

TP n° 4

PHP



1 Présentation du PHP

PHP a été créé en 1994 par Rasmus Lerdorf pour les besoins des pages web personnelles (livre d'or, compteurs, etc.). A l'époque, PHP signifiait *Personal Home Page*. PHP est un langage de script, c'est-à-dire que le code est enregistré sous forme de fichier texte sur le disque dur, et qu'il est exécuté à la demande par un programme chargé de l'interpréter. C'est habituellement l'internaute (par l'intermédiaire de son navigateur Web) qui demande l'exécution d'un script lorsqu'il consulte une page Web. La demande est reçue par le serveur Web (par exemple Apache HTTPD), qui se rend compte qu'il doit la sous-traiter à PHP.

En 1997, PHP devient un projet collectif et son interpréteur est réécrit par Zeev Suraski et Andi Gutmans pour donner la version 3 qui s'appelle désormais « *PHP Hypertext Preprocessor* ».

Sa gratuité et le libre accès à ses sources en fait un langage très populaire surtout auprès de la communauté GNU Linux. Sa syntaxe proche du C, et sa programmation « intuitive » en font le langage qui a fait la plus grande percée auprès des webdesigners en 2001. De nombreux sites lui sont consacrés, on y trouve des scripts, des astuces et même des concours de logos...

2 Documentation

Le site officiel de PHP est extrêmement bien fait. La documentation de n'importe quelle fonction est accessible depuis l'URL :

```
http://php.net/nom_de_la_fonction
```

N'oubliez pas de consulter les commentaires utilisateurs, car de très nombreux commentaires donnent des conseils très utiles.

La documentation complète de PHP est disponible en français à l'adresse :

```
http://www.php.net/manual/fr/
```

3 Mon premier fichier en PHP

Voici le fichier [hello.php](#). Il utilise la fonction `htmlspecialchars` (qui convertit les caractères éligibles en entités HTML) pour afficher un message de bienvenue.

```
<?php echo '<?xml version="1.0"?>'; ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <title>PHP HelloWorld</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  </head>
<body>
<?php
  if(empty($_GET['user'])){
echo htmlspecialchars('Bonjour', ENT_QUOTES, 'UTF-8');
  }else{
echo htmlspecialchars('Bonjour, '.$_GET['user'], ENT_QUOTES, 'UTF-8');
  }
?>
</body>
</html>
```

TP n° 4 PHP

Enregistrez ce fichier sur votre serveur web en `hello.php` et ouvrez-le depuis votre navigateur.

```
http://xxx.xxx.xxx.xxx/hello.php
```

Testez ensuite avec un paramètre :

```
http://xxx.xxx.xxx.xxx/hello.php?user=Gaetan
```

4 Exercices PHP

4.1 Exercice 1 : Compréhension

Soit le code PHP suivant :

```
<?php
    $i=0;
    print( $i++ + "3 voitures" . " camions<br/>" );
    print("$i" . "3 voitures" . " camions <br/>" );
    print( $i + "4.1 voitures" + " camions" . 3/2 . "<br/>" );
?>
```

Indiquez le texte affiché lors de l'exécution de ce code sans écrire le fichier. A quoi correspondent les opérateurs '.' et '+'.

4.2 Exercice 2 : Quelques exemples simples

Ecrivez un fichier PHP qui affiche les éléments suivants :

- La date du jour (fonction `date`)
- La date de dernière modification (fonctions `date` et `filemtime`)
- Le contenu d'un fichier texte (fonction `include`)
- L'adresse IP du client (fonction `getenv`)

5 PHP et mysql

Lire et écrire une base de données est relativement simple en PHP, puisqu'il existe justement des fonctions élaborées dans ce but. Regardons rapidement comment utiliser tout cela.

Je vous rappelle que vous pouvez vous connecter à votre base mysql avec la commande suivante :

```
> mysql -u nom_utilisateur -p nom_de_la_base
```

Connectez-vous avec les droits root sous mysql.

Créez une nouvelle base nommée `tp3db` puis créez un nouvel utilisateur ayant les droits sur cette base.

```
> mysql -u root -p mysql
mysql> CREATE DATABASE tp3db;
mysql> USE mysql;
mysql> GRANT ALL PRIVILEGES ON tp3db.* TO 'phpuser'@'localhost' IDENTIFIED BY 'passphp' WITH GRANT OPTION;
mysql> GRANT ALL PRIVILEGES ON tp3db.* TO 'phpuser' IDENTIFIED BY 'passphp' WITH GRANT OPTION;
mysql> COMMIT;
mysql> USE tp3db;
```

TP n° 4 PHP

Commencez par créer une table de test avec la commande suivante sous mysql.

```
CREATE TABLE personne (
  id int(11) NOT NULL auto_increment,
  nom varchar(20) NOT NULL,
  email varchar(50),
  dateheure datetime,
  PRIMARY KEY (id)
);
```

Pensez à ajouter quelques données dans votre table comme le montre l'exemple ci-dessous:

```
INSERT INTO personne(nom,email,dateheure) VALUES('gaetan','gaetan.rey@unice.fr',NOW())
```

5.1 Connexion avec la base de données

Avant d'envoyer des requêtes à la base de données, il faut déjà s'y connecter.

Pour cela on utilise la fonction `mysql_connect()` :

```
mysql_connect(serveur, utilisateur, mot_de_passe) ;
```

La déconnexion (avant la fin du script) se fait avec la fonction `mysql_close()` :

```
mysql_close() ;
```

5.1.1 Insertion de données dans la table

Une fois la base créée, vous pouvez y ajouter des données à l'aide de PHP. Comme vu précédemment, connectez-vous d'abord avec mysql, puis sélectionnez la base sur laquelle vous voulez travailler. Envoyez ensuite des requêtes (query) à la base.

```
<?php
switch($_POST['action']){
  // INSERTION DANS LA BASE
  case "inserer":
    // déclaration de quelques variables
    $host = "localhost";
    $user = "phpuser";
    $pass = "passphp";
    $bdd = "tp3db";
    $table = "personne";
    // connexion avec MySQL
    @mysql_connect($host, $user, $pass) or die("Impossible de se connecter a la base de
données");
    // Le @ indique a php de ne pas afficher de message d'erreur
    @mysql_select_db($bdd) or die("Impossible de se connecter a la base de données");
    // affichage sélection
    echo '<p>Bonjour '.htmlentities($_POST['nom']).', votre email est ' .
htmlentities($_POST['email']).'</p>';
    // stockage dans la bdd
    $nom = $_POST['nom'];
    $email = $_POST['email'];
    if(!get_magic_quotes_gpc()){
    // si php n'est pas configuré pour le faire automatiquement, on ajoute des \ devant les '
    // deux raisons : autoriser d'ajouter des ', et sécuriser notre requête
    // on y reviendra dans un prochain tutorial
      $nom = addslashes($nom);
      $email = addslashes($email);
    }
}
```

TP n° 4

PHP

```

    $sql = "INSERT INTO " . $table . "(nom, email, dateheure) VALUES('" . $nom . "','" .
$email . "', NOW())";
    $res = mysql_query($sql);
    // déconnexion avec MySQL
    mysql_close();
    break;
    // AFFICHAGE FORMULAIRE HTML
    default:
?>
<form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>"
<input type="hidden" name="action" value="inserer" />
<p>Votre nom : <input type="text" name="nom" /></p>
<p>Votre email : <input type="text" name="email" /></p>
<p><input type="submit" name="Submit" value="Ok" /></p>
</form>
<?
    break;
}
?>

```

5.1.2 Affichage du résultat d'une requête

Vous avez maintenant des données dans la base, il ne vous reste plus qu'à les afficher. Le principe est le même que précédemment : se connecter, puis envoyer des requêtes.

```

<?php
    $sql = "SELECT * FROM " . $table;
    $res = mysql_query($sql);
    // Tant qu'une ligne existe, place cette ligne dans la variable $val
    // sous la forme d'un tableau associatif.
    while ($val = mysql_fetch_assoc($res)) {
        $nom = $val['nom'];
        $email = $val['email'];
        $dateheure = $val['dateheure'];
        echo $dateheure . ' : ' . $nom . '(' . $email . ')<br />';
    }
?>

```

5.2 Exercices PHP et mysql

En vous inspirant de l'exemple précédent, créer une table dans la base tp3db. Cette table « objet » aura un 'id', un 'type', un 'nom' et un 'propriétaire'. Le propriétaire sera en fait l'id d'une des personnes de la table « personne »

- Concevez une page web qui permet d'ajouter un nouvel objet.
- Concevez une page web permettant de connaître tous les objets d'une personne.
- Concevez une page web permettant de connaître toutes les personnes qui possède un objet d'un type donné.

6 Les sessions en PHP

Les sessions sont adaptées à la sauvegarde de données confidentielles ou importantes. On peut citer quelques exemples courants de leur mise en application :

- authentifier un visiteur ;
- garder des informations sur un utilisateur tout au long de sa présence dans votre application ;

TP n° 4 PHP

- gérer le panier d'achat d'un internaute sur votre site marchand ;
- mettre en place des formulaires en plusieurs parties et donc retenir les informations fournies dans les pages précédentes ;
- effectuer un cache par utilisateur de certaines actions coûteuses en ressources.

6.1 Qu'est-ce qu'une session ?

Au lieu de stocker vos informations chez le visiteur, vous les stockez sur le serveur. Techniquement, vous attribuez au visiteur un identifiant. À chaque fois qu'il revient en annonçant cet identifiant, PHP récupérera toutes les informations qu'il avait sauvegardé et qui sont relatives à ce visiteur.

6.2 Lecture et écriture

L'utilisation des sessions est très simple pour le programmeur : la manipulation est presque transparente et il suffit de lire ou d'écrire dans un tableau associatif classique une fois l'initialisation de la session faite.

La session s'initialise avec `session_start()`. PHP essaie alors de lire l'identifiant fourni par l'utilisateur, va chercher le fichier correspondant, et vous met à disposition les informations sauvegardées dans la variable `$_SESSION[]`.

```
<?php
// initialisation
session_start() ;
// tester la présence de la variable 'langage' dans la session
if ( isset( $_SESSION['langage'] ) ) {
    echo 'langage existe dans la session et sa valeur est ' ;
    // lecture de la variable de session 'langage'
    echo $_SESSION['langage'] ;
} else {
    echo 'langage n'existe pas dans la session' ;
}
?>
```

Pour écrire dans une variable de session il suffit d'initialiser la session via `session_start()` et d'écrire dans la variable `$_SESSION[]`.

```
<?php
// initialisation
session_start() ;
$_SESSION['langage'] = 'PHP 5' ;
?>
```

Après cette instanciation il sera possible sur toutes les pages suivantes d'afficher la valeur de la variable de session `$_SESSION['langage']`.

6.3 Suppression d'une session

PHP efface la session de lui-même au bout d'un certain temps.

Si vous voulez forcer la destruction de la session vous pouvez utiliser la fonction `session_destroy()`.

```
<?php
// on initialise et utilise la session
session_start();
$_SESSION['nom'] = 'Pierre' ;
echo $_SESSION['nom']; // affiche Pierre
// divers traitements
```

TP n° 4 PHP

```
// on détruit la session
session_destroy();
unset($_SESSION);
echo $_SESSION['nom'] ; // n'affiche rien
?>
```

Notez que cette fonction ne fait qu'effacer le fichier de données sur le serveur, elle n'efface pas les variables présentes dans `$_SESSION[]` ni ne supprime le cookie. Si vous voulez éviter tout risque de confusion, effacez `$_SESSION[]` aussi.

6.4 Exercices SESSIONS

Créez une page d'accueil `index.php` qui :

- Affiche votre login, un lien de déconnexion et un message de bienvenue si une session PHP contenant votre login existe.
- Affiche un lien vers une page de connexion et un message vous invitant à vous connecter si aucune session PHP est actuellement ouverte.

Créez, ensuite une page `connexion.php` qui va gérer les connexions et déconnexions.

7 Les Cookies en PHP

Un cookie est un petit fichier texte placé dans l'ordinateur de l'internaute à la demande du serveur lors de la visite d'une page. Ce fichier permet de stocker un petit nombre d'informations, comme un mot de passe, le nombre de visites, etc. Il est conseillé d'utiliser les cookies avec parcimonie, car un internaute peut les supprimer ou configurer son navigateur pour les refuser.

AVERTISSEMENT : Les cookies peuvent être considérés par certaines personnes comme une violation de leur vie privée. La CNIL (Commission Nationale de l'Informatique et des Libertés, France) recommande aux webmasters qui ont recours aux cookies de le signaler très clairement sur leur page d'accueil.

Ainsi, il est souvent préférable d'utiliser les sessions à la place des cookies.

Un cookie peut être envoyé depuis un script PHP avec la fonction `setCookie()`. Cette fonction doit être utilisée avant tout envoi d'en-tête HTTP au navigateur (il ne faut rien afficher avant d'utiliser cette fonction).

7.1 Création de cookie

Cette fonction prend pour paramètres :

- **\$name** : le nom du cookie (cette chaîne de caractères ne doit contenir ni virgule, ni espace, ni les deux points « : »).
- **\$value** : la valeur du cookie.
- **\$expires** : la date d'expiration du cookie au format timestamp Unix (c'est-à-dire le nombre de secondes écoulées depuis le 1er janvier 1970), s'elle n'est pas spécifiée, le cookie sera supprimé lors de la fermeture du navigateur.
- **\$path** : le répertoire du site web pour lequel le cookie est valide (les sous-répertoires sont acceptés) : par défaut le répertoire actuel
- **\$domain** : le nom de domaine pour lequel le cookie est valide
- **\$secure** : indique (0 ou 1) si le cookie doit être transmis via une connexion sécurisée par le protocole HTTPS

TP n° 4 PHP

Tous les paramètres sont optionnels sauf \$name.

```
<?php
    setcookie("TP3", "1", time()+24*3600, "/", "unice.fr", 0);
?>
```

Cet exemple envoie le cookie nommé *TP3* de valeur *1*, de date d'expiration : le jour suivant (aujourd'hui + 24 heures). Il est valide dans tout le site web (c'est-à-dire depuis la racine / du site) pour le domaine unice.fr. Il ne requiert pas de connexion sécurisée par HTTPS.

7.2 Lecture de cookie

En PHP il n'est pas nécessaire d'aller chercher quoi que ce soit : PHP le fait tout seul automatiquement. Tous les cookies valides pour le domaine et le répertoire du script PHP en cours vont être mis en mémoire. Ainsi des variables portant le nom des cookies et valant leur valeur respective vont être créées. Le tableau `$_COOKIE` contient la liste des cookies disponibles.

```
<?php
    echo $_COOKIE[TP3]; // affiche "1"
?>
```

7.3 Suppression de cookie

Pour supprimer un cookie il suffit de le recréer à l'identique mais sans valeur et avec une date d'expiration dépassée.

```
<?php
    setcookie("TP3", "", time()-1000, "/", "unice.fr", 0);
?>
```

7.4 Exercices Cookies

7.4.1 Exercice 1 : gestion des connexions

Modifiez les pages écrites lors de l'exercice 1 de la partie SESSIONS et ajoutez y les fonctionnalités suivantes :

- La page de connexion crée un cookie sécurisé contenant le login et un autre le mot de passe de l'utilisateur en plus de créer la session. Cette création devra uniquement se faire lors d'une connexion valide.
- La page de déconnexion détruit le cookie.
- La page d'accueil, dans le cas où il n'y a pas de session, vérifie la présence du cookie de connexion. S'il existe, elle crée la session et s'affiche comme en cas de session ouverte. S'il n'existe pas, elle affiche le même message qu'avant quand la session n'existait pas.

7.4.2 Exercice 2 : gestion des styles

Ecrivez une petite page PHP ainsi que au moins deux mises en forme différentes à l'aide de deux fichiers CSS différents. Offrez sur cette page, la possibilité à l'utilisateur de choisir parmi les styles que vous proposez. Le choix de l'utilisateur sera sauvegardé dans un cookie. Effectuez la manipulation, puis fermez votre page. Rouvrez celle-ci. Le choix de l'utilisateur devra être respecté.

8 Références

Vous pourrez trouver des informations complémentaires à ce cours/TP dans les ouvrages et sites suivants :

TP n° 4 PHP

- PHP 5 - MySQL 5 - AJAX : 1ère édition, 12/2007 Editions ENI, ISBN13 : 978-2-7460-4057-1
- Pratique de MySQL et PHP : 3ème édition, Janvier 2005 Editions O'Reilly, ISBN13 : 9782841773381
- PHP 5 et Apache 2 : 1 édition, 2006-06-08 Editions ENI, ISBN13 : 9782746033252
- Les Cahiers du programmeur PHP 5 : 1 édition, 2004-06-01 Editions Eyrolles, ISBN13 : 9782212112344
- Premières applications Web 2.0 avec Ajax et PHP : Janvier 2008 Editions Eyrolles, ISBN13 : 9782212120905
- PHP 5 Avancé : 3eme édition, 2 octobre 2006 Editions Eyrolles, ISBN10 : 2212120044

Ce cours/TP a été créé en se servant des documents suivants :

- [Cours complet de PHP4 et MySQL](#) par Hugo ETIEVANT
- [Cours PHP](#) par Guillaume Rossolini

