

Traitement des données d'un formulaire
Méthodes POST ET GET


Mise en situation : vous avez un site internet et vous souhaitez enregistrer des commandes d'un client pour produit que vous proposez (par exemple vous louez des appartements, vous vendez des voitures, des articles que vous avez fabriqués ou encore enregistrer les membres d'un club de ping-pong ...)

Problématique : comment récupérer les données issues d'un formulaire afin de pouvoir les traiter ?

I- La méthode POST

Initiation au langage PHP

Objectif : Récupérer les données d'un formulaire
lé cool oté !



Nous sommes le 28-03-2019 et il est 05:32

NOM

PRENOM

AGE

SEXE

VILLE

PAYS

60 visiteurs

On va créer deux fichiers :

index1.php
Traitement.php

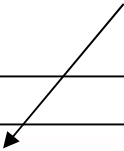
Ces deux fichiers texte contiendront du code html et du code php.

Le code html est compréhensible directement par votre navigateur mais par contre le code php ne l'est pas. Pour que cela fonctionne il vous faut donc :

- Soit un serveur web internet avec php
- Soit un serveur web php en local

Pour nos tests, nous utilisons **easyPhp devserver** qui est gratuit, explication installation à la page suivante.

Lorsqu'on appuiera sur le bouton « envoyer », le programme « traitement.php » va récupérer les données du formulaire et afficher les informations suivantes



Merci pour votre enregistrement

Bonjour Jean Claude Hoareau vous êtes m , vous habitez à St Denis Réunion et vous avez 30 ans
Vous avez plus de 18 ans !
Vous pouvez passer commande

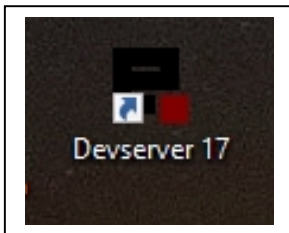
Mais tout d'abord, commençons par l'installation du serveur PHP.

Allez sur ce lien :

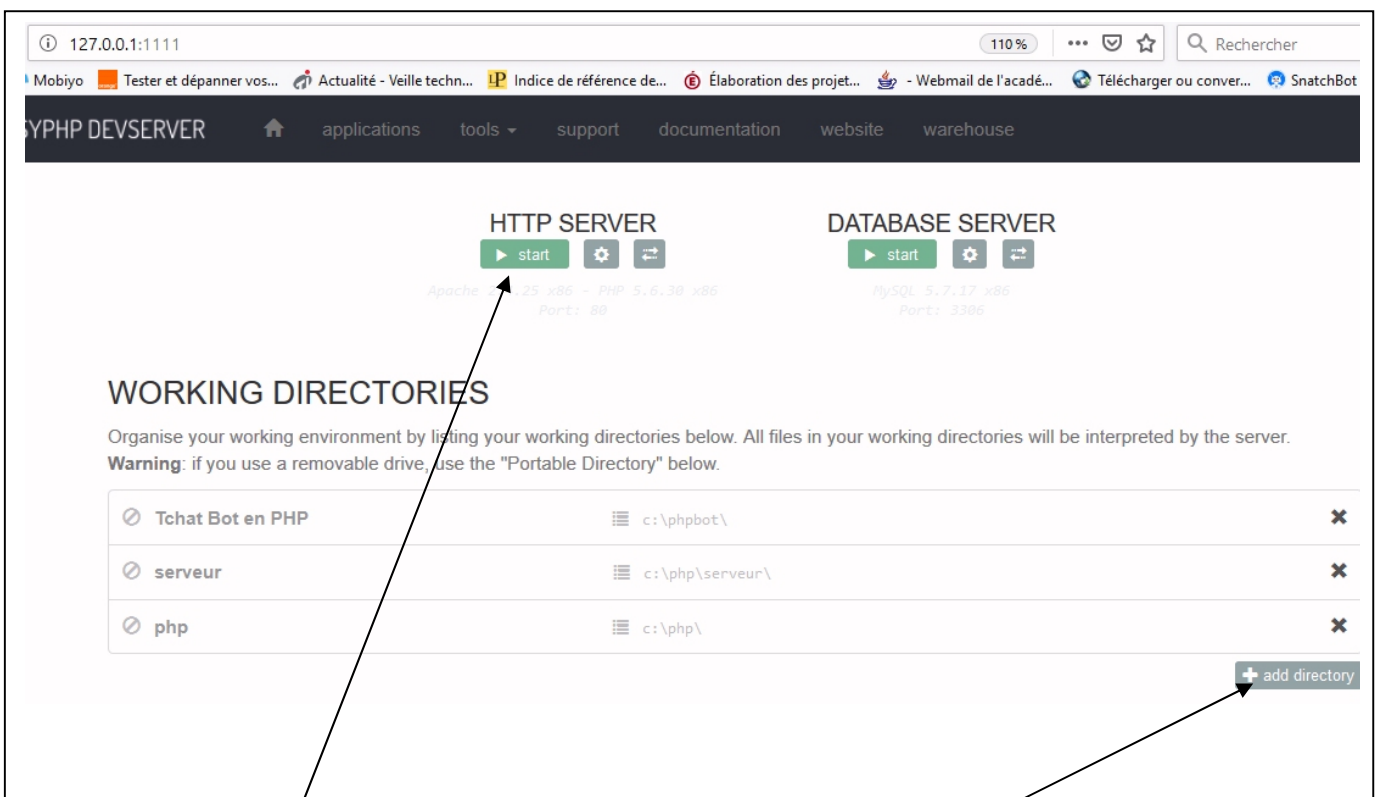
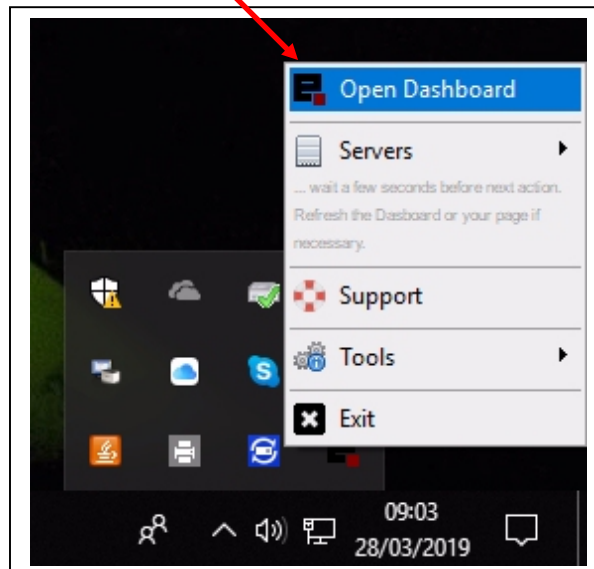
<https://www.easyphp.org/easyphp-devserver.php>

Télécharger et lancer l'installation. Une fois installé, lancer UNE FOIS le programme Devserver : ATTENTION : le serveur se lance en tâche de fond et donc vous ne verrez rien se lancer à priori, c'est tout à fait normal. Ne cliquez pas 10 fois sur le logo, vous le lancerez 10 fois sans le savoir ...

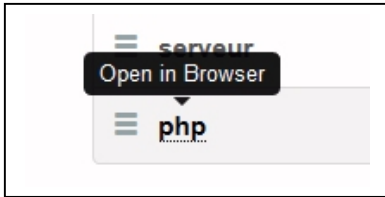
Une fois lancé, rechercher en bas à droite de votre écran le logo et cliquer avec le bouton droit dessus pour avoir le menu contextuel suivant :



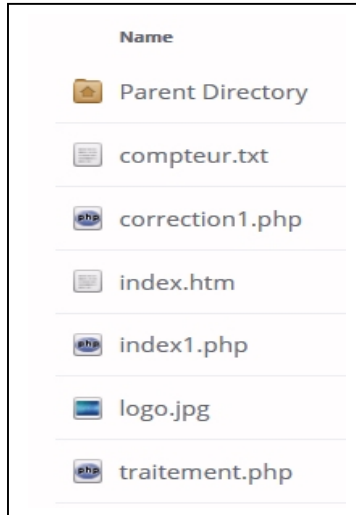
Cliquer sur OPEN Dashboard pour accéder à la page d'accueil de lancement du serveur (accès via votre navigateur web sur l'IP local host 127.0.0.1 :1111)



Lancer le serveur http et créer vos répertoires de travail (où vont se trouver vos fichiers php) Personnellement, je mets mes fichiers dans c:\php\ pour me faciliter la vie ensuite...



Dans le dossier php que vous avez créé, vous devrez mettre vos fichiers texte, extension .php et autres fichiers annexes si nécessaire (images, .htm ...)



Pour éditer mes pages web, programmation python, php, javascript ... j'utilise un seul et même éditeur : GEANY <https://www.geany.org/Download/Releases>
Il reconnaît les extensions de fichiers et permet un affichage du code adapté au niveau des couleurs. A l'usage c'est très pratique.

Le premier fichier que vous devez créer est index1.php, on utilisera la méthode « POST » :

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta http-equiv="Content-Language" content="fr">
5 <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
6 <title>Initiation programmation PHP </title>
7 </head>
8 <body>
9 <p align="left"><font face="Arial" size="4"> Initiation au langage PHP</font></p>
10 <font face="Arial">
11 <?php
12 echo "Objectif : Récupérer les données d'un formulaire. Cours B.Hoareau LLP";
13 echo '<div align="left"><font face="arial" size="2" color="blue"> lé cool oté ! !</font><br /> ';
14 echo '</div> ';
15 $date = date("d-m-Y");
16 $heure = date("H:i");
17 Print("Nous sommes le $date et il est $heure");
18 >?
19
20 <form method="POST" action="traitement.php">
21 <p>
22 <font face="Arial">NOM<br>
23 <input type="text" name="nom" size="32"><br>
24 <br>
25 PRENOM<br>
26 <input type="text" name="prenom" size="32"><br>
27 <br>
28 AGE<br>
29 <input type="text" name="age" size="20"><br>
30 <br>
31 SEXE<br>
32 <input type="text" name="sexe" size="20"><br>
33 <br>
34 VILLE<br>
35 <input type="text" name="ville" size="20"><br>
36 <br>
37 PAYS<br>
38 <input type="text" name="pays" size="20"></font></p>
39
40 <p><input type="submit" value="Envoyer" name="B1"><input type="reset" value="Rétablir" name="B2"></p>
41 </form>
42 </font>
43 <?php
44 $fp = fopen("compteur.txt","r+"); // 1.0n ouvre le fichier en lecture/ecriture
45 $nbvisites = fgets($fp,11); // 2.0n récupère le nombre dans le fichier
46 $nbvisites++; // 3.0n incremente le nombre de visites(+1)
47 fseek($fp,0); // 4.0n se place en debut de fichier
48 fputs($fp,$nbvisites); // 5.0n écrit dans le fichier le nouveau nb
49 fclose($fp); // 6.0n ferme le fichier
50 print("$nbvisites visiteurs"); // 7.0n affiche le compteur a l'ecran
51 >?
52 </body>
53 </html>
                    
```

Le code PHP s'écrit à l'intérieur d'une page HTML standard

Balise de début du code php
Balise de fin de code php

Nom du fichier php qui sera appelé une fois qu'on appuie sur le bouton « envoyer » du formulaire

Pas d'accents ni espace. Ces « name » sont transmis au fichier traitement.php

Encore un peu de code php pour ouvrir un fichier et sauvegarder un compteur de visite

Créez ensuite le fichier traitement.php :

```

1 <html>
2 <head>
3   <meta http-equiv="Content-Language" content="fr">
4   <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
5   <title>Traitement du formulaire</title>
6 </head>
7
8 <body>
9
10  <p> Merci pour votre enregistrement</p>
11  <?php
12
13     $prenom = $_POST['prenom'];
14     $nom = $_POST['nom'];
15     $age = $_POST['age'];
16     $sexe = $_POST['sexe'];
17     $ville = $_POST['ville'];
18     $pays = $_POST['pays'];
19
20     print("Bonjour $prenom $nom vous êtes $sexe , vous habitez à $ville $pays et vous avez $age ans")
21
22     if ($age < 18) {
23         print ("<BR>Vous êtes trop jeune pour passer une commande sur ce site !");
24     }
25
26     else {
27         print ("<BR>Vous avez plus de 18 ans ! ");
28         print("<BR>Vous pouvez passer commande");
29     }
30
31  <?>
32
33 </body>
34 </html>

```

La méthode \$_POST permet de récupérer les informations « name » envoyées par le fichier index1.php. Il faut remettre exactement les même « name » utilisés dans le formulaire.

Affiche le contenu de la variable \$nom

Vous pouvez traiter les données reçues pour réaliser ce que vous voulez. Ici, si -18ans, on interdit de commander en ligne par exemple.

Toujours dans votre dossier, créer aussi un fichier texte « compteur.txt » vide utilisé par php pour enregistrer les visites sur le formulaire et n'oubliez pas le fichier image logo.jpg

Exécuter votre page avec EASY PHP DEVSERVER (via le working directory)

Si à l'exécution l'affichage des accents se fait mal, vérifiez que vous avez bien sélectionné dans votre éditeur geany : Document : Encodage : Unicode utf-8

Faire valider par le professeur

II- La méthode GET.

Vous allez maintenant utiliser la méthode GET pour transmettre le même formulaire. et vous devrez ensuite comparer les deux méthodes :

Pour cela, faites une copie de votre fichier index1.php en index2.php par exemple et traitement.php en traitement2.php

Dans le fichier index2.php, remplacer « POST » par « GET » et indiquer pour l'action « traitement2.php »

De même dans le fichier « traitement2.php », remplacer « POST » par « GET »

Lancer la page index2.php (via easyphp devserver – working directory)

Vérifier que les données sont correctement traitées.

Que remarquez vous au niveau de la barre URL ?

Que pouvez vous en conclure par rapport à la transmission par méthode POST ?

Dans quelles conditions utiliseriez vous telle ou telle méthode ?

III- Questionnaire

- Donner le code source qui permet de créer un bouton et une zone de texte :

- Identifier la nature de ce code (html , css, python, php ...) :

- Dans le code du bouton, donner le mot clef qui permet de lancer le programme php associé au formulaire :

- Donner la ligne qui permet d'indiquer la méthode et l'action associée au formulaire :

- A quoi sert la balise : `<?php` `/ ?>`

- Distinguer ce qui est exécuté coté client (par l'explorateur) ou coté serveur :

Le code html :

L'appui sur un bouton :

Le traitement de l'action (envoi des données) :

Le programme php de traitement des données (traitement.php) :

Le renvoi des informations traitées (sous forme html) :

L'affichage des informations :

Quel protocole est utilisé pour transmettre les informations du formulaire ?

Connaissez vous d'autres protocoles pour transmettre des données ? (fichiers, mail ...)

Proposer un graphique pour illustrer les flux entre client et serveur ainsi que ce qui est traité par l'un ou l'autre dans le cas notre exemple (index1.php et traitement.php) :

Note : lorsque le client accède à la page index1.php, le serveur exécute le code php de index1.php et envoie au client le résultat de l'exécution ainsi que le code html brut de la page