

# TP1 de PHP - un premier survol

**Objectif :** comprendre l'utilité de PHP. Comprendre le fonctionnement client - serveur.

## Introduction

Rappel du vocabulaire de base

- **Une variable** correspond à un emplacement en mémoire auquel on donne un nom. On peut y stocker une valeur : `a=10`. L'emplacement en mémoire nommé `a` contient désormais la valeur 10.
- **Affectation** : signe `=`. Cela consiste à affecter une valeur (ou une nouvelle valeur) à une variable. Ex : `a=10`.
- **Assigner une valeur à une variable** : affecter une valeur (à droite du égal) à une variable (à gauche du égal) : `a=10` permet d'assigner la valeur 10 à la variable `a` (qui sera donc considérée comme un entier). `a=b` permet d'assigner la valeur que contient la variable `b` à la variable `a`.

1. Créez dans votre espace o2switch un fichier `test.php` qui contient le script suivant :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html ; charset=iso-8859-1">
    <title>Page de test</title>
  </head>
  <body>
    <h1>Page de test </h1>
    <?php
      echo "Hello World! \n<br /> Bonjour à tous!" ;
    ?>
  </body>
</html>
```

Veillez à bien indenter vos programmes de sorte qu'ils soient lisibles ! Quand vous ferez des projets plus conséquents cela sera essentiel...

2. Visualisez le résultat de votre script : tapez son url dans un navigateur (firefox de préférence), du type `http://prenom.nom.mmi-velizy.fr/test.php`.
3. Affichez le code source de la page dans votre navigateur. Que constatez-vous ? Quelle différence faites vous entre la balise `<br />` et la séquence `\n` ? Pour le savoir, vous pouvez retirer l'une, puis l'autre de ces séquences, recharger la page à chaque modification (et réafficher le code source) et voir ce qui a changé...

## Formulaires, fonctions, tableaux, relations avec la BDD... Un premier survol

On vous donne ici un programme utilisant des formulaires, des fonctions permettant d'insérer/sélectionner dans une base de données, etc... Le but est que vous commenciez à comprendre ces notions (que l'on reverra plus progressivement ensuite), et que vous compreniez l'utilité de PHP.

`$_POST['nom']` permet d'accéder à la valeur du champ du formulaire qui a `nom` comme attribut name. Cela n'existe que si quelque chose a bien été envoyé par formulaire.

Les premières lignes de code correspondent à des déclarations de fonction (qui font des opérations en lien avec une base de données). Ces morceaux de code ne seront exécutés qu'au moment où ils sont appelés, par le nom de la fonction.

1. Récupérez le fichier `premier_programme.php` sur e-campus, et mettez-le dans votre répertoire web (`public_html`), et testez son fonctionnement.
2. Pour l'instant, un formulaire s'affiche suivie d'une erreur. Au vu de ce qui est expliqué plus haut, pouvez-vous dire pourquoi ?
3. Quelle ligne du code produit en premier quelque chose à l'écran ?
4. Le formulaire n'a pour l'instant pas de bouton permettant de soumettre les données... Rajoutez-le (c'est un input de type submit), et dites à l'aide de l'attribut `action` de la balise `form` que vous voulez que ce soit ce même fichier `premier_programme.php` qui traite le formulaire : c'est à lui que sont envoyés les données du formulaire. Ce même programme sert alors à deux moments différents : avant envoi des données (le formulaire

n'a pas été rempli) et après soumission des données... A l'aide de l'instruction `if` on pourra s'adapter à l'une ou l'autre des situations.

5. Si vous soumettez le formulaire, l'erreur s'affiche-t-elle encore ? Pourquoi ?
6. Complétez l'affichage produit à l'écran pour que ne s'affiche pas seulement le nom entré dans le formulaire, mais un message du type "Bonjour <prenom> <nom>", où <prenom> et <nom> sont les données du formulaire... (et tout ceci dans un paragraphe, balise `p`).
7. changez `method='post'` par `method='get'`. Que faut-il changer pour que le code marche de nouveau ? Observez que les variables sont maintenant passées via l'URL...
8. Déplacez le code de traitement du formulaire (l'affichage du message de bienvenue) dans un autre fichier, et changez le champ action pour que le traitement du formulaire se fasse bien dans ce nouveau fichier... Cela règle le problème de l'erreur qui s'affichait initialement.
9. Appelez la fonction `insérer_etudiant()` pour que les données du formulaire soit mise en base de donnée (on ajoute un nouvel étudiant à la base de données), puis appelez la fonction `affiche_etudiant()` pour afficher tous les étudiants présents dans la base de données. Que se passe-t-il si vous inversez l'ordre dans lequel sont appelées ces fonctions ?
10. Si vous êtes à l'aise : faites en sorte à ce qu'on puisse entrer son âge également (une colonne age est déjà présente dans la base de données `olecadet_base_etudiants`). Faites une fonction permettant de retourner l'étudiant le plus jeune.