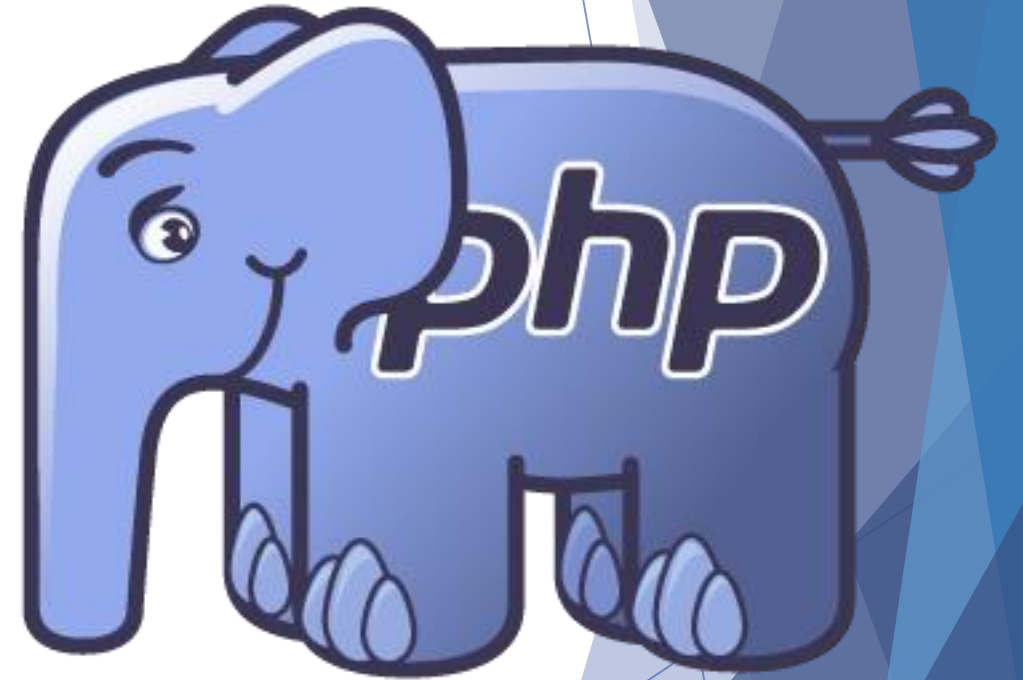




# INTRODUCTION

PHP (Hypertext Preprocessor) est un langage de script principalement utilisé pour le développement web côté serveur. Conçu à l'origine pour créer des pages web dynamiques, PHP s'intègre facilement avec le HTML, ce qui en fait un choix populaire pour les développeurs web souhaitant créer des sites interactifs et dynamiques.



# caractéristiques principales de PHP

## 1. Langage côté serveur

PHP est principalement exécuté sur le serveur. Le code PHP est traité côté serveur, et le résultat est renvoyé sous forme de HTML au navigateur de l'utilisateur, rendant le processus transparent pour l'utilisateur final.

## 2. Simplicité d'utilisation

PHP est simple à apprendre et à utiliser. Sa syntaxe est intuitive et proche des langages comme C, Java ou Perl, ce qui facilite son adoption par les développeurs.

## 3. Intégration facile avec HTML

PHP s'intègre facilement au code HTML, ce qui permet de générer du contenu dynamique sans quitter la structure HTML d'une page. On peut insérer du PHP dans un fichier HTML entre des balises `<?php ... ?>`.

## 4. Compatibilité avec une large gamme de bases de données

PHP prend en charge de nombreuses bases de données, comme MySQL, PostgreSQL, Oracle, et SQLite, facilitant la gestion des données dans les applications web.

## 5. Multiplateforme

PHP est compatible avec différents systèmes d'exploitation, comme Windows, Linux, macOS, et Unix. Les scripts PHP peuvent être exécutés sur divers environnements de serveur web (Apache, Nginx).

## 6. Extensibilité

PHP dispose d'une vaste bibliothèque d'extensions et de frameworks (comme Laravel, Symfony), offrant une flexibilité considérable pour ajouter des fonctionnalités complexes (gestion des sessions, cryptage, traitement des fichiers, etc.).

## 7. Gestion des sessions

PHP facilite la gestion des sessions, permettant aux développeurs de conserver les informations des utilisateurs entre différentes pages web (utilisé pour les paniers d'achat, les connexions utilisateurs, etc.).

## 8. Support d'une large communauté

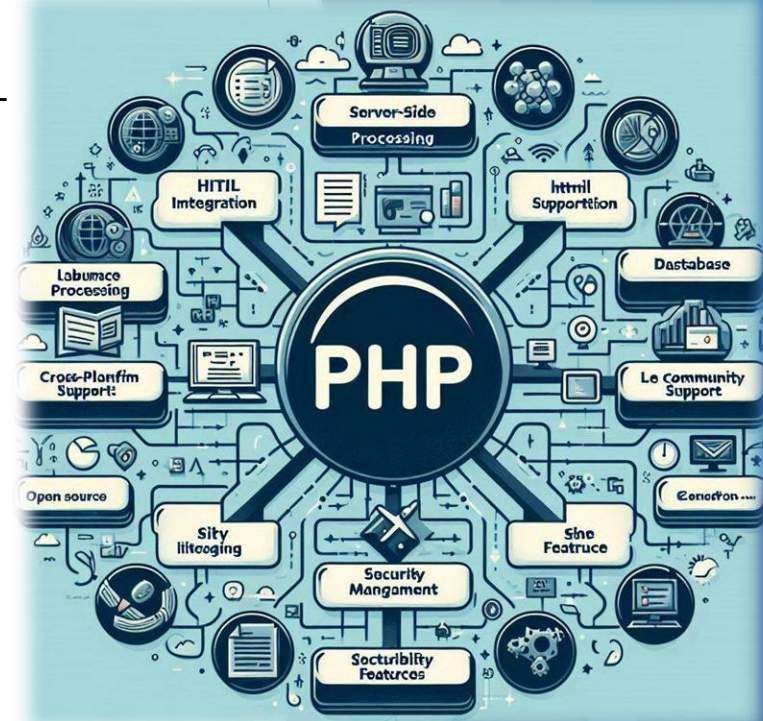
Grâce à son adoption massive, PHP bénéficie d'une grande communauté active qui produit des bibliothèques, des frameworks, et des ressources utiles, avec un support étendu et une abondance de documentation.

## 9. Code open source et gratuit

PHP est un projet open source, libre d'utilisation et de modification, ce qui le rend accessible à tous sans frais supplémentaires.

## 10. Sécurité

Bien que le code PHP puisse être vulnérable s'il est mal utilisé, il propose plusieurs fonctions et outils intégrés pour protéger les applications web (cryptage, validation des entrées, gestion des erreurs, etc.).



# Fonctionnement

## 1. Requête du navigateur

Lorsque l'utilisateur accède à une page web contenant du code PHP, il envoie une requête HTTP (généralement via un navigateur) vers le serveur web qui héberge cette page.

## 2. Intervention du serveur web

Le serveur web (comme Apache, Nginx, ou IIS) reçoit la requête et détecte qu'elle contient du code PHP grâce à l'extension .php du fichier. Le serveur transmet alors la requête au moteur PHP pour exécuter le code.

## 3. Exécution du code PHP

Le moteur PHP traite le fichier PHP ligne par ligne. Il exécute toutes les instructions PHP qu'il contient, comme :

- Interagir avec une base de données,
- Manipuler des fichiers,
- Gérer des sessions,
- Effectuer des calculs ou des traitements.

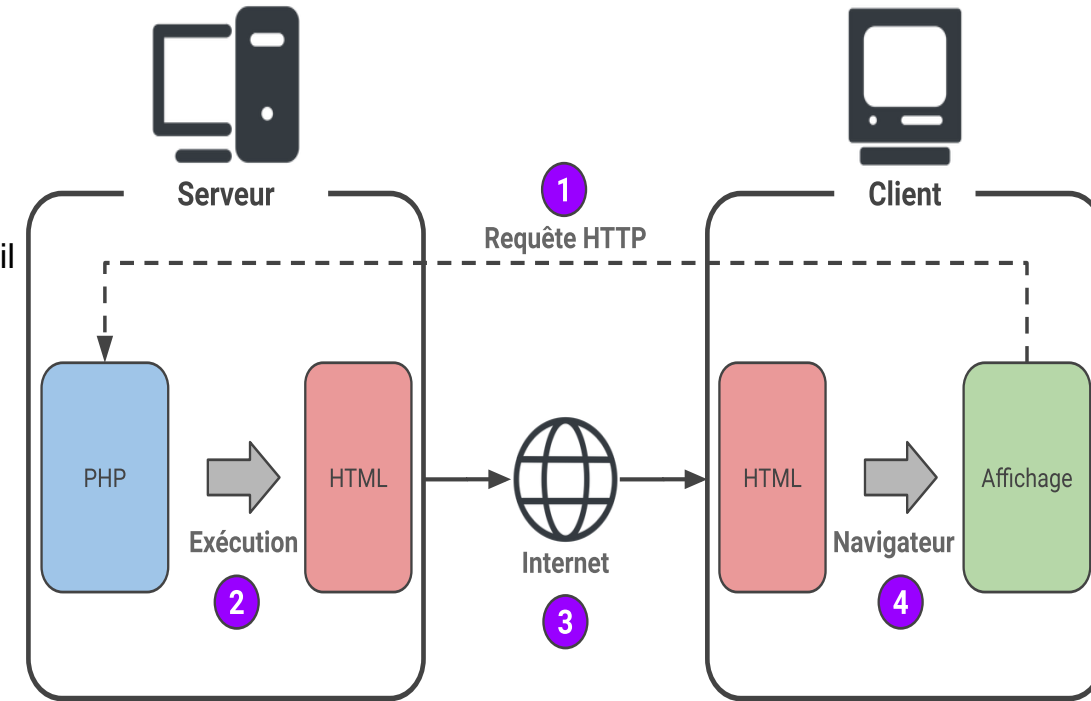
PHP génère ensuite du **code HTML** basé sur les instructions du script.

## 4. Envoi de la réponse

Une fois l'exécution terminée, le serveur web renvoie la page HTML générée par le PHP au navigateur. Ce HTML est alors affiché comme une page web statique dans le navigateur de l'utilisateur.

## 5. Le rôle de l'utilisateur final

L'utilisateur final ne voit jamais le code PHP exécuté, seulement le résultat (généralement du HTML, CSS, et JavaScript) qui est affiché dans le navigateur.



# Installation

## 1. Choisir un environnement de serveur

PHP nécessite un serveur web pour s'exécuter. Tu peux choisir parmi les options suivantes :

### •Serveur local :

- XAMPP** : Un package qui inclut Apache, MySQL, PHP, et Perl.
- MAMP** : Similaire à XAMPP, mais spécialement conçu pour macOS.
- WampServer** : Un environnement de développement sous Windows qui inclut Apache, PHP, et MySQL.

## 2. Télécharger et installer PHP

•Si tu choisis d'utiliser un serveur local comme XAMPP ou MAMP :

- XAMPP** : Télécharge-le depuis [apachefriends.org](http://apachefriends.org)
- MAMP** : Télécharge-le depuis [mamp.info](http://mamp.info).

•Pour une installation manuelle :

- Télécharge la dernière version de PHP depuis [php.net](http://php.net).
- Suis les instructions d'installation fournies dans le fichier INSTALL.

## 3. Configurer le serveur

•Si tu utilises XAMPP, MAMP ou WampServer, l'installation configure généralement Apache pour exécuter PHP automatiquement.

•Pour une installation manuelle, tu devras configurer le serveur web (comme Apache) pour traiter les fichiers PHP :

- Ajoute les directives nécessaires dans le fichier de configuration du serveur (httpd.conf pour Apache).





# SYNTAXE

Un script PHP commence par le `<?php` et se termine par `?>`.  
Le délimiteur PHP `<?php` et `?>` dans l'exemple suivant indique simplement au moteur PHP de traiter le bloc de code inclus comme du code PHP, plutôt que comme du HTML simple.

```
<?php
// le code à exécuter
echo "Hello, world!";
?>
```

