

# Programmation WEB avec PHP

Matière : Programmation WEB avec PHP

Date : 13.06.2022

Formateur : Malerba Sylvain

Auteur : Malerba Sylvain

# SOMMAIRE

01 Introduction

02 Le PHP

03 Les variables

04 Les boucles et conditions

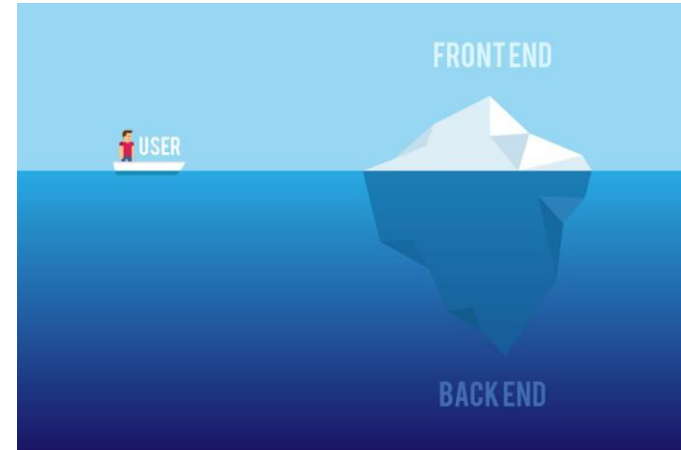
**01**

# **Introduction**

# 01

## Front-end / Back-end

- **Front-end:** Partie visible du site web par l'utilisateur
- **Back-end:** Partie non visible du site web par l'utilisateur ( Gestion des données, gestions des accès etc...)



Source: <https://www.leproductowner.com/>

# 01

## Front-end/Back-end: exemple de langage

- **Front-end**: HTML, CSS , Javascript etc...
- **Back-end (langage)** : PHP, Python, Java etc...
- **Back-end (Base de données)** : MySQL, MongoDB etc...



# 01

## Présentation du langage

- PHP Hypertext Preprocessor (Le premier « P » de PHP est lui-même l'abréviation de « PHP »)
- Langage créé en 1994
- Langage OpenSource
- Site officiel : <https://www.php.net/>



# Pourquoi le PHP?

- **Dynamisation de sites dynamiques**
  - Traitement des données d'un formulaire
  - Traitement des données en Base de données
  - Utilisé par de nombreux CMS ( Wordpress, Prestashop etc...)

# 01

# Pourquoi le PHP?

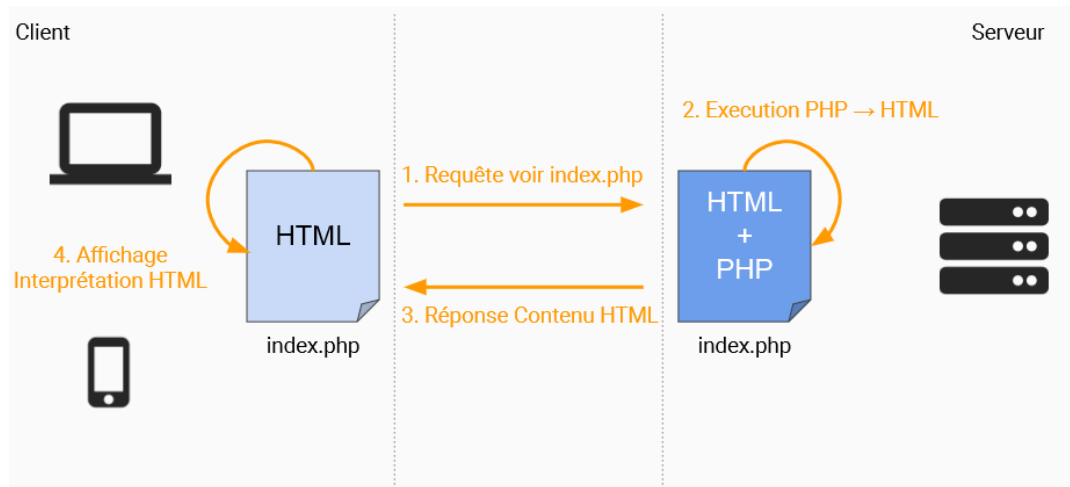
- **API**
  - Création d'une API
  - Utilisation d'une API externe
- **Création de fichiers exécutables (CLI)**



## 01

# Présentation du langage

- Le PHP est un langage de programmation Web exécuté côté serveur
- Si vous analysez le code source d'une page, vous ne trouverez jamais d'instructions PHP, mais seulement généralement du code HTML, CSS et JavaScript



01

# Mise en place de l'environnement



WAMP

<https://www.wampserver.com/>



Windows



MAMP

<https://www.mamp.info/en/downloads/>




MAC OS

# 01

## Mise en place de l'environnement



- Lancer le logiciel WAMP
- Vérifiez que l'ensemble des services soit actif 
- Se rendre dans le répertoire www et créer votre projet
- Ouvrir le localhost



- Créez un répertoire de travail pour votre projet (ex: /Users/Name/Aflokkat/PHP/)
- Configurez MAMP pour faire pointer le DocumentRoot sur ce répertoire
- Lancez le serveur (Bouton "Start")
- Ouvrez un navigateur sur l'adresse <http://localhost:8888/>

# 01

## Choix de l'éditeur



Visual Studio Code



Sublim Text



PHPStorm

- **Téléchargez et installez votre éditeur de code**
- **Configurer votre éditeur de code par défaut**

# 02

## Le PHP

# 02

## PHP : la syntaxe

- Le code PHP doit se situer dans des fichiers .php
- Du code HTML, CSS, Javascript peut être rédigé et interprété dans des fichiers .php (mais ce n'est pas la meilleure des pratiques)
- Le code PHP doit être encadré par « `<?php` » et « `?>` »
- La fonction **echo** permet d'afficher une variable, **print\_r** un tableau
- Le fichier **index.php** est exécuté par défaut dans chaque dossier

- Une instruction se termine par un « ; »
- Les chaînes de caractères sont encadrées par " ou '
- Les nombres n'ont pas besoin d'être encadrés
- Dans une chaîne de caractères, on utilise le caractère d'échappement \
- Les commentaires peuvent être écrits de la manière suivante:
  - // pour commenter une ligne
  - /\* \*/ pour commenter une section

# 02

## Exercice

- Créez un fichier PHP, avec du code HTML à l'intérieur duquel, vous allez y insérer en PHP, le message suivant:  
« **BONJOUR** *Ton prénom!* **Tu as ton âge** et tu es connecté! »



# 02

# Correction

```
<!DOCTYPE html>
<html>
  <head>
    <title>Aflokkat</title>
    <meta charset="utf-8" />
  </head>
  <body>
    <h2>Exercice 1</h2>

    <!-- Solution 1 -->
    <?php
      echo "BONJOUR Sylvain! Tu as 29 ans et tu es connecté! ";
    ?>

    <!-- Solution 2 -->
    <p>BONJOUR <?php echo "Sylvain";?>! Tu as <?php echo "29 ";?>ans et tu es connecté!</p>

  </body>
</html>
```

## Exercice 1

BONJOUR Sylvain! Tu as 29 ans et tu es connecté!

BONJOUR Sylvain! Tu as 29 ans et tu es connecté!

# PHP : la gestion des erreurs

- **En PHP, il est possible d'afficher les erreurs:**
  - **en configurant le php.ini, lorsque nous travaillons en local**
  - **en affichant les erreurs Apache, lorsque nous travaillons sur un serveur Apache**

# PHP : la gestion des erreurs

- Pour configurer le fichier php.ini : il suffit d'écrire la commande suivante dans un fichier PHP, puis d'ouvrir celui-ci dans un navigateur:

```
<?php phpinfo(); ?>
```

# PHP : la gestion des erreurs

- Les informations suivantes apparaissent:

PHP Version 7.2.14



System	Windows NT PC-DE-SYLVAIN 6.3 build 9600 (Windows 8.1 Home Premium Edition) AMD64
Build Date	Jan 9 2019 22:17:49
Compiler	MSVC15 (Visual C++ 2017)
Architecture	x64
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=c:\php-snap-build\deps_aux\oracle\x64\instantclient_12_1\sdk,shared" "--with-oci8-12c=c:\php-snap-build\deps_aux\oracle\x64\instantclient_12_1\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File ( <b>php.ini</b> ) Path	C:\Windows
Loaded Configuration File	C:\wamp64\bin\apache\apache2.4.37\bin\ <b>php.ini</b>

# PHP : la gestion des erreurs

- **Modifiez la configuration `display_errors = On` dans le fichier**

```
; The display of errors which occur during PHP's startup sequence are handled  
; separately from display_errors. PHP's default behavior is to suppress those  
; errors from clients. Turning the display of startup errors on can be useful in  
; debugging configuration problems. We strongly recommend you  
; set this to 'off' for production servers.  
; Default Value: Off  
; Development Value: On  
; Production Value: Off  
; http://php.net/display-startup-errors  
display_startup_errors = On
```

**Relancez le serveur Apache**

# 02 Exercice

- **Générez un code d'erreur en oubliant de fermer une guillemet par exemple.**

# 03 Les variables

# Les variables: la syntaxe

## Une variable :

- Commence par « \$ » et peut-être suivi d'une lettre ou d'un « \_ »
- Ne contient pas de caractères spéciaux
- Ne contient d'espace
- Est sensible à la casse



# 03

## Les variables: la syntaxe

```
<?php  
  
    $nom = "toto";  
    $age = 50;  
    $_prenom = "sylvain";  
    $Nom = "toto2";  
  
?>
```

Syntaxe correcte

```
<?php  
  
    $123nom = "toto";  
    $ag!e = 50;  
    $_pre nom = "sylvain";  
  
?>
```

Syntaxe incorrecte

# Les variables: le typage

**Une variable PHP peut stocker plusieurs types de données :**

- **String:** chaîne de caractère
- **Int:** nombre entier
- **Float:** nombre décimal
- **Boolean:** booléen (true ou false , 0 ou 1)
- **Array:** tableau
- **Object:** un objet PHP
- **NULL:** type null
- **Ressources**

# 03

## Les variables: le typage

```
<?php

    $nom = "Aflokkat"; //string
    $age = 24; // int
    $poids = 85.2; // float
    $cheveux = true; // boolean
    $language = array("Php", "HTML", "CSS"); // Array
    $sylvain = new Person(); // Object
    $var_null = null; // NULL

?>
```

# 03

## Exercice

- **A partir de l'exercice précédent, remplacez votre prénom et votre âge par une donnée stockée dans une variable PHP**

# 03

## Les variables: les opérateurs

Symbole	Signification
+	Addition
-	Soustraction
*	Multiplication
/	Division
%	Modulo

# Les variables: les opérateurs combinés

Symbole	Signification
.=	Concatène puis affecte le résultat
+=	Additionne puis affecte le résultat
-=	Soustrait puis affecte le résultat
*=	Multiplie puis affecte le résultat
/=	Divise puis affecte le résultat

# Les variables: les opérateurs comparaison

Symbole	Signification
==	Compare si deux valeurs sont égales
!=	Compare si deux valeurs sont différentes
<	Vérifie si une valeur est inférieure à une autre
>	Vérifie si une valeur est supérieure à une autre
===	Compare si deux valeurs sont égales en prenant en compte le typage

# 03

## Les variables: la concaténation

```
<?php

    $nom = "Malerba";
    $_prenom = "sylvain";

    echo $nom.$_prenom."<br>";
    echo $nom." ".$_prenom;

?>
```

Code

Malerbasylvain  
Malerba sylvain

Résultat



# 04 Les conditions

```
if (condition) {  
    # code...  
}  
elseif (condition) {  
    # code...  
}  
else{  
    # code...  
}
```

# 04

# Les boucles

# 04 Exercice

**A partir de l'exercice précédent:**

- **Réalisez une concaténation de votre nom et prénom**
- **En fonction de la valeur de la variable \$age faites la condition suivante permettant de déterminer si une personne est mineure ou majeure**
- **Créez 3 variables \$histoire, \$math, \$anglais, et attribuez leur une note sur 20, puis réalisez une addition :**
  - Si la somme est supérieure à 30, affichez « J'ai mon année ! »

**BONUS: calculez la moyenne des 3 notes précédentes**

# 04

## Les conditions : les opérateurs logiques

Symbole	Signification
and ou &&	Renvoie true lorsque toutes les comparaisons sont validées
or ou	Renvoie true lorsque au moins une comparaisons est validée
!	Renvoie true lorsque la comparaison est fausse

# 04

## Les conditions : les opérateurs ternaires

*Condition ? Si condition vraie alors : Sinon*

```
echo $age >= 18 ? "je suis majeur" : "je suis mineur"
```

# Les conditions : les fonctions de vérification

- **isset()** : permet de vérifier qu'une variable existe
- **empty ()** : permet de vérifier qu'une variable est vide
- **is\_null()**: permet de vérifier si une variable est nulle

# 04

## Exercice

**Je fais les courses !**

**Déclarez 2 variables \$lait, \$farine et attribuez leur une quantité (nombre entier)**

**Implémentez le code suivant:**

- **Vérifiez que les 2 variables ne soient pas vides**
- **Comparer la quantité de lait par rapport à celle de farine (affichez un message pour chaque condition)**
- **Si la somme de ces quantités est supérieure à 0, créez une variable \$total puis l'afficher en fin de programme**

# 04

## Solution

```
<?php

$lait = 1;
$sucre = 1;

if(!empty($lait) && !empty($sucre)){

    if($lait > $sucre){
        echo "Il y plus de lait que de sucre";
    }elseif ($lait == $sucre) {
        echo "Il y a autant de lait que de sucre";
    }else{
        echo "Il y a moins de lait que de sucre";
    }

    if($lait+$sucre > 0){
        $total = $lait+$sucre;
    }

    if (isset($total)) {
        echo "<br>Il y a ".$total." ingrédient(s)";
    }
}

?>
```

Il y a autant de lait que de sucre  
Il y a 2 ingrédient(s)



# 04

## Les conditions : le switch

```
switch (variable) {  
    case 'value':  
        # code...  
        break;  
    case 'variable':  
        # code...  
        break;  
  
    default:  
        # code...  
        break;  
}
```

**En PHP, il existe plusieurs types de boucles**

- **while()**
- **do .. while()**
- **for()**
- **foreach**

# 04

## Les boucles :while

```
while ( $condition ) {  
    # code...  
}
```

**ATTENTION: vérifiez toujours qu'il y ait une fin à la condition**

# 04

## Les boucles :do ...while

```
do {  
    # code...  
} while ( $condition);
```

**ATTENTION: vérifiez toujours qu'il y ait une fin à la condition**

# 04

## Les boucles :for

```
for ($i=0; $i < ; $i++) {  
    # code...  
}
```

# 04

## Les boucles :foreach

```
foreach ($variable as $key => $value) {  
    # code...  
}
```

### Cas n°1

```
foreach ($variable as $value) {  
    # code...  
}
```

### Cas n°2

# 04

## Exercice

**Dans cet exercice, nous allons afficher une grille de départ de 20 pilotes de F1 (balise <table>) avec la consigne suivante:**

- **Utilisez une boucle while pour afficher les 10 premiers pilotes**
- **Utilisez une boucle for() pour afficher les 10 derniers pilotes**
- **2 pilotes par lignes (à gauche les nombres impaires, à droite les paires)**

**Bonus: afficher en rouge les pilotes paires et en bleu les pilotes impaires**

# 04

# Solution

```
<?php
$nb_pilote = 20;
$pilote = 1;

$pair = "red";
$impair = "blue";

echo "<table><tr>";

while ( $pilote <= 10) {
    if($pilote % 2 == 0){
        echo "<td style='color: ".$pair."'> Pilote n°".$pilote."</td></tr>";
    }else{
        echo "<tr><td style='color: ".$impair."'> Pilote n°".$pilote."</td>";
    }
    $pilote ++;
}

for ($i=$pilote; $i <= $nb_pilote ; $i++) {
    if($i % 2 == 0){
        echo "<td style='color: ".$pair."'> Pilote n°".$i."</td></tr>";
    }else{
        echo "<tr><td style='color: ".$impair."'> Pilote n°".$i."</td>";
    }
}

echo "</tr></table>";
```