

# Technologies du Web - TP 3 : PHP

PHP est un langage exécuté côté serveur. Usuellement, le serveur est une machine distante connectée au réseau; dans ce TP c'est votre machine qui sera à la fois le client et le serveur. La partie serveur sera assurée par Apache. Le client c'est votre navigateur, comme d'habitude.

À l'UM2, Apache est préconfiguré pour servir des pages se trouvant dans votre répertoire personnel. Le répertoire racine se nomme "public\_html"; s'il n'existe pas vous devrez le créer : ouvrez votre répertoire personnel, cliquez sur "fichier" puis "créer un dossier"; nommez-le "public\_html". Les pages PHP que vous créerez dans ce TP devront être placées dans ce dossier, ou dans un de ses sous-dossiers.

Le protocole utilisé pour communiquer avec un serveur Web est HTTP. Pour parler à votre propre machine en tant que serveur Web, dégainez votre navigateur et rendez-vous à l'adresse suivante, qui est dépendante de votre login : <http://localhost/~votrelogin>  
Si vous avez créé dans "public\_html" une page nommée "biniou.php", vous y accéderez donc par <http://localhost/~votrelogin/biniou.php> et ainsi de suite.

## Exercice 1 - premiers pas

Créez votre première page PHP, nommée par exemple "essai.php". Pour illustrer l'aspect dynamique de PHP, nous allons afficher la date et l'heure.

Dans votre fichier PHP, programmez un squelette de page HTML minimal et ajoutez dans le <body> une portion de code PHP qui affiche la date et l'heure. Pour obtenir la date et l'heure, on utilisera la fonction date(), par exemple : date("Y/m/d h:i:s")

Testez votre page : <http://localhost/~votrelogin/essai.php>

Que se passe-t-il lorsque vous rechargez la page ?

Étudiez rapidement la documentation de la fonction date pour comprendre son fonctionnement : <http://fr2.php.net/manual/fr/function.date.php>

Ajoutez une nouvelle portion de code pour afficher l'adresse IP du client, comme vu en TD.

## Exercice 2 - message personnalisé

Programmez l'exercice 2 du TD.

La liste des messages sera contenue dans un tableau. Au début de votre page PHP, déclarez en utilisant array() à bon escient (<http://php.net/manual/fr/function.array.php>), un tableau de sept

éléments dans lequel chaque élément représentera un message correspondant au jour de la semaine.

Vérifiez que cela fonctionne avec le jour en cours, puis écrivez une boucle qui permette d'afficher tous les messages à la suite.

## Exercice 3 - paramètre d'URL

Enlevez ou commentez la boucle, puis modifiez le code conformément à l'exercice 3 du TD.

Pour lire les paramètres, quelle variable *superglobale* doit-on utiliser ? Renseignez-vous sur leurs différences : <http://php.net/manual/fr/reserved.variables.php>

## Pour les geeks curieux

**Curl** est un client HTTP en ligne de commande. Ce petit logiciel s'utilise depuis un terminal Linux et permet d'étudier les interactions entre client et serveur lorsque vous accédez à une page; il aide aussi à déboguer les problèmes de serveur.

**Exemple d'utilisation:** `curl -X GET "http://www.google.fr"`

Essayez d'accéder à votre page avec la commande **curl** avec et l'option `-vI` (c'est un `i` majuscule). Observez tout ce que contiennent les entêtes HTTP.

## Annexe 1 - remarques sur PHP et Apache

- pour que le code PHP soit exécuté par le serveur, il est indispensable d'y accéder par le protocole HTTP. Si vous ouvrez votre fichier PHP avec le navigateur, comme vous ouvririez un fichier contenant seulement du HTML, le code PHP ne sera pas exécuté
- un fichier PHP doit porter l'extension (càd avoir un nom se terminant par) `".php"` sinon le code PHP ne sera pas exécuté

## Annexe 2 - bonnes pratiques

- indentez votre code, comme toujours !
- donnez des noms explicites aux variables (pas de `$d` mais plutôt `$dateDuJour`)
- en cas de doute ou pour toute information sur les fonctions disponibles, consultez la documentation de PHP, agrémentée de nombreux exemples : <http://php.net/manual/fr/>