramètre optionnel
        if (!is_numeric($pv) || $pv < 0 || $pv > 100000000)
            $this->pv = self::PV_INITIAL;
        else
            $this->pv = $pv;
    }

    // Accesseurs
    public function getPv() {
        return $this->pv;
    }

    public function isDead() {
        return $this->pv == 0;
    }
}

// Création d'une classe enfant de Perso
class Magicien extends Perso {
    private $magie;
}

// Création d'une instance de classe
$perso = new Perso(1000);
// Utilisation de l'objet
echo 'Votre personnage a ' . $perso->getPV() . ' PV.';

// Constantes de classes
echo 'Le PV par défaut attribué à un nouveau personnage est de ' . Perso::PV_INITIAL . '.';

// Destruction de l'objet
unset($perso);
```

L'utilisation

En 2002, le PHP est utilisé par plus de 8 millions de sites Web à travers le monde, en 2007 par plus de 20 millions et en 2013 par plus de 244 millions. De plus, PHP est devenu le langage de programmation web côté serveur le plus utilisé depuis plusieurs années.

Année	Part de marché
2010	75 %
2013	75 % ¹³
2016	82 % ¹⁴

Enfin en 2010, le PHP est le langage dont les logiciels open source sont les plus utilisés dans les entreprises, avec 57 % de taux de pénétration.

Versions

La version actuelle est la version 7.2.9, sortie le 16 août 2018. Elle utilise *Zend Engine 2* et introduit une modélisation des objets plus performante, cette version apporte une gestion des erreurs fondées sur le modèle des exceptions (les exceptions permettent de simplifier, personnaliser et d'organiser la gestion des « erreurs » dans un programme), ainsi que des fonctionnalités de gestion pour les entreprises. La version 5 apporte beaucoup de nouveautés, telles que le support de SQLite (SQLite permet d'avoir accès à une base de données SQL sans avoir de processus SGBD séparé) ainsi que des moyens de manipuler des fichiers et des structures XML basés sur libxml2.

- Une API simple nommée SimpleXML
- Une API Document Object Model assez complète
- Une interface utilisant les objets DOM et SimpleXML
- Une meilleure gestion des objets par rapport à PHP 4, avec des possibilités qui tendent à se rapprocher de celles de Java.

Depuis juin 2011 et le nouveau processus de livraison de PHP, le cycle de livraison de PHP est d'une mise à jour annuelle comportant des changements fonctionnels importants. La durée de vie d'une branche est de 3 ans et trois branches stables sont maintenues. Cela signifie que lorsqu'une nouvelle version de PHP 5.x sort, la version 5.x-3 n'est plus supportée.

Historiquement, le PHP disposait d'une configuration par défaut privilégiant la souplesse à la sécurité. Cette souplesse a permis à de nombreux développeurs d'apprendre le PHP mais le revers de la médaille a été que de nombreuses applications PHP étaient mal sécurisées. Le sujet a bien été pris en main par le PHPGroup qui a mis en place des configurations par défaut mettant l'accent sur la sécurité. Il en résultait une réputation de langage peu sécurisé, réputation d'insécurité qui n'a plus de raison d'être.

La communauté

Le nombre de développeurs PHP en France est la plus vaste de tous les langages de programmation. Elle est la plus représentée avec environ 35% de projets technologiques français réalisés en PHP, tandis que les populations favorisant le JavaScript et le Java la talonnent avec environ 18 et 17% d'utilisateurs. Le PHP est autant prisé par la communauté pour plusieurs raisons.

Une grande communauté, c'est le principal avantage de l'utilisateur (dizaine de milliers de bibliothèques, nombreuse documentation et tutoriels, etc.).

Économique, le PHP est open source donc gratuit et ne nécessite aucune licence d'utilisation, le PHP colle avec l'esprit de solidarité et d'entraide des développeurs et attire ainsi beaucoup d'utilisateurs débutants comme confirmés.

Une prise en main facile, d'une grande simplicité d'utilisation ce langage attire beaucoup les débutants.

Performance et rapidité, c'est grâce à certains Frameworks comme Laravel et Symfony qui font du PHP l'un des langages les plus rapides et efficaces.

Le langage PHP comme beaucoup d'autres langages peuvent être utilisés de trois façons différentes :

- Purement (sans Framework).
- Sous un Framework propriétaire.
- Sous un Framework Open-Source.

Framework

Les Framework PHP sont très nombreux et certains d'entre eux sont très connus comme Laravel, CodeIgniter, Symfony, Zend Framework, Yii, CakePHP, sont les cinq Framework les plus répandus et utilisés.

Laravel :

La première version du Framework a été créée en juin 2011, il est open-source et respecte le principe modèle-vue-contrôleur. Il est entièrement développé en programmation orientée objet. La version 5.0 de Laravel nécessite au minimum PHP 5.46 son installation est basée sur le gestionnaire de paquets Composer2.

Ses fonctionnalités principales résident dans le routage de requête, le mapping objet-relationnel, d'authentification, migration de base de données.

CodeIgniter :

La première version de CodeIgniter a été publiée le 28 février 2006. La version majeure (2.0.0) est sortie le 28 janvier 2011, elle est symbolisée par l'abandon de l'utilisation du PHP 4 au profit de PHP 5.

Il dispose d'une communauté très active, ses membres ont développés de nombreuses bibliothèques qui permettent un usage complémentaire au Framework. Le Framework encourage fortement l'utilisation de l'architecture Modèle-Vue-Contrôleur.

Symfony :

Symfony a été créée le 18 octobre 2005, en septembre 2017, Symfony passe la barre du milliard de téléchargements. Il se base sur la version 5 de PHP et impose l'utilisation du Modèle-Vue-Contrôleur.

Disposant de performances optimisées et d'un système de cache pour diminuer le temps de réponse cela fait pour lui un atout majeur. Il inclut aussi la gestion des URL qui permet à une page d'avoir une URL distincte dans l'arborescence, il ajoute aussi un système de configuration utilisant le langage YAML et le support d'AJAX.

Zend :

Le Zend Framework est un cadre (Framework) créé en mars 2006. Il est distribué sous la Licence BSD Modifiée (licence libre). Zend Framework, aussi appelée ZF, fut développé dans le but de simplifier le développement Web tout en donnant des bonnes pratiques et la conception orientée objets en offrant des outils aux développeurs. ZF permet aussi d'utiliser nativement le principe de MVC (Modèle-Vue-Contrôleur), mais n'y oblige pas.

Yii :

Le Yii Framework ("Yes, It Is") est un cadre (Framework) pour PHP 5. Il utilise la programmation orientée objet. Destiné au développement Web, il nécessite la version 5.1.0 de PHP. Sa communauté est très active. Yii est le successeur officiel de Prado.

Ces fonctionnalités sont une architecture Modèle-Vue-Contrôleur, un accès aux bases de données par DAO/ActiveRecord, la gestion de caches, le support de AJAX via jQuery, la génération automatique du code PHP pour les opérations de base (CRUD), le contrôle des saisies utilisateurs sur les formulaires, la migration de bases de données.

CakePHP :

CakePHP est un Framework web libre écrit en PHP distribué sous licence MIT. Il suit le motif de conception Modèle-Vue-Contrôleur, il imite le fonctionnement de Ruby on Rails. Créé en 2005. Il dispose d'un grand nombre de tutoriels.

Compatible avec PHP5 et PHP7, il permet l'intégration de CRUD simplifié pour l'utilisation des bases de données SQL. Il apporte une rapidité et flexibilité avec un moteur de Template. Une validation des données. Des composants de sécurité, ainsi que de gestion des droits et de gestion des sessions. Cache des vues et des actions flexibles. Scripts en ligne de commande permettant la génération automatique de code à partir du modèle physique de données.

Bibliothèque

Il est plus difficile de créer une extension qu'une bibliothèque, c'est pourquoi les principales extensions sont généralement hébergées sur le site de php.net.

Les extensions PHP sont divisées en trois catégories :

- Les extensions qui font partie de la distribution dite standard.
- Les extensions PECL.
- Les extensions externes.

Les extensions standard sont utilisées très fréquemment par la majorité des développeurs PHP, elles sont donc incluses dans toutes les distributions de PHP.

Les extensions PECL sont habituellement moins sollicitées car elles répondent à des besoins bien plus spécifiques. Elles sont toutefois également disponibles sur le site de PHP.

Les extensions externes sont en fait toutes celles que le site de PHP ne répertorie pas. Il faut prendre garde avec ces extensions, car elles ne sont pas validées par le PHP Group. Leur installation est souvent largement plus complexe pour les utilisateurs de Windows, car la procédure de compilation n'est pas toujours bien détaillée et car les fichiers binaires ne sont pas compilés par le PHP Group. Certains groupes publient de très bonnes extensions qui ne sont pourtant pas proposées sur le site de PHP, il faut donc juger avec discernement...

Avantage et Inconvénients

Avantages :

- Syntaxe classique et pratique proche du C.
- Disponible sur une grande majorité des serveurs.
- Facile à prendre en main.
- S'inscrit directement dans les pages.
- Complet avec plus de 500 fonctions : accès fichier, mail, expressions, etc.
- Ecriture souple.
- Elle offre aussi un haut degré de portabilité, ce qui permet aux utilisateurs de s'enregistrer et de partager des containers sur une large variété d'hôtes au sein d'environnements publics et privés.

Inconvénients :

- Problèmes dus à la gestion automatique des types de données.
- Absence de débogueur.
- Assez lent.

Conclusion

En conclusion, le PHP est très abordable pour commencer à développer pour aller plus loin dans ses fonctionnalités on peut utiliser différentes librairie et Framework pour nous permettre d'optimiser et d'aller plus loin et nous permet d'optimiser certaines fonctionnalités. Au vu du nombre de développeur pratiquant le PHP, une grande communauté c'est formé autour. Malgré l'instabilité et les failles de sécurités au début des versions de PHP le langage est devenue par la suite stable et sécurisé.

Bibliographie :

<https://fr.wikipedia.org/wiki/PHP>

<http://www.phpdebutant.org/article118.php>

<https://medium.com/bluecoders/pourquoi-les-bons-developpeurs-php-sont-ils-si-durs-a-trouver-a4e48ee4b0f2>

<https://fr.wikipedia.org/wiki/Laravel>

<https://fr.wikipedia.org/wiki/CodeIgniter>

<https://fr.wikipedia.org/wiki/Symfony>

https://fr.wikipedia.org/wiki/Zend_Framework

https://fr.wikipedia.org/wiki/Yii_Framework

<https://fr.wikipedia.org/wiki/CakePHP>

https://php.developpez.com/faq/bibliotheques-extensions/?page=divers#biblio_definition

<http://www-igm.univ-mlv.fr/~dr/XPOSE/php/>

<https://www.ovh.com/fr/blog/non-langage-php-pas-mort/>