

# Les formulaires et le PHP

---

## Création du script PHP

Savoir créer un formulaire, c'est bien, mais ce serait encore mieux si les informations saisies pouvaient être traitées.

Pour cela, il va falloir écrire un fichier **traitement.php** auquel fait appel la balise *form*.

Il faudra donc modifier l'attribut *action* de la balise *form*.

Ce qui suit vous donne donc quelques connaissances élémentaires du langage PHP qui nous suffiront pour cette séance.

Pour que le code PHP puisse être interprété par votre machine, il faut y installer PHP. Nous allons utiliser pour cela le logiciel XAMPP.

## Les bases du PHP

Le PHP est un langage pensé spécialement pour créer des pages Internet *dynamiques*, c'est-à-dire dont l'affichage peut évoluer en fonction des actions de l'utilisateur. Le HTML pur ne permet pas ce comportement et ne permet de créer que des pages Internet *statiques*.

Le code PHP peut être intégré au sein d'un fichier HTML (on peut mélanger HTML et PHP!) dont on aurait modifié l'extension en .php ou être placé dans un fichier indépendant ne comportant que du code PHP mais ayant l'extension .php. Le code PHP doit être inséré dans une balise spéciale :

```
<?php ... code PHP ... ?>
```

## Les variables

Quand vous cliquez sur le bouton d'envoi du formulaire, vos saisies sont enregistrées dans des variables qu'il faut savoir appeler pour que le script PHP puisse les traiter. Le nom d'une variable est toujours précédé du symbole \$. Ensuite, la syntaxe du PHP ressemble étrangement à celle du Java.

Par exemple :

```
<?php  
    $nombre1=15;  
?>
```

On affecte la valeur 15 à la variable nombre1. Vous remarquez le point virgule pour terminer l'instruction !

Les variables peuvent prendre une valeur particulière, NULL, qui signifie que la variable n'a pas de valeur ! Cela peut être intéressant, en particulier pour les formulaires, pour tester si l'utilisateur a complété toutes les questions. Exemple d'algorithme :

Si \$Nom=NULL alors écrire « Veuillez entrer votre nom ! ».

À propos, les variables issues d'un formulaire ont une syntaxe particulière.

Si le formulaire créé utilise la méthode POST. Si on écrit l'instruction suivante :

```
$_POST['Nom'] = 'Dupont';
```

On affecte la valeur Dupont à la variable Nom qui a été saisie dans un formulaire utilisant la méthode POST.

Les notations dans le fichier HTML qui définit votre formulaire et celles présentes dans votre script PHP doivent concorder.

### Exemple :

Supposons que, dans le formulaire, se trouve le code suivant :

```
<label for="prenom">Votre prenom : </label>
<input type="text" name="prenom" id="prenom">
```

L'utilisateur tape Romain pour répondre. Lorsque le bouton d'envoi est utilisé, une nouvelle variable est créée

`$_POST['prenom']` ; et elle a pour valeur Romain.

### L'instruction echo

L'instruction echo permet d'écrire à l'écran. Par exemple :

```
echo 'Bonjour';           écrit le mot Bonjour à l'écran.
echo $_POST['prenom'];    écrit le contenu de la variable $_POST['prenom'] à l'écran.
echo 'J\'aime la spécialité NSI';  L'antislash \ permet d'afficher des apostrophes dans
                                    une chaîne de caractères.
```

Pour combiner l'affichage de chaînes de caractères et l'affichage de valeurs de variables, il faut utiliser des points comme séparateurs. Par exemple :

```
echo 'Vous avez entré ' . $_POST['prenom'] . ' comme prénom.';
```

Cette instruction affichera : « Vous avez entré Romain comme prénom. »

Mais on aurait aussi les stocker dans des variables :

```
$prenom=$_POST['prenom'];
```

Et ensuite :

```
echo 'Vous avez entré ' . $prenom . ' comme prénom.';
```

## La structure if ... else

Sa syntaxe est très similaire à celle utilisée en Java. Jugez-en plutôt :

```
if (condition) // si la condition est respectée
    { série d'instructions.}
else // si la condition n'est pas respectée
    {série d'instructions.}
```

Exemple :

```
<?php
if ($moyenne >= '10')
    {echo 'Vous avez la moyenne ce
trimestre.';} else
    {echo 'Votre moyenne est insuffisante, au travail !';}
?>
```

Symboles utilisés pour les comparaisons :

==	Est égal à
>	Est supérieur à
<	Est inférieur à
>=	Est supérieur ou égal à
<=	Est inférieur ou égal à
!=	Est différent de

Exemple de code :

```
<?php

$prenom=$_POST['prenom'];

$nom=$_POST['nom'];

echo 'Vous avez entre ' . $prenom . ' comme prenom.';

echo '<br>';

echo 'Votre nom est ' . $nom ;

echo '<br>';

?>
```

## Manipulation 1

Créez le fichier **traitement.php**, et recopiez le programme précédent.

Vous voyez que le code récupère les valeurs nom et prénom.... et afficher leurs valeurs.

Pour le tester, Lancer le logiciel **XAMPP Control Panel**, puis cliquer sur « START » pour les deux premiers modules.

Cliquez sur la droite sur « Explorer », puis ouvrir le dossier « htdocs ».  
Dans « htdocs », créez un dossier à votre « nom », puis mettez y vos fichiers.

Lancer « Firefox » et entrez l'adresse suivante : *localhost / « votre nom » / « nom du fichier ».html*

*Vous devriez voir votre page html et pouvoir la tester.*

## Manipulation 2

---

Rajouter plusieurs instructions conditionnelles pour compléter votre page.

Exemple afficher :

Nom,  
Prénom,  
Age,  
Age avec un calcul à partir de l'année de naissance,  
si vous êtes un homme ou une femme,  
votre sport préféré,  
vos goûts.....

Faire ensuite un petit questionnaire qui compte les points et corrige les questions fausses ...