

TDS de PHP - variables, fonctions, tableaux.

Rappels 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`.

Exercice 1 - premiers programmes

A réaliser sur machine.

1. Créez dans le dossier `public_html` de 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, du type `http://votrelogin.velizy-mmi.o2switch.site/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...
4. Faites volontairement une erreur, en effaçant par exemple le guillemet fermant de votre `echo`. Un message d'erreur s'affiche-t-il ? Si ce n'est pas le cas, allez dans le cpanel de votre espace, vers le bas trouvez le menu "sélectionner une version de PHP", cliquez dessus, puis choisissez l'onglet "options". Le champ "`display_errors`" doit être coché.
5. Faites maintenant un autre script, `info.php` :

```
<?php
  phpinfo();
?>
```

puis visualisez son résultat pour voir l'utilité de la fonction `phpinfo()`.

Exercice 2 - chaînes de caractères

1. Affectez à une variable `$age` une valeur entière correspondant à votre âge. Affichez la chaîne de caractère `J'ai X ans` où `X` est remplacé par la valeur de `$age`. Vous procéderez de deux façons :
 - En utilisant la concaténation des chaînes de caractères ;
 - Puis à l'aide de la seule fonction `echo`.
2. Assignez une chaîne de caractères à une variable `$chaine1` et donnez la même valeur à une autre variable `$chaine2`. Faites un script qui écrit à l'écran si les deux chaînes sont identiques ou non :
 - En comparant les deux chaînes à l'aide de l'opérateur `==`,
 - En comparant les deux chaînes à l'aide de la fonction `strcmp`. Il vous faudra auparavant voir comment fonctionne cette fonction, par exemple dans la documentation officielle <http://fr.php.net/manual/fr/>.
 - Quand est-ce qu'une chaîne est considérée comme "plus petite" qu'une autre par `strcmp` ? Vérifiez le. Vérifiez également si "bonjour" et "Bonjour" sont considérées comme identiques.

Exercice 3 - Structures de contrôle.

1. Après avoir trouvé l'erreur classique qui s'y est glissée, réécrivez le programme suivant, mais en utilisant une instruction `switch` :

```
$jour=4;
if ($jour=1){
    echo "Nous sommes lundi";
} elseif ($jour==2){
    echo "Nous sommes mardi";
} elseif ($jour==3){
    echo "Nous sommes mercredi";
} elseif ($jour==4){
    echo "Nous sommes jeudi";
}
}
```

2. Ce script ne marche pas ! Où sont les erreurs ? Que fait-il ? Comment le modifier pour qu'il passe à la ligne entre deux affichages successifs ?

```
<html>
<body>
<?php
    for ($i=1;$i>=10;$i++){
        $res = $i*13
        echo '13 * $i = $res\n';
    }
?>
</body>
</html>
```

3. Faites un script PHP qui permet d'afficher un champ de texte (vide) avec un fond d'une couleur aléatoire. La ligne HTML qui permet de faire un champ de texte (vide et inactif) de couleur rouge est :

```
<input type='text' name='couleur' disabled='disabled' style='color:#FF0000; background-color:#FF0000' />
```

Votre script PHP doit donc produire cette ligne, mais avec une variable `$couleur` à la place de `#FF0000`... Dans une couleur, les deux premiers chiffres après le dièse sont le rouge (compris entre 0 et 255 en hexadécimal, donc entre 00 et FF), les deux suivants sont le vert, et les deux derniers le bleu. Vous devez donc utiliser la fonction `rand()` pour choisir un nombre aléatoire entre 0 et 255, le convertir en hexadécimal grâce à la fonction `dechex()`, recommencer pour le vert et pour le bleu, et former avec ces trois nombres hexadécimaux aléatoires votre variable `$couleur`... Utilisez internet et surtout la documentation officielle pour voir comment utiliser `rand()` et `dechex()`...

4. Mettez dans le même répertoire que votre fichier PHP, ou dans un sous-répertoire `images`, 9 images de votre choix, que vous appellerez `image1.jpg`, `image2.jpg`, etc... (si ce sont des jpeg bien sûr). Faites une page en pur HTML qui permette de les afficher toutes les neuf, dans un tableau (3 images par lignes). Vous pouvez créer un fichier CSS pour gérer la taille des images (elles doivent toutes avoir la même taille).

Quand c'est fait, effacez le code HTML correspondant au tableau d'images et essayez de le générer de nouveau à l'aide d'un code PHP : dans un premier temps, cherchez à afficher toutes les images, à l'aide d'une seule boucle `for`, toutes alignées à la suite des autres. Quand cela marche, faites en sorte que toutes les trois images on passe à la ligne (cela peut se faire à l'aide de la balise `<br>`). Enfin, quand cela marche, faites l'affichage à l'aide de deux boucles `for` imbriquées (l'une pour les lignes, l'autre pour les colonnes), en enlevant les balises `<br>` et en utilisant un tableau (balises `<table>`, `<tr>`, et `<td>`).

5. (**facultatif, à faire si vous êtes en avance**) Écrivez un script qui affiche à l'écran tous les nombres de Fibonacci plus petits que 6765. Les deux premiers nombres de Fibonacci sont 0 et 1, et chaque numéro de la suite de Fibonacci s'obtient en faisant la somme des deux numéros antérieurs (le troisième nombre de Fibonacci est donc 0+1 soit 1 ; le quatrième, 1+1, etc...).
6. Refaites le même exercice, mais en utilisant une structure `do {...} while ( ... )` au lieu d'une structure `while(...){ ... }`. Quelle est la différence entre ces deux structures ?

Exercice 4 - fonctions

1. Emilie, qui travaille chez un FAI, doit créer un fichier où elle doit mettre, les uns à la suite des autres (un mail est séparé du suivant par deux tirets bas consécutifs) tous les mails de bienvenue à envoyer aux nouveaux clients. Or tous ces mails sont presque identiques : "Monsieur UnTel, nous avons le plaisir de confirmer que vous êtes à présent bien inscrit chez nous. Vous découvrirez bien vite l'attention que nous réservons à nos clients etc etc...". Seul le mot "Monsieur" change si le destinataire est une femme (pour le mot "Madame", donc...) ainsi

que le nom du destinataire. Or pour l'instant, Emilie fait tout manuellement : elle copie-colle un mail et change à la main le premier mot si nécessaire ainsi que le nom. Mais cela s'avère assez fastidieux... Proposez-lui un script PHP qui lui simplifie un peu la tâche en utilisant une fonction !

2. Soit la fonction `mystere` :

```
function mystere($n){
    $val=1;
    for ($i=1;$i<=$n;$i++)
        $val*=$i;
    return $val;
}
```

Simulez le fonctionnement de cette fonction lorsqu'on lui passe la valeur 4 en argument. Quel sera le résultat affiché ? Que fait cette fonction ? Pouvez-vous en écrire une version récursive ?

3. Écrivez une fonction `somme($n)` qui fait la somme des n premiers chiffres. Ainsi, `somme(3)` doit retourner 6, `somme(4)` doit retourner 10, etc... Testez votre fonction. Simulez sur papier son fonctionnement lorsqu'on lui rentre le paramètre d'entrée 4.

Exercice 5 - dates

1. Il vous faudra là consulter la documentation officielle pour voir comment fonctionne la fonction `date`. Le but est d'afficher dans une page web la date d'aujourd'hui, sous le format suivant :

```
Nous sommes le 5/09/2008;
il est 14:12:18+
```

2. Faites une fonction prenant en argument une année `$annee` et retournant `true` si l'année est bissextile, `false` sinon.

Pour continuer à la maison...

Si vous souhaitez continuer vos scripts PHP chez vous, il est fortement conseillé d'installer sur votre ordinateur personnel un serveur local simulant votre serveur distant. Cette démarche vous évitera d'avoir à transférer, pour chacun de vos tests, tous vos fichiers sur le serveur distant de votre hébergeur.

Il existe sur le web des packages complets pour Windows ou Mac, qui permettent d'installer en une seule opération tous les éléments constituant un serveur local (serveur web Apache, interpréteur de code PHP, base de données MySQL 5, phpMyAdmin...).

1. Sur windows, télécharger et installer le package WAMP5 1.6.4a disponible sur <http://www.wampserver.com> dans le dossier C:/wamp5 par exemple. Tous les scripts devront alors être enregistrés dans le sous-dossier www, dans lequel vous pourrez créer plusieurs sous-dossiers correspondant à vos différents sites. Lors du redémarrage du serveur, ils apparaîtront dans la rubrique Vos projets de la page d'accueil de Wampserver.
2. Sous Mac OS X, le logiciel Apache 1.3 est déjà installé par défaut. Pour démarrer le serveur, il suffit d'activer le service Partage Web personnel dans la rubrique Partage des Préférences Système. Le module PHP correspondant au serveur web Apache inclus dans Mac OS X est téléchargeable sur <http://www.entropych.com/software/macosx/php> Il vous suffit alors de suivre les instructions d'installation du package PHP 5.1.4 pour Apache 1.3.

Exercice 6 - tableaux

1. Jean a 20 ans, Jacques en a 30, Pierre 25. Faites un tableau `$age`, dont les indices sont les prénoms, qui permette de mémoriser ces informations. Affichez l'âge de Pierre à l'aide du tableau.
2. Faites un tableau qui associe à des noms de fruits le nombre de calories correspondant. Ensuite, faites une fonction qui parcourt votre tableau de fruits et qui en sortie affiche le nom du fruit le plus calorifique.
3. Faites une fonction qui prend en entrée une phrase et qui indique en sortie quel est le i -ème mot de cette phrase. On utilisera la fonction `explode()` dont vous regarderez le fonctionnement dans la documentation officielle.
4. On veut maintenant créer un tableau `$identite` dont chaque ligne contienne l'identité d'une personne : son âge, son prénom, et son nom. Les indices du tableau seront "Personne 1", "Personne 2", "Personne 3". On opte pour la méthode suivante :

```
$identite=array(
    "personne 1"=> array("nom" => "dutronc", "prenom"=>"michel", "age"=>"30"),
    "personne 2"=> array("nom" => "duroc", "prenom"=>"émilie", "age"=>"23"),
    "personne 3"=> array("nom" => "dupersil", "prenom"=>"évelyne", "age"=>"36"));
```

- (a) Rajoutez dans ce tableau une 4ème personne, Etienne Duvillage, 48 ans.

- (b) Faites ensuite afficher à l'écran tous les noms, et seulement les noms, des personnes présentes dans le tableau.
5. (a) Jean a fait ses courses et décide de mémoriser ses achats informatiquement. Il décide de stocker tous les produits qu'il a acheté dans un tableau, en PHP. Créez le tableau `chariot` qui contient les produits suivants : chocolat, poulet, citrons, oignons (ainsi, `$chariot[0]` vaut "chocolat").
- (b) Rajouter du curry dans le tableau `chariot`.
- (c) Jean relève le contenu d'un chariot chaque mois. Il décide donc d'utiliser un tableau bidimensionnel, de sorte, par exemple, que `$chariot["janvier"]` contienne un tableau avec les achats relevés en janvier. Créer un tel tableau dans lequel sont recensés, en janvier, des poivrons, du jambon et du sel, en février du curry, du poulet, et des yahourts, et en mars du chocolat, des citrons et des oignons.
- (d) Faites une fonction qui prend en entrée un mois et qui permette à Jean de lister ce qu'il avait acheté ce mois.
- (e) Faites une fonction qui permette de lister tous les produits achetés depuis le début de l'année.
- (f) **A la maison** : faites une fonction qui calcule la moyenne des nombres d'un tableau de nombre (ce tableau sera donc passé en paramètre de la fonction).
- (g) **A la maison** : faites une fonction qui calcule le produit de tous les nombres d'un tableau de nombre.

Exercice 7 - Formulaires

- Faites un formulaire html qui demande un nom (`type="text"`), un mot de passe (`type="password"`), et qui demande le sexe (`type="checkbox"`). Si le mot de passe entré est SRC2 ; faites un script `traitement_formulaire.php` qui traite ces données : si le mot de passe est SRC2, il affiche "bonjour M. X" ou "bonjour Mme X", selon la case cochée, où X est le nom rentré dans le champ du nom.
- Faites un formulaire de commande de voiture : l'utilisateur rentrera son nom dans une zone de texte, la marque de la voiture dans une autre zone de texte, puis devra cocher les cases (`checkbox`) indiquant les options qu'il choisit parmi :
 - Climatisation
 - Injection au méthane
 - Ventilation des rotules
 - alarmeLe script php chargé de traiter ce formulaire devra afficher la phrase, si X est le nom du client et Y la marque de la voiture qu'il souhaite acheter : "la voiture de rêve de M. X est une Y avec options :". Cette phrase sera suivie d'un tableau html à une colonne résumant les options choisies.

Exercice 8 - Fichiers

- Faites un script qui propose un formulaire permettant d'ajouter un client (nom, prénom, sexe). Ce formulaire est à chaque fois reproposé, de sorte que l'on peut ajouter autant de clients que nécessaire. Il s'agit de stocker, à chaque fois, ces clients dans un fichier `clients.txt` (une ligne par client).
- Après avoir entré de la sorte 3 ou 4 clients dans le fichier `clients.txt`, faites un autre script chargé d'afficher la liste des noms (et seulement les noms) des clients à l'écran, récupérés dans le fichier `clients.txt`. On supposera ici pour simplifier que tous les noms sont écrits en un seul mot. On pourra utiliser `explode`.

Exercice 9 - Téléchargement de fichiers - upload

- Faites un petit formulaire dont le but est de charger sur le serveur un fichier (`<input type="file"...>`). N'oubliez pas l'attribut `enctype` de la balise `form` (`enctype="multipart/form-data"`).
- Un nom temporaire est attribué sur le serveur au fichier uploadé, ce nom est stocké dans le tableau super-global `$_FILES` de la sorte : `$_FILES['fichier']['tmp_name']` dans le cas où l'input `type="file"` avait comme attribut name appelé "fichier". Il suffit d'utiliser la fonction `move_uploaded_file()` (cf documentation) pour finaliser le transfert en changeant le nom temporaire en le nom définitif (le nom du fichier est stocké dans `$_FILES['fichier']['name']`). Attention toutefois aux problèmes de droit d'écriture sur le serveur, il faudra peut-être les changer pour qu'un fichier puisse effectivement être créé sur le serveur (local dans notre cas, `public_html`). S'il y a eu une erreur lors du transfert, son numéro de code est stocké dans `$_FILES['fichier']['error']`, et la fonction `move_uploaded_file()` renverra le booléen `false`. A l'aide de ces éléments d'explication et des exemples de la documentation officielle, réalisez le transfert d'une image présente sur votre bureau vers un dossier de votre choix dans `public_html`.
- Avec des bases de données : il faut non seulement transférer l'image mais aussi stocker son nom et les autres informations utiles dans une base de données. Créez une table `image` sur votre base de données `mysql`. Celle-ci aura au moins trois champs : *titre* de l'image, un `tag` la décrivant, la `date` à laquelle elle a été créée. Modifiez votre formulaire pour que les données de l'image (nom, tag, date) soient stockées dans la base de données au moment où vous la chargez sur le serveur.

4. Avec HTML5, on peut uploader plusieurs fichiers en même temps. Il faut pour cela que la page soit en HTML5, que l'input de type file ait un attribut `multiple=""`, et mettre des crochets derrière son nom : `name = ''fichier[]''` par exemple. Dans ce cas, `$_FILES['fichier']['name']` sera non pas une simple chaîne de caractère mais un tableau contenant les noms des différents fichiers à uploader. A l'aide de ces explications, mettez en place un formulaire d'upload multiple et faites-le fonctionner.

Exercice 10 - Programmation objet

1. Faites une classe `Page_Web` munie de l'attribut privé `$titre` et des méthodes `setTitre($titre)`, `en_tete()`, `corps()`, `pied()`, `afficher_page()` chargés, respectivement, de changer le titre de la page, d'afficher l'en-tête de la page, d'afficher le corps de la page, d'afficher le pied de page, et d'afficher la page en entier.
2. Faites une classe `Page_Formulaire` qui hérite de la classe `Page_Web` et qui propose en plus les fonctions `formulaire_debut($action)`, `formulaire_fin()`, `formulaire_texte($nom)`, `formulaire_bouton()` et `afficher_page` qui permettent, respectivement, d'afficher le début du formulaire (l'action à exécuter est stockée dans la variable `$action`), d'afficher la fin du formulaire (`</form>`), d'afficher une boîte de type texte (de nom `$nom`), un bouton `envoyer`, et d'afficher la page avec le formulaire.
3. Faites une classe `Animal` avec un attribut privé `$nom` et des méthodes comme `manger`, `deplacer`, `parler...` qui affichent l'action faite par votre animal. Faites ensuite une classe `Oiseau` et une classe `Reptile` qui héritent de la classe `Animal` et qui redéfinissent les méthodes `deplacer` et `parler` (`parler()` affichera `cuicui` pour un oiseau par exemple...). Créez un oiseau et un reptile et faites les se déplacer et manger...

Exercice 11 - Bases de données

1. Nous allons créer une base de données `mySQL` à l'aide de `phpmyadmin` sur `venus` (<http://venus/phpmyadmin/>). Utilisez votre login, il n'y a pas de mot de passe. On va créer une base de données `supermarché` qui utilise trois tables : `Acheteurs`, `Produits` et `Achats`. La première représente les clients du supermarché, la deuxième les produits disponibles, et la troisième les achats du jour.

Acheteurs		
id_acheteur	prénom	compte
1	Martin	7011
2	Jean	1010
3	Pierre	2098

Produits		
id_produit	nom	prix
1	Souris	6
2	Ordi	1000
3	Livre	20

Achats		
id_achat	id_acheteur	id_produit
1	1	2
2	1	1
3	3	3

2. (SELECT) Faites une fonction php qui permette d'afficher les noms de tous les utilisateurs (grâce à une requête SQL), puis une autre fonction qui affiche les noms de tous les utilisateurs dont le prénom commence par P.
3. (SELECT) Faites une fonction qui, à l'aide d'une requête `mySQL`, affiche toutes les informations sur chaque achat du jour (`id_acheteur`, nom de l'acheteur, compte de l'acheteur, `id_achat`, `id_produit`).
4. (INSERT) Faites une fonction qui permet d'ajouter un nouvel utilisateur dans la base.
5. (UPDATE) Le numéro de compte de Jean a changé. C'est désormais le 9966. Faites un script qui à l'aide d'une requête `mySQL` met à jour la base de données.
6. (DELETE) Pierre s'est désabonné. Supprimez-le de la base de donnée.

Exercice 12 - Sessions

Faites deux pages qui ont chacune un lien l'une vers l'autre. Tant que vous n'êtes pas authentifiés, elles montrent un formulaire d'authentification (login et mot de passe (SRC2)) à remplir ; dès que l'utilisateur est identifié, le formulaire n'apparaît plus et à la place on voit la phrase "Bonjour, Dupont", si Dupont est le login rentré dans le formulaire. A la ligne suivante, ces deux pages indiquent le nombre de clics sur les liens que vous avez fait depuis que vous êtes logués.

Exercice 13 - Cookies

Créez, à l'aide d'un cookie (cf `setcookie()`), une page qui affiche le nombre de fois qu'elle a été visitée. Vous mettrez un bouton "stop" dont le rôle est d'effacer le cookie.

Exercice 14 - Manipulation d'images

Commencez par télécharger une image jpeg dans votre répertoire de travail, image sur laquelle on travaillera. Le but de l'exercice est de récupérer, grâce à la librairie GD (déjà installée), une image jpeg (fonction `imagecreatefromjpeg()`), récupérer ses dimensions dans un tableau (`getimagesize()`), l'afficher (on utilise les fonctions `header()` pour envoyer l'en-tête contenant le type MIME de l'image, et la fonction `imagejpeg()`), puis en créer une version miniaturisée (dimensions divisées par 4) grâce aux fonctions `imagecreatetruecolor()` et `imagecopyresampled()`. Cherchez, pour chaque fonction, dans le manuel pour voir quels paramètres elle prend en entrée et comment elle fonctionne.