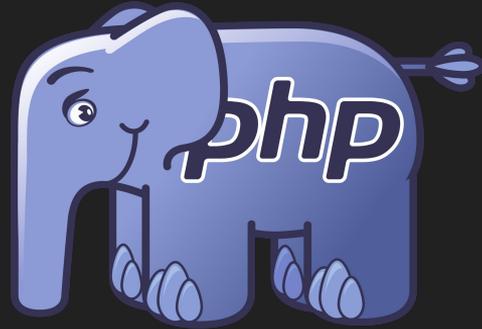


PHP : Hypertext Preprocessor



PHP : Philosophie

- Historique.
- Cycle de vie de PHP.
- Mise à jour de PHP.
- PHP est open source et a une licence de type BSD.
- PHP propulse de nombreux CMS, possède de nombreux frameworks * et outils et a une communauté très active.

* Un framework est un “cadre de travail”. L’objectif d’un framework est généralement de simplifier le travail des développeurs en leur offrant une architecture “prête à l’emploi”.



Historique

- 1994: Création du langage par Rasmus Lerdorf au Canada, pour suivre le trafic des visiteurs sur son CV.
- 1995: Rasmus partage le code.
- 1997: Zeev Suraski et Andi Gutmans sortent la version 3.
- 1999: PHP 4 intègre un nouveau moteur: le Zend Engine.
- 2010: Côté serveur, il englobe 75% du marché.
- 2021: Version 8.1, maintenance par la PHP Development Team (<https://www.php.net/credits.php>).

Cycle de vie de PHP

- Toutes les versions de PHP ont un cycle de vie d'environ trois ans, durée après laquelle elles ne sont plus prises en charge.
- Une version de PHP est activement prise en charge pendant près de deux ans après le lancement de la première version.
- Lorsqu'une version de PHP arrive en fin de cycle de vie, elle ne fait plus l'objet d'aucune mise à jour. Passé ce délai, son utilisation n'est plus sécurisée.
- PHP supported Versions: <https://www.php.net/supported-versions.php>

Mise à jour de PHP



- Meilleure sécurité: les failles de sécurité sont comblées, le code obsolète n'est plus supporté.
- Plus rapide: temps d'exécution plus court, besoins moins élevés en mémoire.
- Améliorations et nouvelles fonctionnalités: ajout de nouvelles fonctions (<https://www.php.net/releases/>).

Mise à jour de PHP



- C'est un processus qui doit être planifié.
- Il faut faire de nombreux tests en local ou sur un serveur de développement afin de vérifier que tout fonctionne toujours.
- Il faut parfois retravailler le code.
- Mettre à jour est un processus "obligatoire" pour des raisons de sécurité.

PHP est open source

- PHP est un langage open source (PHP Licensing: <https://www.php.net/license/index.php>).
- L'open source est à l'origine un modèle de développement logiciel, prônant l'ouverture et la collaboration via la libre distribution du code source.
- Le terme "open source" a une considération technique. Le terme "libre" une considération philosophique ([Open source et logiciel libre, une définition en un clic](#)).

PHP a une licence de type BSD

- La licence BSD ([Berkeley software distribution license](#)) est une licence libre utilisée pour la distribution de logiciels.
- Elle permet de réutiliser tout ou partie du logiciel sans restriction, qu'il soit intégré dans un logiciel libre ou propriétaire.
- On peut donc créer des sites et des applications sous PHP et les vendre ou les distribuer sans devoir publier le code source: Content management systems, modules, templates...
- A titre d'exemple, Java nécessite des licences payantes pour créer du contenu qui sera vendu.

PHP propulse de nombreux CMS (Systèmes de gestion de contenu)



WORDPRESS



Drupal



PrestaShop



Joomla!®

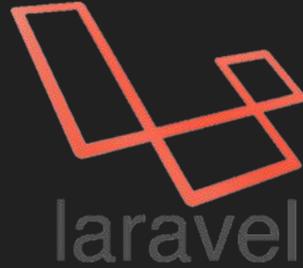


opencart®



Magento™

PHP possède de nombreux frameworks et outils, ainsi qu'une communauté très active



PHP : Technique

- PHP est un langage dynamique
- Architecture statique + schéma
- Développement Front-End
- Architecture dynamique + schéma
- Développement Back-end
- Développement Full-stack
- PHP en local

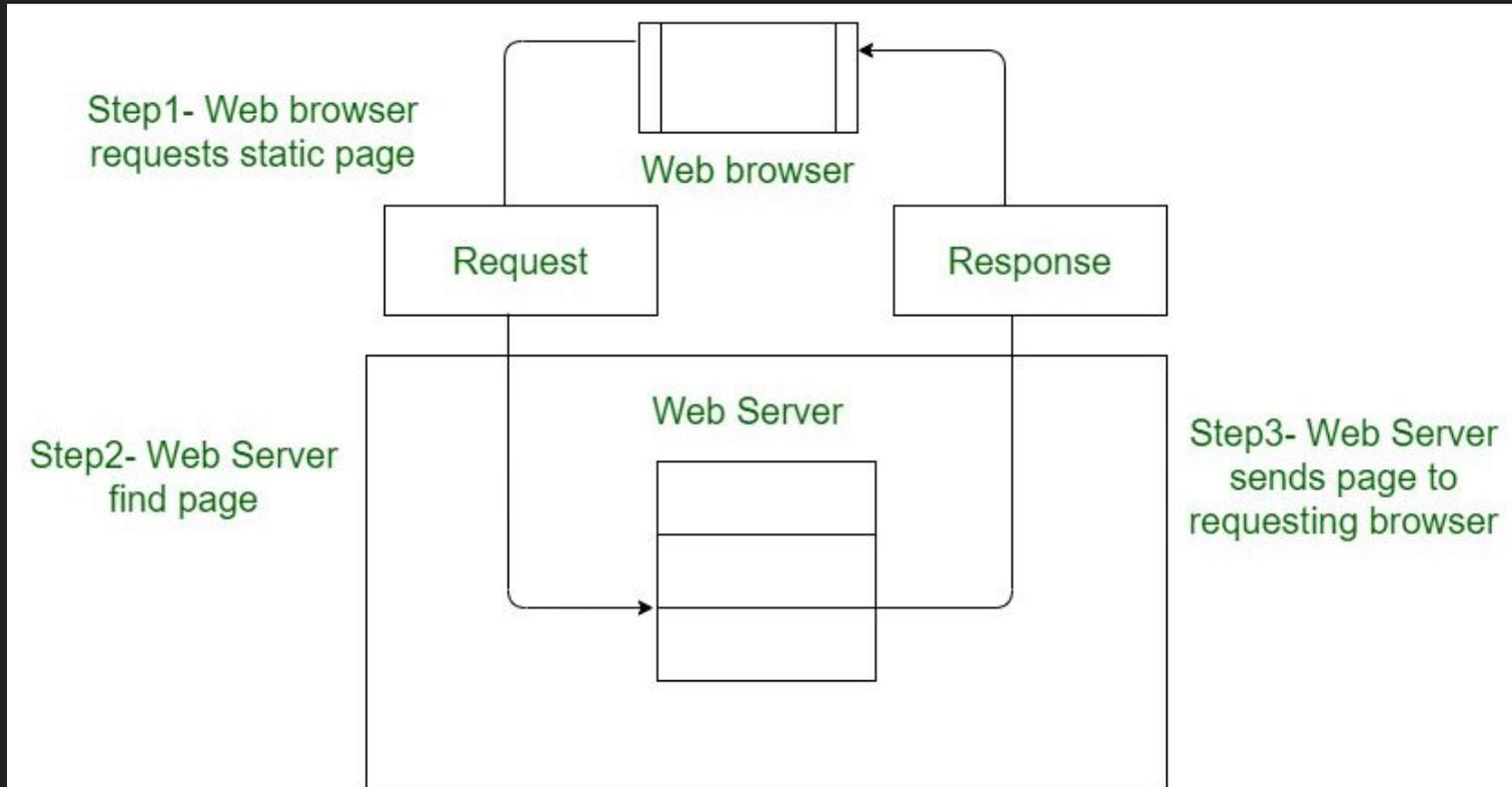
PHP est un langage dynamique

- PHP est un langage de script interprété / exécuté côté serveur puis envoyé à la machine client.
- Le PHP peut générer du code HTML de façon dynamique, c'est à dire que le contenu renvoyé par le serveur sera différent en fonction des actions des différents utilisateur.
- Ce contenu peut provenir d'une base de données car PHP est capable d'interroger et de lire des bases de données, de les modifier (mise à jour et effacement de données) ou d'ajouter des éléments dans celle-ci (CRUD: Create, Read, update, Delete).

Architecture statique

- L'utilisateur veut afficher une page en cliquant sur un lien.
- Une requête est envoyée au serveur pour afficher cette page.
- Le serveur renvoie la page telle quelle au client (le serveur ne parcourt pas le contenu).
- Les pages renvoyées sont toujours les mêmes quel que soit l'utilisateur, d'où le nom "statique".
- La page et son contenu sont interprétés par la machine client (navigateur) qui va analyser le code, l'interpréter et l'afficher correctement.
- Le navigateur ne sait interpréter que du HTML (contenu), du CSS (mise en forme) et du Javascript (gestion d'événements).

Architecture statique schéma



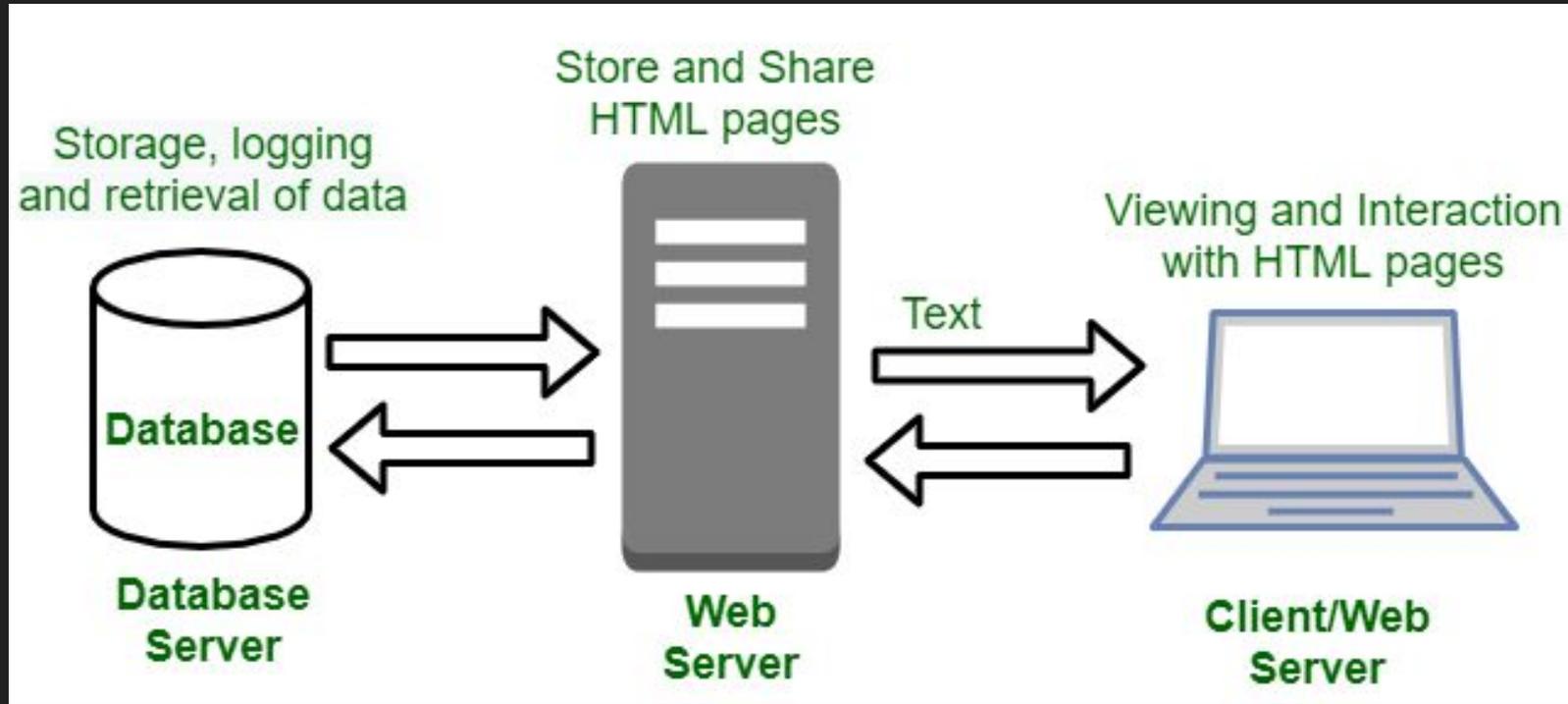
Développement Front-End

- Ce qu'on appelle la partie "client" est tout ce qui est exécuté sur le navigateur de l'utilisateur.
- Le HTML n'est pas un langage de programmation, mais un langage de représentation. Le CSS est un langage de feuille de styles.
- Javascript est un langage de programmation et peut être utilisé sous formes de bibliothèques ou framework (React, VueJS, Angular).
- La personne en charge de ces technologies est le développeur Front-End. Si son travail ne consiste qu'à reproduire une maquette en utilisant HTML / CSS, c'est un intégrateur.

Architecture dynamique

- L'utilisateur veut afficher une page en cliquant sur un lien ou en remplissant un formulaire.
- Une requête est envoyée au serveur pour afficher cette page et les données correspondant.
- Le serveur interprète le code PHP et si c'est requis par le code PHP, des requêtes sont faites vers la base de données.
- Le serveur renvoie une page HTML au client avec les données adaptées en fonction des actions qui avaient été effectuées par l'utilisateur.

Architecture dynamique schéma



Développement Back-end

- Ce qu'on appelle la partie "serveur" (ou server-side) est tout ce qui est interprété côté serveur.
- PHP est capable de traiter et de renvoyer des données en fonction des actions de l'utilisateur.
- Ces données ne seront pas les mêmes en fonction des actions ni même des différents utilisateurs.
- PHP peut ne pas renvoyer de page HTML mais juste des données non formatées. Dans ce cas l'on parle de PHP utilisé pour créer une API (interface de programmation d'application).
- La personne en charge de ces technologies est le développeur Back-end.

Développement Full-stack

- Un développeur full-stack est un développeur s'occupe à la fois de la partie front-end et de la partie back-end.
- Hormis pour la réalisations d'API (ne renvoyant que des données brutes), un développeur PHP qui réalisera aussi les pages HTML et leur mise en forme devra avoir des compétences de front-end et donc être full-stack.
- Exception: dans le cadre d'une architecture Modèle-Vue-Contrôleur (MVC) ou d'un CMS, il sera possible de séparer les différentes parties front-end et back-end au sein d'une même application, et donc de séparer les tâches des développeurs.

PHP en local

Pour exécuter des applications PHP en local il est nécessaire d'installer 3 éléments:

- PHP (bien choisir sa version, notamment en fonction du cycle de vie, mais aussi en fonction de ce qui est proposé par l'hébergeur).
- Un serveur, qui va interpréter PHP mais aussi contacter la base de données: Apache, Nginx... A noter que depuis la version 5 PHP propose son serveur interne (à n'utiliser que pour des tests).
- Un système de gestion de base de données (SGBD) : MySQL, MariaDB, PostgreSQL, Oracle...
- Attention, l'environnement en ligne et celui en local devront correspondre...

PHP clé en main

Il existe des solutions "clé en main" qui intègrent Apache, MySQL et PHP ainsi que PHPMyAdmin pour gérer la DB.:

- [LAMP](#) (Linux) / [WAMP](#) (Windows) / [MAMP](#) (Mac).
- [XAMPP](#) (Linux / Windows / Mac).

PHP: avez-vous questions?

