

TP PHP

Notions fondamentales de PHP : Variables, les types, les constantes, les tableaux, les instructions de contrôle, les fonctions, inclusion des fichiers, les formulaires, GET et POST

Créé par : Neila Ben Lakhal

TP PHP

Notions fondamentales de PHP : Variables, les types, les constantes, les tableaux, les instructions de contrôle, les fonctions, inclusion des fichiers, les formulaires, GET et POST

Pourquoi un serveur local ?

Faute de disposer d'un serveur local sur votre ordinateur personnel, vous seriez obligé pour tester vos pages PHP de les transférer sur le serveur distant de votre hébergeur puis d'appeler ces pages en vous connectant au site à l'aide de votre navigateur. La moindre erreur de code ou la moindre modification vous obligerait à répéter toute cette procédure, d'où une importante perte de temps.

Il est donc indispensable d'installer sur votre poste de travail un serveur local simulant votre serveur distant et vous permettant d'effectuer en direct tous les tests désirés. Vous aurez alors dans votre navigateur exactement le même aspect pour toutes ces pages que les visiteurs de votre site quand vous aurez opéré le transfert de vos fichiers sur le serveur distant qui l'hébergera.

Le serveur local comprend les éléments suivants, disponibles séparément aux adresses entre parenthèses :

- Serveur Apache (<http://www.apache.org>).
- Interpréteur de code PHP (<http://www.php.net>)
- Base de données MySQL (<http://www.mysql.com>).
- Utilitaire phpMyAdmin, qui permet de créer et de gérer bases et tables de données
- MySQL(<http://www.phpmyadmin.net>).
- Utilitaire SQLiteManager, qui permet de créer et de gérer bases et tables de données
- SQLite (<http://www.sqlitemanager.org>).

On peut trouver sur le Web divers packages complets pour Windows, Linux ou Mac, qui permettent d'installer en une seule opération tous ces éléments, évitant du même coup les problèmes de configuration.

Wampserver, téléchargeable à l'adresse <http://www.wampserver.com> , est destiné aux ordinateurs sous Windows et justement l'un de ces environnements.

Une fois la procédure de téléchargement terminée, il vous suffit de lancer l'exécutable, qui installe automatiquement Apache, PHP, MySQL, SQLite phpMyAdmin et SQLitemanager sur votre ordinateur. Si, pendant la phase d'installation, vous avez choisi d'installer PHP en tant que service Windows, le serveur est lancé automatiquement à chaque démarrage du système d'exploitation.

WampServer 3.1.0 inclus :

Apache 2.4.27 – PHP 5.6.31, 7.0.23, 7.1.9 – MySQL 5.7.19 – MariaDB 10.2.8 – PhpMyAdmin 4.7.4 – Adminer 4.3.1 – PhpSysInfo 3.2.7

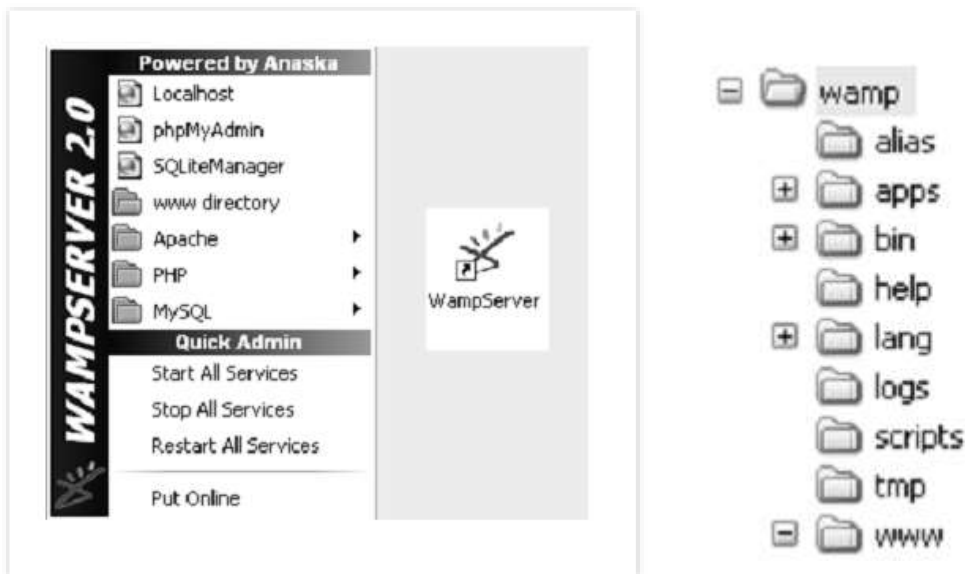


FIGURE 1 LES OPTIONS D'ADMINISTRATION, L'ICONE DE WAMPSEVER , ET ARBORESCENCE DU DOSSIER D'INSTALLATION DE WAMPSEVER

Si vous avez réalisé l'installation dans le dossier C:\wamp, vous obtenez l'arborescence illustrée à la figure 1.

Pour pouvoir être exécutés par le serveur local, tous les scripts que vous écrivez doivent être enregistrés dans le sous-dossier www. Dans ce dernier, vous pouvez créer un ou plusieurs sous-dossiers correspondant à chaque site que vous voulez tester.

Au prochain lancement du serveur (actualisation du page du serveur aussi), ils apparaîtront dans la page d'accueil de Wampserver dans la rubrique « vos projets » La page du serveur local est aussi accessible a travers l'adresse <http://localhost> (aussi <http://127.0.0.1>).

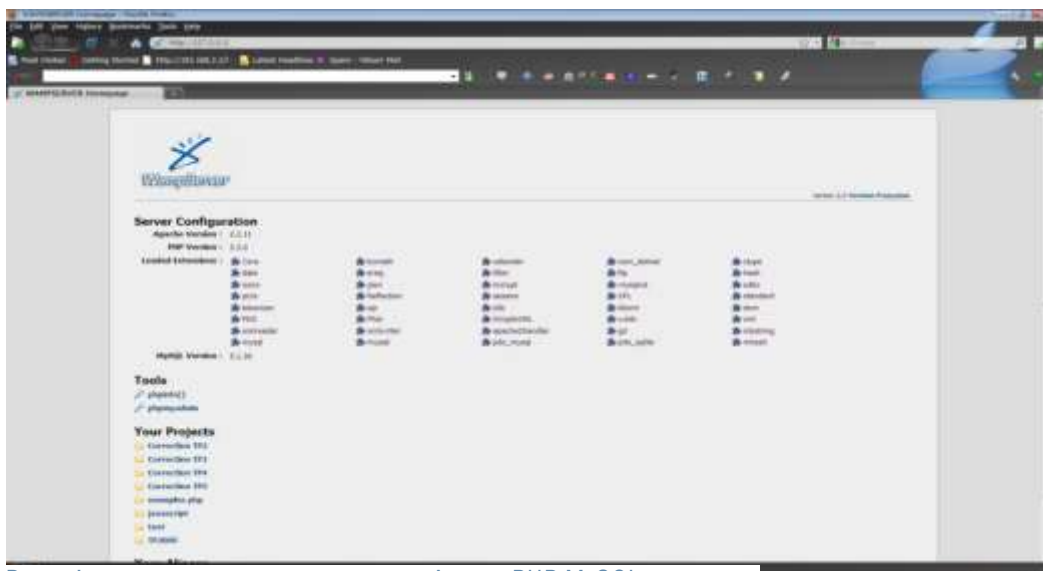


FIGURE 2 PAGE D'ADMINISTRATION DU SERVEUR LOCAL APACHE PHP MySQL

Votre première page php :

Exercice 1. Écrire le script PHP suivant :

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

Exercice 2. Enregistrer le script sous le nom info.php dans un dossier TP du dossier WWW de WAMP

Sous PHP, tous les scripts commencent par la ligne `<?php` et se terminent par `?>`. Noter que, sauf recommandation spéciale de votre hébergeur, tous les fichiers qui contiennent des instructions PHP sont enregistrés avec l'extension `.php`. Les extensions `.php3`, `.php4`, `.php5` ou `.phtml` se rencontrent sur certains serveurs, suivant la configuration effectuée par l'administrateur.

Pour l'exécution de votre script `info.php`, taper l'adresse : <http://localhost/TP/info.php>

Affichage avec echo et print :

Exercice 3. Enregistrer le code html suivant sous le nom hello.php dans un dossier TP du dossier WWW de WAMP

```
<!DOCTYPE html>
<html>
<head>
<title>Titre de la page</title>
</head>
<body><h2>Bienvenue sur le site PHP 5 </h2>
</body></html>
```

Exercice 4. Ajouter au script les balises HTML nécessaires pour que le message « Hello PHP » soit affiché en tant que titre de niveau h3 suivie d'une ligne horizontale. (Astuce : utiliser la balise `<h3>` et la balise `<hr/>`)

Exercice 5. Soit le code suivant, ajouter-le à la suite dans votre page hello.php de 4 façons possibles :

```
<h2>Bienvenue sur mon site PHP </h2>
```

- En tant que code HTML.
- En tant que script PHP en utilisant `echo`
- En tant que script PHP en définissant une variable et en l'affichant avec `echo`.
- En tant que script PHP en définissant une variable et en l'affichant autrement (pas le `echo` !!).

Affectation par valeur affectation par référence

Exercice 6. Déclarer 2 variables : nom et prenom. Les initialiser avec votre nom et votre prénom et les afficher sur la page en utilisant 3 modes syntaxiques différents :

- 2 commandes echo
- 1 commande echo avec 1 seule chaîne de caractère
- 1 commande echo avec le point de concaténation

Exercice 7. Quel est le résultat d'exécution du script suivant :

```
<?php
$mavar1="Paris";
$mavar2="Lyon";
.....
$mavar1="Nantes";
//Affectation par valeur de $mavar1 et $mavar2
$mavar1="Paris";
echo "\$mavar1= ", $mavar1, "<br />";
$mavar2="Lyon";
echo "\$mavar2= ", $mavar2, "<br />";
echo "\$mavar1= ", $mavar1, "<br />";
echo "\$mavar2= ", $mavar2, "<br />";
echo "modification de \$mavar1 <br />";
$mavar1="Nantes";
echo "\$mavar1= ", $mavar1, "<br />";
echo "\$mavar2= ", $mavar2, "<br />";
echo "modification de \$mavar2 <br />";
$mavar2="Marseille";
echo "\$mavar1= ", $mavar1, "<br />";
echo "\$mavar2= ", $mavar2, "<br />";??>
```

Exercice 8. Créer une balise de titre H1 : « Calcul sur les variables ». Affecter respectivement les valeurs 150, 50.750 et 10 aux variables prix_table, prix_armoire et Nombre. Calculer le prix HT total pour les 10 armoires et 10 tables. Comparer le prix de l'armoire et de la table et afficher quel est le prix le plus élevé. On affichera également le type de chaque variable.

Exercice 9. Affecter une valeur à une variable nombre et affichez la somme des entiers de 1 à nombre. On réalisera cet exercice avec l'instruction FOR puis avec l'instruction WHILE.

Exercice 10. Écrire le script d'une fonction qui réalise le produit de N nombres qui lui sont passés en tant qu'éléments du tableau \$tab, affiche le nombre de paramètres et retourne leur produit.

Exercice 11. Réécrire le script de la fonction de l'exercice 12 en ajoutant les balises HTML adéquates pour que l'affichage des résultats de la multiplication d'un tableau contenant les nombres 7, 12, 15, 3 et 21 soit affiché ainsi :

7	12	15	3	21	Le produit vaut : 79380
---	----	----	---	----	-------------------------

Exercice 14. Initialiser un tableau de 4 cases (contenant des nombres) et en faire la somme.

- sans faire de fonction
- en créant une fonction somme
- en créant un fichier spécifique qui contient la fonction somme.

Exemples de valeurs pour obtenir les résultats ci-dessous : 3, 2, 10, 5.

Exercice 15. Écrire un tableau multidimensionnel associatif dont les clés sont des noms de personne et les valeurs des tableaux indicés contenant le prénom, la ville de résidence et l'âge de la personne.

Clé	valeur		
Mohamed	Ben Ali	Tunis	31
Nizar	Ben Amor	Sfax	34
Ahmed	Ben Salah	Tunis	28

Afficher la structure du tableau en utilisant La fonction `print_r()`.

Exercice 16. Écrire un tableau multidimensionnel associatif dont les clés sont les prénoms de personne et les valeurs des tableaux associatifs dont les clés sont le nom, la ville de résidence et l'âge de la personne avec une série de valeurs associées

Clé	valeur			
	clé	nom	ville	age
Mohamed	Valeur	Ben Ali	Tunis	31
Nizar	clé	nom	ville	age
	Valeur	Ben mohamed	Sfax	34
Ahmed	clé	nom	ville	age
	valeur	Ben Salah	Tunis	28

Exercice 17. Utiliser une boucle `foreach` pour lire les tableaux des exercices 16 et 17.

Exercice 18. Construire une page qui permet de saisir un nom et un mot de passe. Renvoyer l'utilisateur sur une autre page et lui afficher si son mot de passe est correct ou non (NB : le mot de passe valide sera «mot»). Sur cette 2ème page : prévoir un bouton retour à la première page qui contient le formulaire.

Refaire l'exercice en utilisant la méthode GET à la place de POST afin de récupérer vos informations :

- Remarquer le changement de l'URL avec la méthode GET.
- Quelles méthodes choisir pour passer des informations confidentielles comme mot de passe par exemple ?

Exercice 19. Créer un formulaire comprenant un groupe de champs ayant pour titre "Adresse client". Le groupe doit permettre la saisie du nom, du prénom, de l'adresse, de la ville et du code postal. Les données sont ensuite traitées par un fichier PHP `ex19.php` séparé récupérant les données et les affichant dans un tableau HTML. Le fichier `ex19.php` contient seulement du code dans des blocs de script php (cad pas de balise HTML hors du script php).

Le formulaire est le fichier `ex19.html` qui ne contient que du code HTML.

Exercice 20. Créer un formulaire demandant la saisie d'un prix HT et d'un taux de TVA. Le script **calcule le prix TTC et** affiche le montant de la TVA et le prix **TTC calculé** dans deux zones de texte créées dynamiquement. Le formulaire maintient les données saisies.