Anthony Laplane

PHP Théorie

studi

Introduction

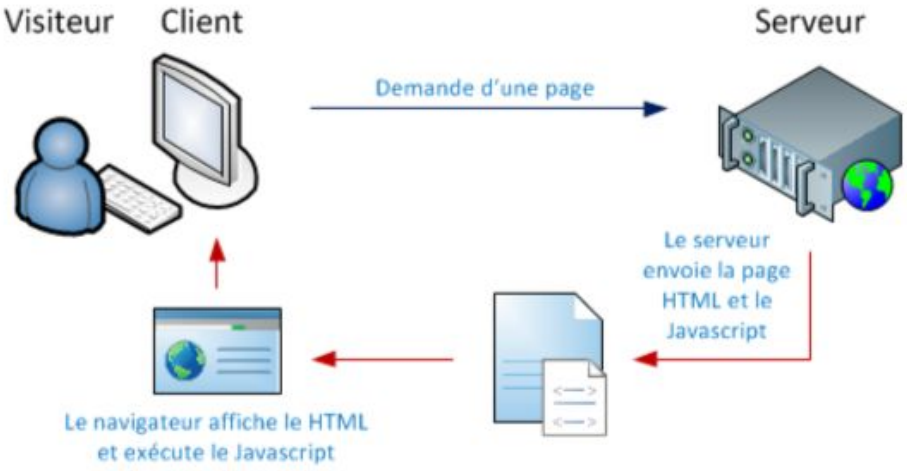
PHP est un langage de programmation tout comme javascript à la différence que javascript est un langage client (lorsqu'il est utilisé sur une page web) alors que php est un langage serveur.

Langage exécuté côté client : Lorsque le code est exécuté, cela se passe sur l'ordinateur de l'utilisateur (le client)

Langage exécuté côté serveur : Lorsque le code est exécuté, cela se passe sur le serveur

Introduction

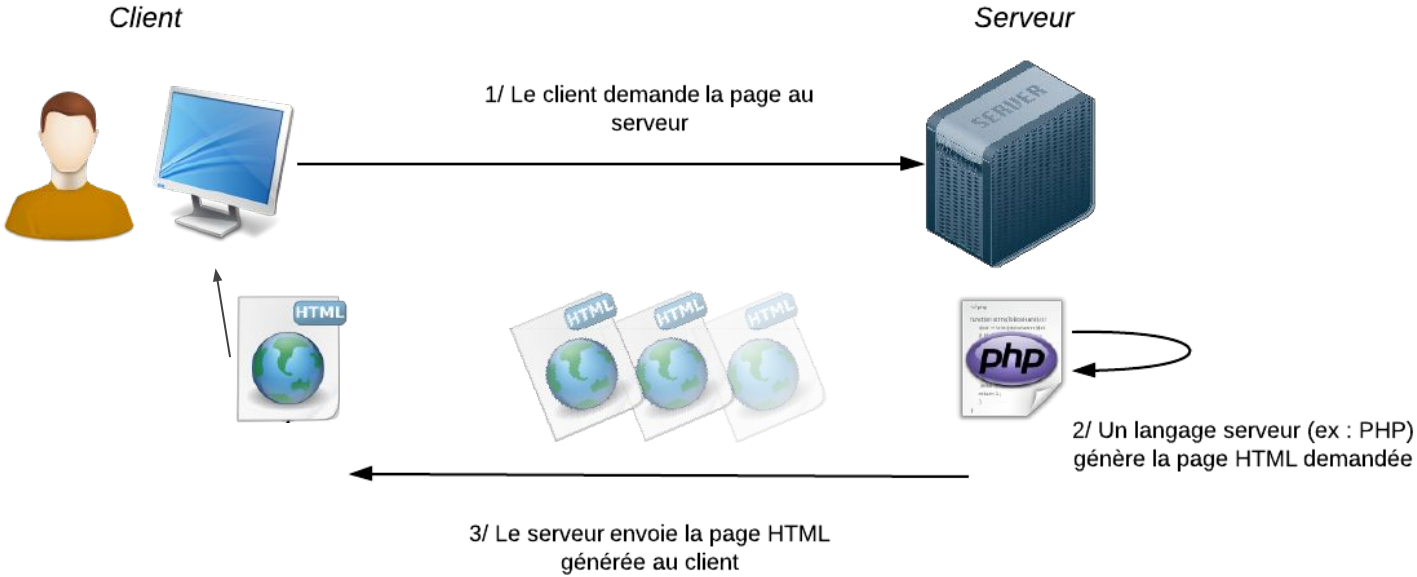
C’est parce que javascript est exécuté côté client qu’il peut modifier la page dans le navigateur (pour ajouter de l'interactivité).



JavaScript est un langage dit client-side, c'est à dire interprété par le client (le navigateur).

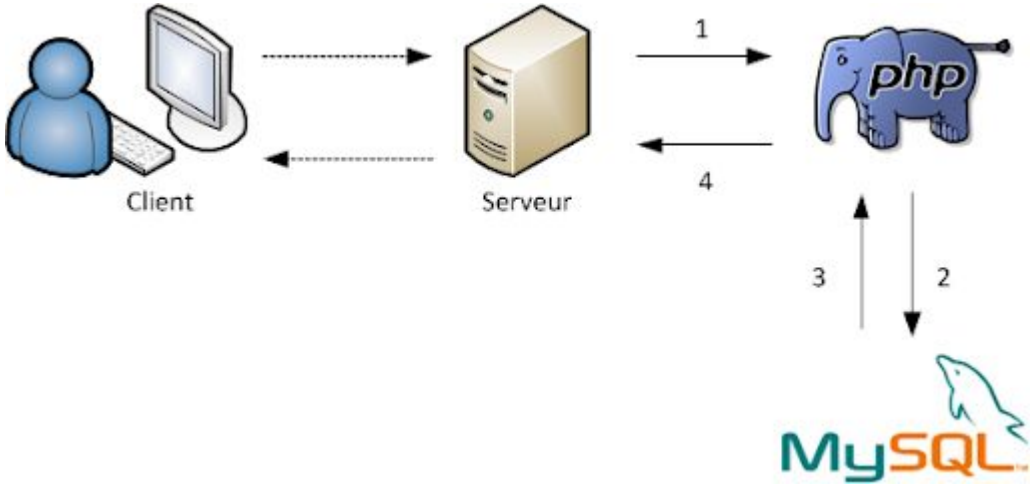
Introduction

PHP va donc être exécuter sur le serveur, avant même que la page soit affichée à l'écran.



Introduction

Pour fonctionner, php a besoin d’un serveur web (apache) et d’une base de données (dans la plupart des cas).



Introduction

Le langage php permet de créer des sites dynamiques avec des données issues d'une base de données (ex: mysql).

La plupart des CMS utilisent le langage PHP (Wordpress, Drupal, Prestashop, Joomla etc.).

Plus de 75% des sites utilisent PHP.

Quelques alternatives à PHP :

Fichier php

Un fichier php se termine par l'extension php (ex: page.php).
Le code php que l'on va écrire dans un fichier php va être entre une balise d'ouverture **<?php** et de fermeture **?>**

```php
<?php

//je peux écrire entre les balises le code php

?>
```

Fichier php

Un fichier php va souvent mélanger du php et du html. Le html doit être écrit en dehors des balises php

```
<h1>Titre en html</h1>

<?php

//je peux écrire entre les balises le code php

?>

<h2>Titre en html</h2>
```

Echo

L’instruction echo va nous permettre d’afficher un message sur la page

```php
<?php
echo 'Bonjour';
// ou
echo('Bonjour');
?>
```

Commentaires

Comme en javascript, on peut ajouter des commentaires dans notre code pour le rendre plus lisible.

```php
<?php
// Commentaire sur une ligne


/*
 Un commentaire qui pourra
 contenir plusieurs ligne
*/
?>
```

Variables

Une variable est une zone mémoire qui pourra contenir une valeur. C’est une sorte de boîte dans laquelle on va pouvoir stocker une information.

```php
<?php
// On déclare une variable age avec comme valeur 16
$age = 16;

// On affiche le contenu de la variable age sur la page avec echo
echo $age;

?>
```

Constantes

En PHP, une constante est une valeur qui ne peut pas être modifiée pendant l'exécution du script. Une constante est définie grâce à la fonction "define()", qui prend deux arguments : le nom de la constante et sa valeur.

```php
<?php

   // Une constante n'est défini qu'une seule fois
   define("_APP_VERSION_", "1.2.6");

   echo _APP_VERSION_;


?>
```

Instruction if

Les conditions en PHP sont définies grâce à des mots-clés tels que "if" (si), "else" (sinon) et "elseif" (sinon si).

```php
<?php
// On déclare une variable age avec comme valeur 16
$age = 16;

// On teste l'âge. Si l'âge est supérieur à 18, on affiche un message, sinon un autre message
if ($age >= 18) {
  echo 'adulte';
} elseif ($age >= 13) {
  echo 'ado';
} else {
  echo 'enfant';
}
?>
```

Instruction switch

switch permet de comparer une expression à une liste de valeurs. Elle peut être utilisée de manière similaire à la structure "if"/"elseif"/"else", mais peut être plus concise et plus lisible dans certains cas.

```php
<?php
$numJour = 3;

switch ($numJour) {
    case 1:
        echo "Lundi";
        break;
    case 2:
        echo "Mardi";
        break;
    case 3:
        echo "Mercredi";
        break;
    default:
        echo "Numéro de jour invalide";
}
?>
```

Les opérateurs

- Opérateurs de comparaison de valeur :
  - ">" (strictement supérieur à)
  - "<" (strictement inférieur à)
  - ">=" (supérieur ou égal à)
  - "<=" (inférieur ou égal à)
  - "==" (égal à) ou "===" (pour vérifier le type)
  - "!=" (différent de) ou "!==" (pour vérifier le type)
- Opérateurs logiques en PHP :
  - "&&" (ET logique)
  - "||" (OU logique)
- Opérateurs arithmétiques :
  - "+" (addition)
  - "-" (soustraction)
  - "*" (multiplication)
  - "/" (division)
  - "%" (modulo)

Les tableaux

Un tableau (array en anglais) est une structure de données qui permet de stocker une liste d'éléments de manière organisée. Un tableau peut contenir des éléments de n'importe quel type (chaînes de caractères, nombres, objets, etc.).

```php
<?php
    // Tableau simple
    $fruits = array("Banane", "Orange", "Pomme");
    echo $fruits[0];  // Affiche "Banane"

    // Tableau associatif
    $fruits2 = array("Banane" => "Jaune", "Orange" => "Orange", "Pomme" => "Verte");
    echo $fruits2["Banane"];  // Affiche "Jaune"
?>
```

Boucle foreach

La boucle "foreach" permet de parcourir un tableau et d'exécuter un bloc de code pour chaque élément du tableau. La boucle "foreach" prend en compte les clés et les valeurs du tableau.

Exemple sans clé

```php
<?php
    $fruits = array("Banane", "Orange", "Pomme");

    foreach ($fruits as $fruit) {
        echo "$fruit";
    }
?>
```

Boucle foreach

Exemple avec clé

```php
<?php
    $fruits = array("Banane" => "Jaune", "Orange" => "Orange", "Pomme" => "Verte");

    foreach ($fruits as $fruit => $couleur) {
        echo "$fruit est de couleur $couleur\n";
    }
?>
```

Boucle for

La boucle "for" permet d'exécuter un bloc de code plusieurs fois de manière itérative. La boucle "for" se compose de trois parties : l'initialisation, la condition de fin de boucle et l'incrémentation.

```php
<?php
    for ($i = 1; $i <= 10; $i++) {
        echo "$i ";
    }
    // Affiche : 1 2 3 4 5 6 7 8 9 10
?>
```

Boucle while

La boucle "while" permet d'exécuter un bloc de code de manière itérative tant qu'une condition est vraie. Elle se compose d'une condition qui est évaluée avant chaque itération de la boucle.

Exemple sans clé

```php
<?php
    $i = 1;
    while ($i <= 10) {
        echo "$i ";
        $i++;
    }
    // Affiche : 1 2 3 4 5 6 7 8 9 10
?>
```