

# TP : PHP

## I. Se connecter au serveur SFTP et lire le code de test.php

Ouvrir FileZilla et Firefox (WEBeleve) en même temps.

Télécharger (ou plutôt éditer) dans Notepad le fichier `test.php`.

➤ Comprendre que :

- les variables commencent par « \$ »
- les « ; » donne la fin d'instruction
- « minute » est noté « i » car m signifie month

➤ Afficher le code source dans Firefox et voir qu'il n'y a pas de PHP

En effet, dès que le serveur reçoit une requête GET pour `test.php` :

Le serveur commence par compiler le PHP, l'instruction `echo` génère du code HTML

Le résultat de cette compilation est donc du code HTML.

Le serveur envoie le fichier HTML au client.

Il est impératif que l'extension du fichier soit `*.php`

Contrairement au javascript, le client ne verra jamais les instructions PHP (lorsqu'on effectue un quizz, normalement on le code en PHP, et surtout pas en JS)

## II. Boucle for en PHP et if

➤ Modifier `test.php` pour ajouter un code qui écrit 100 paragraphes :

```
<p>ligne 1</p>
```

```
...
```

```
<p>ligne 100</p>
```

Pour créer une boucle en PHP :

```
for ($i = 0 ; $i < 10 ; $i++) {  
    // insérer ici les instructions (terminées par ";" )  
}
```

➤ Modifier à nouveau pour ajouter un trait toutes les 10 lignes.

Pour vous aider voici un code qui donne la parité d'un entier

```
if ($nb % 2 == 0) {echo "$nb est pair";}   
else {echo "$nb est impair";} }
```

## III. Une table de multiplication

Lire la documentation de la fonction `random_int` sur la documentation officielle :

<https://www.php.net/manual/fr/function.random-int.php>

En PHP, l'opérateur de concaténation est le point :

`"17"." ans"` donne `"17 ans"`

➤ Ecrire un code PHP qui affiche la table de multiplication d'un entier choisi au hasard.

Bien entendu, on n'hésitera pas à utiliser une balise titre `<h1>` , un bloc paragraphe `<p>` et les retours à la ligne `<br>`.

Bienvenue dans le serveur W

PHP fonctionne.

Nous sommes le 15-02-2020

### Table de 11

```
0 x 11 = 0  
1 x 11 = 11  
2 x 11 = 22  
3 x 11 = 33  
4 x 11 = 44  
5 x 11 = 55  
6 x 11 = 66  
7 x 11 = 77  
8 x 11 = 88  
9 x 11 = 99  
10 x 11 = 110
```

## IV. Double boucle for en PHP (facultatif)

- Modifier test.php pour créer un carré 10x10 avec un symbole en utilisant une variable \$symbole="" (La difficulté est qu'il faut enlever les paragraphes <p> dans le echo)

## V. Une page de réponse avec GET

Créer un nouveau fichier analyse.php et y copier-coller la structure du HTML5 et ce code PHP :

```
<?php
$nom = $_GET["prenom"];
$age = $_GET["age"];
echo "Bonjour, $nom, et vous avez $age ans.";
?>
```

Dans Firefox, taper en URL, cette requête GET :

[https://ftp.nsirenn.es.fr/<votre\\_identifiant>/analyse.php?nom=toto&age=17](https://ftp.nsirenn.es.fr/<votre_identifiant>/analyse.php?nom=toto&age=17)

- Mince, une petite erreur s'est glissée ! Corrigez-là !

En fait, \$\_GET est un dictionnaire prédéfini de PHP, qui contient toutes les variables passées par l'URL, sous la forme : ma\_page.php?cle1=valeur1&cle2=valeur2&cle3=valeur3

## VI. Un formulaire en méthode GET

- Créer un nouveau fichier « formulaire.html » avec un formulaire 'minimaliste' demandant nom et âge (pas de PHP ici, donc \*.html)  
Pas besoin d'identifiant dans les balises.

```
nom : <input type="text"><br>
age : <input type="text">
```

- Puis modifier cette base que ce formulaire soit soumis au serveur et que l'action résultante soit une page web réponse « analyse.php ».

```
<form action="analyse.php" method="GET">
    nom : <input type="text" name="prenom"> <br>
    age : <input type="text" name="age"> <br>
    <input type="submit" value="soumettre">
</form>
```

- Exécuter et remarquer que l'URL contient les variables (qu'on peut modifier dans l'URL d'ailleurs).

## VII. Un formulaire en méthode POST

- Modifier les 2 fichiers précédents :
  - Dans le fichier formulaire** : Changer la méthode de transmission en method="POST"
  - Dans le fichier d'analyse** : Changer la méthode de récupération des variables en tableau \$\_POST[]
- Observer l'URL  
Vous ne devriez voir aucune variable cette fois ... c'est plus confidentiel !