

Vielle PHP



FERY Kevin BTS2

SOMMAIRE

Introduction	page 3
Histoire	page 4
Fonctionnement	page 5
Langage	pages 6 -11
Syntaxe de base	page 6
Variables	page 6
Affecter des valeurs	page 6
Constantes	page 7
Opérateurs	page 7
Tableaux	pages 7 - 8
Boucles	pages 8 - 10
Fonctions	pages 10 - 11
Autres possibilités	page 12

Introduction

Tout d'abord PHP est l'acronyme de Hypertext Preprocessor. C'est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques (page générée à la demande en fonction d'information) via un serveur HTTP. PHP est un langage impératif orienté objet. Il s'agit d'un langage de script interprété côté serveur.

Il est considéré comme une des bases de la création de sites web dits dynamiques mais également des applications web. PHP a permis de créer un grand nombre de sites web célèbres, comme Facebook ou encore Wikipédia.

Le logo considéré comme officiel du PHP est l'éléPHPant avec l'acronyme PHP sur son côté. Il est créé en 1998 par Vincent Pontier.

Histoire

Le langage PHP fut créé en 1994 par Rasmus Lerdorf pour son site web. L'objectif premier de ce langage était de conserver une trace des visiteurs de son CV. PHP voulait initialement dire Personal Home Page tools. PHP est un langage inspiré du langage Perl et du langage C. Ce dernier réalise les 2 premières versions de ce langage (PHP 1 et PHP 2). Il décida de publier le code source de PHP en 1996.

C'est un an plus tard en 1997 que sorti PHP 3, en effet, deux étudiants au nom de Andi Gutmans et Zeev Suraski ont repris le code source publié un an auparavant pour en redévelopper le cœur de ce langage, ce qui renomma PHP en ce qu'on connaît aujourd'hui ; en Hypertext Preprocessor. Ils redéveloppèrent également le moteur interne de PHP aboutissant à Zend Engine.

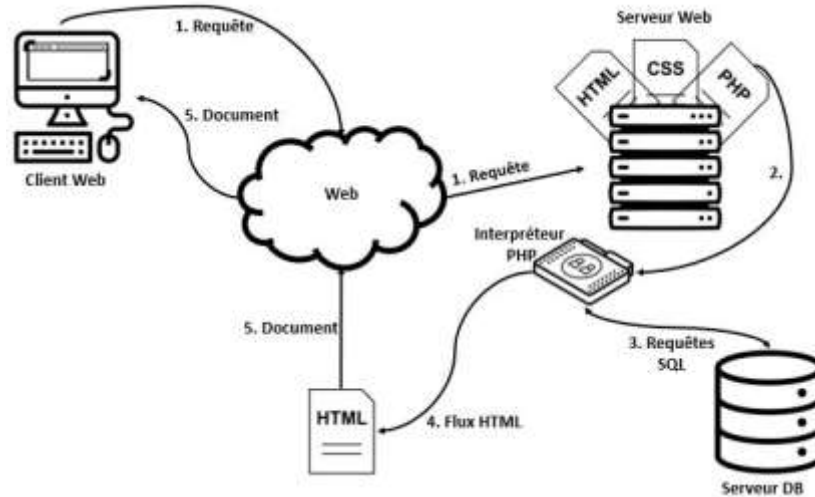
Puis PHP 4 fût officiellement publié en 2000, cette évolution se basant sur le Zend Engine apportait de meilleurs serveurs, des serveurs HTTP, et une sécurisation accrue. La version 5 de PHP est, quant à elle, sortie en 2004, elle utilise Zend Engine 2. La différence majeure avec la précédente version et la création d'un véritable mode de programmation d'objet et des fonctionnalités de gestion pour les entreprises.

Cependant, il n'y a jamais eu de version PHP 6, en effet, cette version devait être développée en Unicode, or le projet ayant pris trop de retard et ayant des soucis de développement, n'eut jamais lieu.

Ensuite, la version 7 de PHP est parue en 2015. En plus de nouvelles fonctionnalités, cette version utilise un nouveau modèle de moteur appelé PHP-NG ou Next Generation, ce nouveau moteur nécessite moins de serveurs pour desservir le même nombre d'utilisateurs qu'auparavant.

Enfin, aujourd'hui PHP est sous sa 8^{ème} versions. Parue en 2020, cette dernière rajoute de nombreuses fonctionnalités. Ce langage à grand succès est utilisé par des dizaines de millions de sites Web à travers le monde tels que Le Monde, Facebook ou encore Yahoo. PHP est maintenant utilisé dans de nombreux CMS (système de gestion de contenu) comme Prestashop ou Magento et dans de nombreux frameworks tels que Symfony, Zend ou encore Laravel par exemple.

Fonctionnement



Premièrement, le client, le plus souvent un navigateur web envoie une requête http via une URL vers un serveur (1).

Puis, cette requête est traitée par le serveur web, il identifie le type de document à renvoyer : s'il s'agit d'un document sans interprétation (ici sans PHP) alors il va le renvoyer immédiatement, sinon, le document a besoin d'une interprétation alors il va le traiter (2).

Ensuite, lors du traitement du document, ce dernier est interprété pour vérifier si fait appel à des données externes provenant d'un serveur de bases de données (3).

Puis, l'interprétation de la page PHP produit une page, HTML fournit au serveur, lisible côté client (4).

Enfin, le serveur Web renvoie cette page / document au client pour l'affichage (5).

Langage

Le langage étudié ici est une partie de l'immense possibilité de ce que propose PHP en procédurale. Nous n'étudierons pas d'objet. La documentation officielle de PHP est consultable à l'adresse suivante : « php.net/docs.php ».

Syntaxe de base :

La syntaxe du PHP ressemble à celles des langages Java, Perl et du C. Cependant, le PHP peut facilement être mélangé avec du code HTML au sein d'un fichier PHP.

Le PHP est délimité par les balises ouvrantes « <?php » et fermante « ?> ». Ces balises permettent de délimiter quand le serveur doit interpréter le code.

Comme en C ou en Perl, les instructions doivent se finir par des points-virgules à la fin de chaque instruction.

PHP supporte également plusieurs types de commentaires tels que les commentaires sur une ligne « // » ou encore « # » et les commandes sur plusieurs lignes « /* */ ».

Les variables sont typées et doivent être déclarées avec le symbole « \$ » avant chaque utilisation.

Variables :

Il existe une multitude de types de variables en PHP en voici des exemples :

- Boolean (true or false) : vrai ou faux
- Number (integer, float et les opérateurs de calcul) : nombre simple et décimal
- String (string et opérateur de concaténation « . ») : chaîne de caractères
- Array : tableau
- Object : Programmation Orienté Object (type spécial)
-

A noter que « var_dump » permet d'afficher les informations d'une variable : var_dump(\$variable).

Affecter des valeurs :

Il est possible d'affecter des valeurs à des variables :

```
< ?php
    $a = 'bonjour' ;
?>
```

Constantes :

Une constante est un identifiant qui représente une valeur simple. La valeur d'une constante ne peut jamais être modifiée durant l'exécution du script. Par convention, les constantes sont toujours écrites en majuscules car elles sont sensibles à la casse (différent en fonction des majuscules, exemple : nom différent de Nom).

```
<?php
    define(«NOM », « CONTENU »);
?>
```

Pour appeler la constante :

```
echo NOM ;
```

Cela va afficher :

CONTENU

Opérateurs :

Opérateurs de comparaison :

« == » = égal
« === » = identique (avec le même type)
« != » = différent
« < > » = différent
« !== » = différent
« < » = plus petit
« > » = plus petit
« <= » = inférieur ou égal
« >= » = supérieur ou égal
« <=> » = combiné (disponible depuis PHP 7) :
inférieur/égal/supérieur

Opérateurs logiques :

« ! » = différent
« && » = et
« || » = ou
« xor » = soit l'un soit l'autre mais pas les deux
« or » = ou

Tableaux :

Pour définir un tableau on utilise le mot-clé « array » ou les « [] ».

```
<?php
    $chiffre = array(1,2,3,4,5,6,7,8,9);
?>
```

Un tableau peut également avoir plusieurs dimensions, c'est-à-dire qu'un tableau peut contenir plusieurs tableaux à l'intérieur de lui-même.

```
<?php
    $tab = array(
        array(« premier_tableau »)
        array(« deuxième_tableau »)
```

```
);  
?>
```

Boucles :

If :

Les boucles sont les bases de l'algorithmie, ici, une expression est convertie en sa valeur booléenne. Si l'expression vaut TRUE, alors PHP exécutera l'instruction, mais si l'expression vaut FALSE, alors l'instruction sera ignorée.

```
<?php  
    If(condition){  
        instruction :  
    }  
?>
```

Else :

Les boucles « if » peuvent être suivies par un « else ». Il permet de continuer l'algorithme si le « if » exécuté précédemment a échoué.

```
<?php  
    If(condition){  
        instruction :  
    }  
    else{  
        instruction2;  
    }  
?>
```

Switch :

Il existe les « switch », cette boucle permet de comparer une variable à une série de cas. Le but de cette boucle est de limiter les « if-else » et donc d'optimiser le code.

```
<?php  
    $i = 1 ;  
    switch($i){  
        case 0 :  
            echo « i égal à 0 » ;  
            break ;  
        case 1 :  
            echo « i égal à 1 » ;  
            break ;
```



```

        case 2 :
            echo « i égal à 2 » ;
            break ;
        default :
            echo « i n'est ni égal à 0, ni à 1, ni à 2. » ;
    }
?>

```

Ici on compare la variable « i » qui est égal à 1. L'objectif de ce « switch » est de trouver sa valeur. Il y a le cas où, « i » est égal à 0, puis à 1 et enfin à 2. A la fin de chaque « switch », il y a une valeur par défaut si aucune condition ne rentre dans l'algorithme.

While :

La boucle « while » signifie que l'instruction passée dans la boucle va être répétée tant que la condition ne sera pas remplie. A noter que l'algorithme rentre dans la boucle uniquement si la condition est correcte.

```

<?php
    $i = 1;
    while($i <10){
        echo $i . « est plus petit que 10 » ;
        $i++ ;
    }
?>

```

La boucle va donc s'arrêter quand la variable « i » sera plus grande que 10. « \$i++ » permet d'incrémenter la valeur de la variable, soit d'ajouter 1 à chaque tour de boucle.

Do While :

Les boucles « do while » ressemblent aux boucles « while », c'est le même fonctionnement, cependant, contrairement à la boucle « while » celle-ci s'exécute tout le temps, elle n'a pas besoin de rentrer dans des conditions pour s'exécuter.

```

<?php
    $i = 1;
    do{
        echo $i . « est plus petit que 10 » ;
        $i++ ;
    } while($i <10) ;
?>

```

For :

Les boucles « for » sont généralement utilisées quand le nombre d'itérations est connu à l'avance. La boucle est quoiqu'il arrive exécuter pour la première fois. Pour que la boucle continue de s'exécuter, elle doit répondre à des conditions.

```
<?php
    for($i=1 ; $i < 10 ; $i++)
    {
        echo $i . « est plus petit que 10 » ;
    }
?>
```

Ici, la variable « i » est déclarée et initialisée à 1 directement dans la boucle, la condition est tant que « i » est plus petit que 10 et il y a l'incréméntation pour « i » obtienne plus un à chaque tour de boucle.

Foreach :

La boucle « foreach » fonctionne principalement que sur les tableaux ou les objets. Cette boucle permet de parcourir ces éléments.

```
<?php
    foreach($nomTableau as $nomVariable){
        echo $nomVariable;
    }
?>
```

Cette boucle permet de parcourir un tableau grâce à une variable appelée « nomVariable ». Cette variable permet d'exécuter des instructions, ici, on affiche tous les éléments du tableau.

Fonctions :

Il est possible d'écrire des fonctions en PHP. Les fonctions n'ont pas besoin d'être définies avant d'être utilisées et elles sont insensible à la casse.

```
<?php
    function Nom(argument){
        instruction;
        return ;
    }
?>
```

Il est également possible d'inclure des boucles, des tableaux et tous autres d'éléments procédurales dans une fonction. De plus, qu'une fonction retourne un résultat. Il existe des fonctions appelées des méthodes qui sont identiques à des fonctions, sauf qu'elle ne retourne rien.

A noter qu'il existe beaucoup de fonctions déjà pré-écrites par PHP comme par exemple « strisop() » qui recherche la position de la première occurrence dans une chaîne ou encore « str_replace() » qui remplace toutes les occurrences de la chaîne de recherche par la chaîne de remplacement.

Fonctions de test :

Il existe de multiples fonctions utiles pour les opérateurs de tests comme par exemple la fonction « isset » permettant de déterminer si la variable définie est NULL ou encore la fonction « empty » qui détermine si la variable est vide ou non.

Autres possibilités

Comme dit précédemment, PHP est un langage dit dynamique, ce dernier permet de créer ce que l'on appelle des CRUD (création, lecture, insertion et modification / suppression). Ce langage est directement relié au SQL ce qui permet d'exécuter des requêtes permettant de gérer des bases de données pour créer des sites et applications de gestion.

Le PHP permet également de sécuriser un formulaire que ce soit contre des attaques pirates ou contre tous types de problèmes liés à la sécurité. Il permet par exemple de chiffrer un mot de passe (en Argon2 ou encore en sha256) pour sécuriser tout type de connexion.