**Fiche mémo – PHP MySQL**

Véronique ROUAULT

# INSTALLATION du serveur php en local

* Pour créer des **sites web dynamiques**, nous devons installer des outils qui transformeront notre ordinateur en serveur afin de pouvoir tester notre site.
* Les principaux outils dont nous avons besoin sont :
  + **Apache** : le serveur web ;
  + **PHP** : le programme qui permet au serveur web d'exécuter des pages PHP ;
  + **MySQL** : le logiciel de gestion de bases de données.
* Bien qu'il soit possible d'installer ces outils séparément, il est plus simple pour nous d'installer un paquetage tout prêt : WAMP sous Windows (MAMP sous Mac OS X ou XAMPP sous Linux).
* **WAMP** est l’acronyme de **Windows Apache MySQL PHP**.
* Il est conseillé d'utiliser un éditeur de texte qui colore le code source comme Sublime Text pour programmer convenablement en PHP.  
  Pour les personnes plus expérimentées qui travaillent sur de gros projets, PHPStorm est l’éditeur payant le plus utilisé.
* Le serveur local est appelé par l’adresse : [*http://localhost/*](http://localhost/)

# LES BASES TROUVÉES SUR CODECADEMY

Le code PHP peut être écrit directement dans le code HTML

Le PHP s’écrit **à l'intérieur des balises <?php et ?> et** il faut terminer la ligne par un **point-virgule.** À la fin de chaque déclaration (l’unité la plus petite d’un code autonome)

<?php

echo "Véronique !";

?>

L'opérateur de concaténation est juste un point (.)

echo "Véronique,"." "."veux-tu apprendre le PHP"." ?"; // écriture d’une chaîne de caractères

echo 17 \* 123; // écriture d’une opération numérique

$monNom="Véronique";

$monAge = 32; // déclaration d’une variable

# liste des operateurs de comparaison

* > Est supérieur à
* < Est inférieur à
* <= Est inférieur ou égal à
* > = Est supérieur ou égal à
* == Est égal à
* ! = Est différent de (est non égal à)

if (cette condition est vraie) {

     / / Faire ceci

   }

   else {

     / / Faire cela

   }

$monNom="Thomas";

$monAge=50

if ($monNom != "Véronique") {

echo "Je ne connais pas ton âge";

}

else {

echo "Ton nom est "."$monNom"." et tu as ". $monAge;

}

?>

**Syntaxe du Switch** : Une instruction switch est semblable à une instruction if / elseif / else car elle vous permet de déterminer le bloc de code à exécuter en fonction de plusieurs conditions.

switch (2) {

case 0:

echo 'La valeur est 0';

break;

case 1:

echo 'La valeur est 1';

break;

case 2:

echo 'La valeur est 2';

break;

default:

echo "La valeur n'est ni 0, ni 1, ni 2";

}

Une instruction switch est constituée du mot-clé switch, d'une variable à vérifier, et d'une paire d'accolades { }. Ici, nous vérifions la valeur de la variable $monNum.

1. Ensuite, nous avons un bloc case pour chaque comparaison. Par exemple           case 1:          echo "1";          break;      vérifie si $monNum est égal à 1. Si c'est le cas, il écrit echo "1", et on utilise break pour quitter l'instruction switch.
2. Dans le cas contraire, les prochains blocs case s'exécutent.
3. Si aucun bloc case n'est vérifié, le bloc default est exécuté.

Tombe à l’eau (falling through)  : ajouter des blocs switch les uns après les autres sans ajouter de break

$i = 5;

switch ($i) { *(doit être retiré si on utilise endswitch par un ; )*

case 0:

echo '$i est égal à 0.';

break;

case 1:

case 2:

case 3:

case 4:

case 5:

echo '$i est entre 1 et 5.';

break;

case 6:

case 7:

echo '$i est entre 6 et 7.';

break;

default:

echo "Je ne connais pas la valeur de \$i.";

} *// l’accolade de fin peut être remplacée par endswitch)*

Utilisation de "Endswitch". Le sucre syntaxique !

C'est ce qu'on appelle la syntaxe alternative. Elle existe pour fournir du [sucre syntaxique](http://http:/fr.wikipedia.org/wiki/Sucre_syntaxique)

Il n'y a aucune différence entre l'utilisation soit de la syntaxe avec accolades (premier exemple) ou de l'autre syntaxe alternative (deuxième exemple). La seconde fournit de la lisibilité (elle est plus facile à lire), donc elle est généralement utilisée lorsqu'on mélange du code HTML et du code PHP dans le même fichier.

<?php

$monNom="Romain";

switch($monNom):

case"Véronique":

echo "Tu t'appelles ".$monNom;

break;

case "Thomas":

echo "Tu ne t'appelles pas Véronique";

break;

echo "Bienvenue dans le groupe ".$monNom;

break;

default:

echo "qui es-tu ?";

endswitch

?>

# LES tableaux en php

<?php

$tableau = array("Oeuf", "Tomate", "Haricot", "Chips", "Saucisse");

echo $tableau [2] ; // affiche le 3eme élément du tableau : Haricot

echo $tableau {2} ; // notation identique aux crochets

?>

*Changement des éléments d’un tableau :*

<?php

$langages = array("HTML/CSS",

"JavaScript", "PHP", "Python", "Ruby");

$langages[2] = "C++"; // remplace le 3eme élément du tableau par C++

echo $langages[2]; // affiche le 3eme élément du tableau

?>

*Supprimer des éléments d'un tableau :*

<?php

   $tableau = array("rouge", "bleu", "vert");

   unset($tableau[2]); // supprime « vert » du tableau

?>

<?php

   unset($tableau); // supprime le tableau

?>

# LES boucles en php

for ($annee = 2004; $annee < 2050; $annee = $annee + 4) {

echo "<p>$annee</p>";

}

***Voila comment cela se passe :***

1. La boucle for commence par le mot-clé for. Cela indique à PHP de se préparer à boucler !
2. Ensuite, on trouve un ensemble de parenthèses (( )). À l'intérieur des parenthèses, nous indiquons trois choses à PHP, séparés par des points-virgules (;) : comment débuter la boucle ; la condition de répétition ; et ce qu'il faut faire entre la fin d'une boucle et la suivante (par exemple, compter de un en un).
3. Après la partie entre parenthèses, la partie entre accolades ({ }) indique à PHP le code à exécuter à chaque itération de la boucle.

<?php

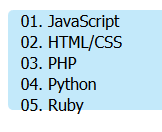
for($i=1; $i <= 100; $i=$i + 10) {

echo $i; // affiche les nombre de 10 en 10 jusqu’à 100 inclus

}

?>

***La boucle « foreach » :***

*<?php // affiche :*

*$langs = array("JavaScript", "HTML/CSS", "PHP", "Python", "Ruby");*

***foreach*** *($langs* ***as*** *$lang) {*

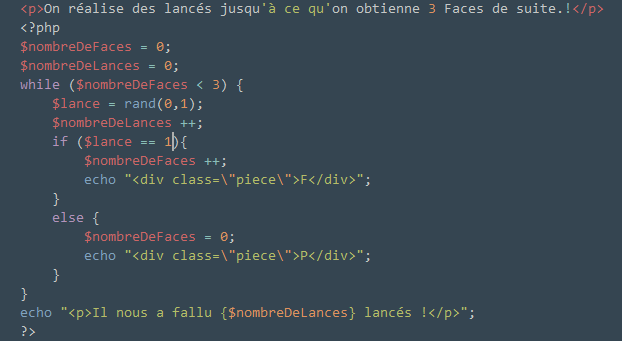
*echo "<li>$lang</li>";*

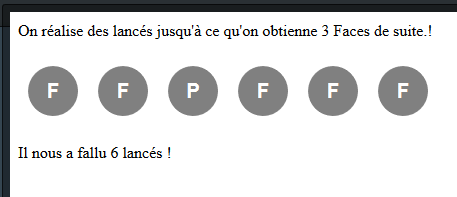
*}*

*unset($lang);* // supprime le tableau $lang

*?>*

***L’utilisation de la boucle « while » :***



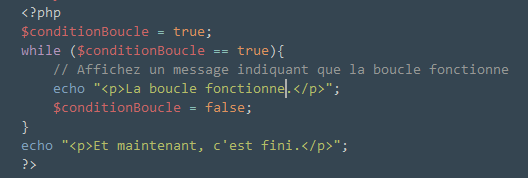
*Affichage :*

*Explications :*

Avant la boucle, les variables sont définies pour compter le nombre de faces consécutives ($nombreDeFaces), et le nombre total de lancés ($nombreDeLances).

La boucle s'exécute tant que $nombreDeFaces est inférieur à 3. A l'intérieur de la boucle, rand(0,1) ​​génère de façon aléatoire soit un 0 ou un 1 représentant respectivement le pile et le face, puis, on augmente $nombreDeLances de un pour connaître le nombre total de lancers. Si le résultat est face (1 est interprété comme un true - vrai) alors, on augmente $nombreDeFaces de un et une balise <div>F pour face est affichée dans à la page. Si ce n'est pas le cas (le tirage aléatoire a donné un "pile"), $nombreDeFaces revient à 0 et une balise <div>P pour pile est affichée dans à la page.

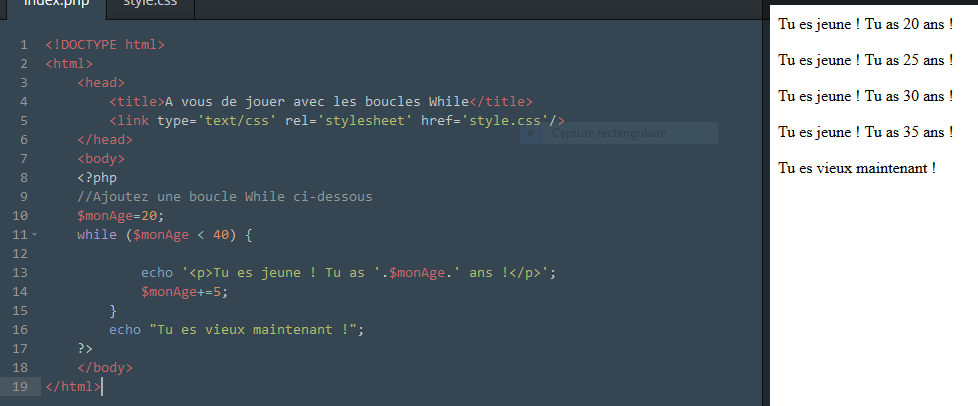
Lorsque la boucle est terminée, le nombre total de lancés ($nombreDeLances) est affiché dans la page au sein d'un paragraphe.



Il est très important, lors de l'écriture des boucles, de s'assurer que la boucle se terminera à un moment donné. Sinon, la boucle sera infinie

C'est pour éviter cela que nous mettons $conditionBoucle = false;

Si vous soumettez une boucle infinie dans un exercice, vous aurez besoin de recharger complètement la page pour l'arrêter.



<?php

$monAge=20;

while ($monAge < 40) {

echo 'Tu es jeune ! Tu as '.$monAge.' ans !';

// écrit l’info à la suite le point est nécessaire

echo '<p>Tu es jeune ! Tu as $monAge ans !</p>';

// écrit un paragraphe

// guillemets nécessaires autour des balises <p> mais pas de . entre les éléments

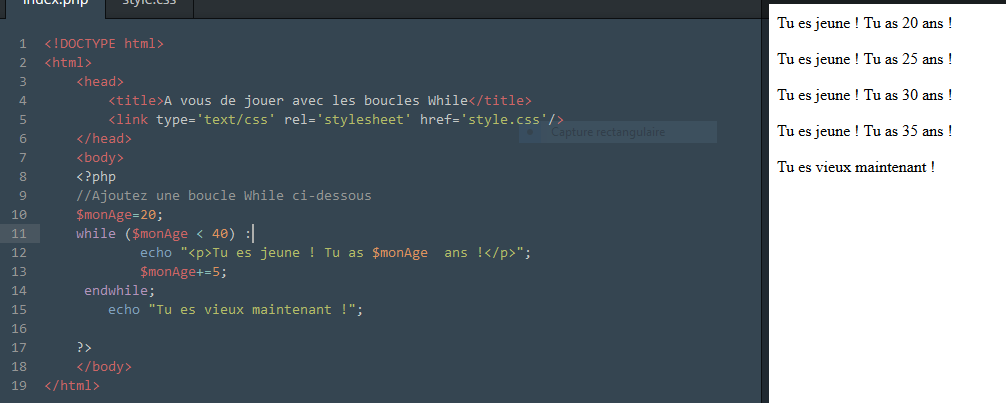
$monAge+=5;

}

echo "Tu es vieux maintenant !";

?>

***Utilisation de "Endwhile" :***



<?php

$monAge=20;

while ($monAge < 40) :

echo "<p>Tu es jeune ! Tu as $monAge ans !</p>";

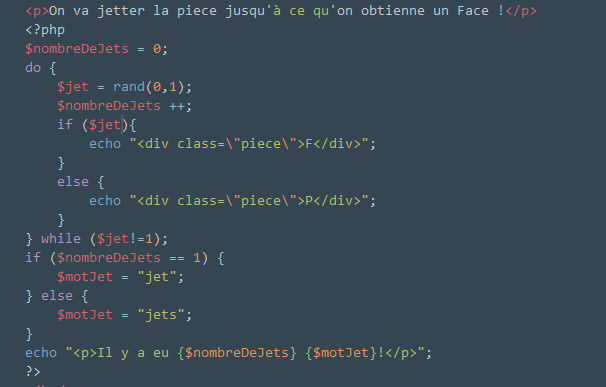
$monAge+=5;

endwhile;

echo "Tu es vieux maintenant !";

?>

Do-While : une boucle do/while est utilisée pour être sûr que le code à l'intérieur s'exécute au moins une fois.



<?php

// Ecrivez votre boucle do-while ici

$monAge=50;

do {

echo "<p> Tu es jeune ! Tu as $monAge ans ! </p>";

$monAge+=5;

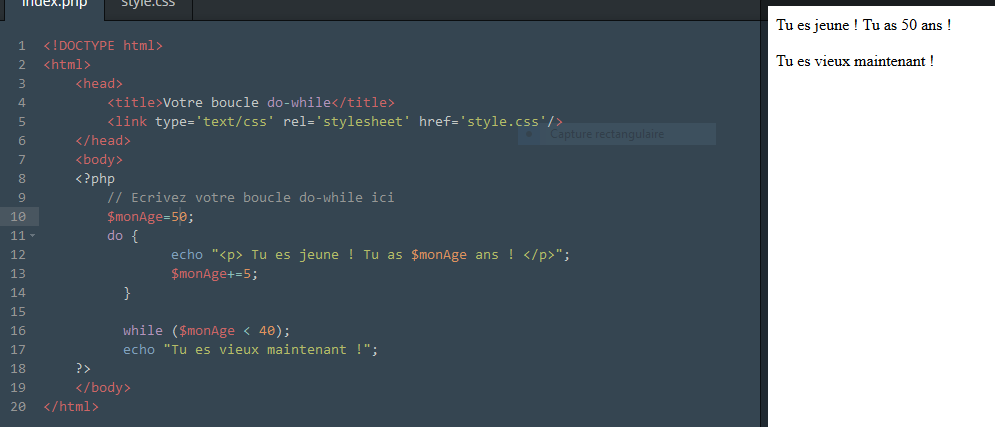
}

while ($monAge < 40);

echo "Tu es vieux maintenant !";

?>

# LES fonctions en php

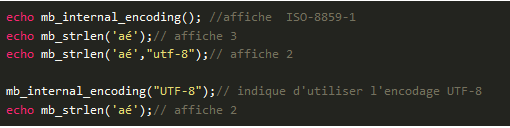
Manipuler les chaînes de caractères

**strlen(*« paramètre »*)** Calcule le nombre de caractères d’une chaîne

ATTENTION : Configurer la page en utf8 correctement pour que les accents ne soient pas comptés comme un carratère.

mb\_internal\_encoding("UTF-8"); // indique d'utiliser l'encodage UTF-8

echo mb\_strlen('aé'); // affiche bien 2

* Configuration php.ini  
  Le module "mbstring" s'occupe de la manipulation des chaînes tandis que le module "iconv" permet de convertir différents types d'encodages.  
  Les conversions entre ISO et UTF-8 peuvent se réaliser à l'aide des fonctions utf8\_encode() et utf8\_decode().  
  Dans la plupart des cas vous n'aurez pas à intervenir sur la configuration du php.ini.  
    
  Utilisation des fonctions de traitement des chaînes multi octets  
  Pour reprendre l'exemple initial, mb\_strlen() sera utilisée à la place de strlen().   
  Liste des fonctions adaptées au [traitement des chaines UTF-8](http://fr2.php.net/manual/fr/ref.mbstring.php)  
    
  PHP travaille pour l'instant en ISO, il faudra donc spécifier l'encodage utilisé pour les fonctions qui le demande ou modifier l'encodage initial.  
    
  Typiquement :

<?php

$prenom=strlen("Veronique"); // Trouve la longueur de votre nom

echo $prenom; // et affiche la longueur à l'écran !

?>

**substr(*3 paramètres*)** Extrait un morceau d’une chaîne

*Les 3 paramètres : (Chaine, Position du 1er caractère , Nb de caractère à partir du point de départ)*

**strtoupper( )** transforme la chaîne de caractère en paramètre en MAJUSCULES

**strtolower()**  transforme la chaîne de caractère en paramètre en minuscule

<?php

// Extrait une partie de phrase

// et affichez-le a l'écran :

$maPhrase = "Je s'appelle Groot !";

$verbe = substr($maPhrase, 5 , 7);

echo $verbe

?>

<?php

// Transformer votre prénom en majuscule et affichez-le a l'écran :

$monPrenom = strtoupper("Groot");

echo $monPrenom;

?>

</p>

<p>

<?php

// Transforme votre prénom en minuscule et affichez-le a l'écran :

$monPrenom = strtolower("Groot");

echo $monPrenom;

?>

</p>

</html>

**strpos()** recherche un extrait dans une chaîne de caractères.

La fonction retourne comme résultat la position du premier caractère de l'extrait trouvé.

Si l'extrait n'est pas trouvé, la fonction retourne false (faux).

$pos=strpos("emily", "e");

// Ici, $pos vaut 0 (le "e" a été trouvé en première lettre (donc position 0)

$pos=strpos("emily", "i");

// Ici, $pos vaut 2 (le "i" a été trouvé en 3ieme lettre, donc position 2

$pos=strpos("emily", "ily");

// Ici, $pos vaut 2 (l'extrait "ily" a été trouvé, son premier caractère est en position 2)

$pos=strpos("emily" "zxc");

// Ici, $Pos vaut false (l'extrait "zxc" n'a pas été trouvé)

<?php

// Affichez la position d'une lettre de votre prénom

$monPrenom = strpos("Groot","o");

echo $monPrenom;

?>

<?php

// Faite la vérification de la présence d'un caractère qui n'est pas

// dans votre nom et affichez un message d'alerte

if (strpos("Groot", "i") === false) {

echo "Désolé, pas de 'i' dans 'Groot'";

}

?>

# LES fonctions mathematiques en php

round(). Cette fonction arrondit les nombres décimaux ou un nombre décimal à une certaine précision (au centième près, par exemple).

il faut passer le nombre décimal à arrondir en premier paramètre. Puis, le second paramètre indique le nombre de décimal souhaité pour l'arrondie.

Si on n'indique pas ce second paramètre, PHP considère que l'on ne souhaite aucune décimale (nombre de décimal = 0 , donc on souhaite un nombre entier).

NB : Le language PHP connait la valeur du nombre Pi. il suffit d'utiliser le nom "M\_PI" pour que PHP le remplace par le nombre Pi avec 14 décimales (le plus souvent)

// Arrondir le nombre Pi à partir de 3,1416...

$resultat = round(M\_PI);

echo $resultat; // Affiche 3

// Pour obtenir un arrondie de Pi avec 4 décimales (4 chiffres après la virgule),

$resultat = round (M\_PI, 4);

echo $resultat; // Affiche 3,1416

<?php

$result = round(2.526564); // Arrondit un nombre décimal en un entier

echo $result; // affiche le résultat à l'écran

?>

<?php

$result = round(2.526564,3); // Arrondit un nombre décimal à la 3ieme décimale

echo $result; // affiche le résultat à l'écran

?>

**rand()** Permet d'obtenir un nombre aléatoire (au hasard).

Par défaut, rand() fournit un nombre aléatoire compris entre 0 et 32767. On peut fournir 2 paramètres (min et max) à cette fonction de façon à ce que le nombre aléatoire obtenu soit compris entre ces deux nombres

echo rand (); // Affiche un nombre compris entre 0 et 32767

echo rand (1,10); // Affiche un nombre entre 1 et 10

<html>

<p>

<?php

echo rand(); // Afficher un nombre aléatoire a l'écran

?>

</p>

<p>

<?php

$nom = ("Veronique"); // Définir la variable nom

$longueur = strlen ($nom); // Renvoie la longueur de la variable nom

$random = rand(0, $longueur-1); // Renvoie un nb aléatoire entre 0 et la longueur calculée -1 (à cause du départ à 0)

$result = substr($nom, $random, 2); // Renvoie 2 caractères aléatoirement

echo "<p> $result </p>";

$result = substr("Veronique", rand(0, strlen ($nom)-1), 2); // Idem sur une seule ligne

echo $result;

?>

</p>

</html>

# LES fonctions des tableaux

array() Fonction Tableau

array\_push() Permet d'ajouter un nouvel élément à la fin du tableau.

array\_push(prend deux arguments) un tableau et un élément à ajouter à la fin de ce tableau

count() Renvoie le nombre d'éléments du tableau.

**Exemples :**

$couleursPreferes = array ();

array\_push ($couleursPreferes, "rouge");

array\_push ($couleursPreferes, "vert");

array\_push ($couleursPreferes, "orange");

array\_push ($couleursPreferes, "saumon clair");

array\_push ($couleursPreferes, "émeraude");

echo count($couleursPreferes); // Affiche 5

**Trier les tableaux :**

sort() : Cette fonction a besoin du tableau en paramètre et fournit en résultat : true (en cas de succès du tri) ou false (en cas d'échec).

NB : Tri par ordre croissante s’il s’agit d’une suite numérique ou par ordre alphabétique s’il s’agit d’une suite de lettre ou d’une chaîne de caractères (prise en compte de la première lettre)

join() : Permet de créer une chaîne de caractères contenant chaque élément du tableau en les séparant d'un caractère qui est fournit en paramètre (le premier).

Le second paramètre est le tableau.

Le résultat est une chaîne de caractère.

$monTableau = array (5, 3, 7, 1);

sort($monTableau);

print join(",", $monTableau); // Affiche "1,3,5,7"

PHP a aussi la fonction de tri inverse: rsort ().

$monTableau = array (5, 3, 7, 1);

rsort($monTableau);

print join(":", $monTableau); // Affiche "7:5:3:1"

<?php

// Crée un tableau avec les membres de votre famille ou amis

$membreGroupe = array ("Romain", "Khaled", "Thomas", "Vero");

array\_push ($membreGroupe, "IdFixe");

// Trie le tableau pour avoir les noms en ordre alphabétiques

sort($membreGroupe);

$longueurTab = count($membreGroupe);

echo '<p>'. $longueurTab .'</p>';

echo join(",",$membreGroupe);

// Aléatoirement, sélectionne un gagnant !

$random = rand (0, count($membreGroupe)-1);

echo '<p>'.$random.'</p>';

echo '<p>'.$membreGroupe [$random].'</p>';

// Affichez le nom du gagnant en majuscule !

$gagnant = $membreGroupe [$random];

echo '<p>'.strtoupper ($gagnant).'</p>';

?>

|  |
| --- |
| **Vu sur le net :**  var\_dump ($tableau) Renvoie toutes les infos du tableau  **Le code :**  $membreGroupe = array ("Romain", "Khaled", "Thomas", "Vero");  var\_dump($membreGroupe);  **Renvoie :**  array(5) { [0]= string(6) "IdFixe" [1]= string(6) "Khaled" [2]= string(6) "Romain" [3]= string(6) "Thomas" [4]= string(4) "Vero" } |

# syntaxe des fonctions en php

La structure typique d'une fonction est la suivante :

<?php

function afficheNom () {

echo "Véronique"; // Affiche le nom

return // Retourne le résultat de la fonction

}

afficheNom (); // Appelle la fonction

?>

1. La mot-clé *function* indique que le code qui suit sera une fonction définie par le programmeur.
2. *afficheNom* indique le nom de la fonction. Le nom d'une fonction peut contenir des chiffres, des lettres, de soulignement ou des tirets. Au moment de son utilisation, peut importera que le nom comporte des majuscules ou des minuscules. On dit que le nom des fonctions est insensible à la case.
3. Les paramètres ou arguments, permettrons de passer des informations à la fonction pour qu'elle puisse effectuer sa tâche.
4. Enfin, les instructions seront les lignes de code que la fonction exécutera au moment où elle sera utilisée.

<?php

function JeMePresente ($unNom, $unAge) {

echo "Bonjour ! Mon nom est ".$unNom.", et j'ai ".$unAge." ans.";

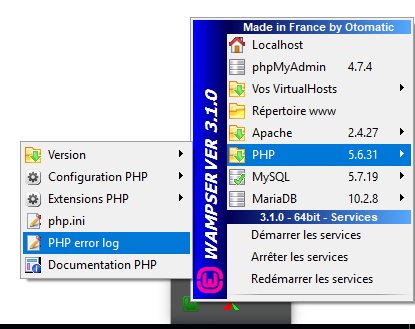
}

jeMePresente ("Véronique",50);

?>

**Raccourcis du code PHP dans l’HTML :** <?= $maVariable ?>

|  |
| --- |
| **PHP error log** |

****

Ouvre un fichier dans Sublime qui permet d’avoir un détail sur les erreurs de code :

