

Licence Informatique S4 – Prog. Web - TP n°1

Pour effectuer ce TP, veuillez préalablement suivre la procédure suivante. Cliquez dans la barre de menu à gauche sur le « file browser » (en forme de casier). Une fois cela effectué, une fenêtre va s'ouvrir: cliquez sur *Autres emplacements* dans le menu de gauche de cette fenêtre, puis tapez ce qui suit dans le champs *Connexion à un serveur* : `ssh://10.1.16.112`

Après avoir validé, vous êtes invités à renseigner un *nom d'utilisateur* et un *mot de passe* : rentrez ceux qui vous ont servis à ouvrir votre session Ubuntu. Une autre fenêtre apparaît avec une liste de répertoires. Allez dans le répertoire *home*, puis dans le répertoire dont le nom correspond à la première lettre de votre login, et enfin allez dans le répertoire correspondant à votre login : ce dernier contient un répertoire *public_html*. Pour créer vos pages web, déposez vos fichiers dans ce répertoire *public_html*.

Les pages seront accessibles sur le réseau `etud-sc.u-picardie.fr` en vous rendant sur la page : `http://10.1.16.112/~votre_login/votre_fichier.php`

Exercice 1 : Créez un fichier `hello.php` ayant le contenu suivant.

```
<html>
  <body>
    <p>
      <?php echo "hello word ! <br> bonjour monde !";
            $a = 3; $b = 2;
            $c = $a + $b;
            echo "variable c = $c" ;
      ?>
    </p>
  </body>
</html>
```

Déposez le dans votre *public_html* et rendez-le lisible.
Essayez dans votre navigateur [/~votre_login/hello.php](http://10.1.16.112/~votre_login/hello.php).

Exercice 2 : Corrigez le script `acorriger1.html` ci-dessous. Puis créez le fichier correspondant dans votre *public_html*.

```
<html>
  <body>
    <p>
      <?php
        echo "Dupont ", "<br>"; // nom
        echo "Isidore" /* prenom */
        a = 1;
      ?><br>
      <?php
        echo "surnom "le dur" <br>" ;
        echo "variable a valeur : ".a ;
      >
    </p>
  </body>
</html>
```

```
</p>
</body>
</html>
```

Exercice 3 : Corrigez et créez le script `acorriger2.php` ci-dessous

```
<html>
<body>
  <h2 align=center>A corriger 2</h2>
  <p>
    <?php
      $tab = array[3, 5, 7, 1, 2, 8, 6, 4] ;
      echo 'tableau<br /><table border="1">';
      for($i = 0 ; $i < count($tab); $i++) {
        echo "<tr><td>'. $i.'</td><td>";
        echo "$tab[$i]</td></tr>";
      }
      echo '</table>';
      for(int $i = 0 ; $i < count($tab); $i++)
        somme = $tab[$i] + somme;
      echo "<br />somme = somme" ;
    ?>
  </p>
</body>
</html>
```

Exercice 4 :

Simple Calculette

Premiere valeur :

Seconde valeur :

Le formulaire HTML ci-dessus correspond à une simple calculette. Le code de ce formulaire (*calculette.html* qui est donné plus bas) appelle un script *calcul.php* qui affiche le résultat comme suit :

Simple Calculette

3 + 5 = 8

Voici le code de *calculette.html* :

```
<html>
  <body>
    <h2 align=center> Simple Calculette</h2>
    <p>
      <form method="get" action="calcul.php">
        Premiere valeur : <input type="text" name="valeur1"><br>
        Seconde valeur : <input type="text" name="valeur2"><br>
        <input type="submit" name="operation" value="+">
        <input type="submit" name="operation" value="-">
      </form>
    </p>
  </body>
</html>
```

Question 1. Ecrire le script *calcul.php*

Question 2. Afin de voir la transmission des données par le formulaire, son action est mise à « GET ». Quelles modifications seraient nécessaires pour une transmission non visible par « POST »?

Question 3. Comment peut-on effectuer l'appel à calcul.php pour faire une opération sans passer par calculette.html et sans modifier le script calcul.php ?

Exercice 5 : Reprendre la calculette de l'exercice 4 pour écrire tout dans un seul fichier.

Exercice. 6 : Complétez le script array.php pour afficher un tableau HTML des distances entre 2 villes

```
<html>
  <body>
    <p>
      <?php
        $distance= array(array("paris", "nice", 900),
```

paris	nice	900
nice	tours	800
tours	lille	500
paris	lille	300
nice	lille	1200

```

                                array("nice", "tours", 800),
                                array("tours", "lille", 500),
                                array("paris", "lille", 300),
                                array("nice", "lille", 1200) );
                                .....
                                ?>
                                </p>
                                </body>
                                </html>

```

Exercice 7 : Complétez le script array1.php pour obtenir la distance entre 2 villes qui sont données en arguments dans la barre d'adresse lors de l'appel à array1.php.

```

<html>
<body>
<p>
<?php
    if (!isset($_GET['depart']) || !isset($_GET['arrivee']))
        die('manque les villes');
    $ville1=$_GET["depart"];
    $ville2=$_GET["arrivee"];
    $distance= array(array("paris", "nice", 900),
                      array("nice", "tours", 800),
                      array("tours", "lille", 500),
                      array("paris", "lille", 300),
                      array("nice", "lille", 1200) );
    .....
    ?>
</p>
</body>

```

Exercice 8 : Compléter le script array2.php pour obtenir la distance entre 2 villes qui sont renseignées de la même façon que dans l'exercice précédent.

```

<html>
<body>
<p>
<?php
    if (!isset($_GET['depart']) || !isset($_GET['arrivee']))
        die('manque les villes');
    $ville1=$_GET["depart"];
    $ville2=$_GET["arrivee"];
    $distance=array("paris" => array("nice" => 900, "lille" => 300),
                    "nice" => array("paris" => 900, "tours" => 800, "lille" => 1200),
                    "tours" => array("nice" => 800, "lille" => 500),
                    "lille" => array("paris" => 300, "nice" => 1200, "tours" =>
800));
    .....
    ?>
</p>
</body>
</html>

```

Exercice 9 : Compléter le script array3.php qui affiche un tableau de toutes les villes destination et leur distance à partir d'une ville de départ donnée. Un exemple de l'affichage que l'on peut obtenir, avec Paris comme ville de départ, est donné plus bas.

```
<html>
<body>
  <?php
    if (!isset($_GET["depart"]))
      die("manque la ville");
    $ville1=$_GET["depart"];
    $distance=array("paris">array("nice">900,"lille">300),
                    "nice">array("paris">900,"tours">800,"lille">1200),
                    "tours">array("nice">800,"lille" =>500),
                    "lille">array("paris">300,"nice">1200,"tours">800));
    if (isset($distance[$ville1])) {
  ?>
  <h2>ville de d  part : <?php echo $_GET["depart"] ?></h2>
  <table border="1">
    <tr><th>destination</th><th>distance</th></tr>
    <?php
      ....
    } else
      echo '<p>pas de liaison    partir de cette ville</p>';
    ?>
  </body>
</html>
```

ville de départ : paris

destination	distance
nice	900
lille	300

Exercice 10 : Réalisez le jeu « plus grand, plus petit ». Un nombre entier est choisi entre 0 et 100. Le joueur essaie de deviner en se guidant à chaque fois sur la réponse fournie : soit plus grand, soit plus petit. Le nombre à deviner est généré avec la fonction `mt_rand()`. Le champ hidden permet de passer l'information à deviner d'une page à l'autre sans que l'utilisateur-client ne la voie (sauf s'il regarde le source HTML!)

Complétez le script `jeu.php` dont voici quelques indications :

```
<html>
  <body>
    <h2 align=center>Jeu : plus grand, plus petit</h2>
    <p>
      <?php ...
        if (!isset($_POST['a_trouver']))
          $a_trouver = mt_rand(0,100) ;
        ...
      ?>
      <form method="post" action="jeu.php">
        Proposez une valeur entre 0 et 100 :
        <input type="text" name="propose"><br>
        <input type="hidden" name="a_trouver" value="<?php echo $a_trouver ?>">
        <input type="submit" name="submit" value="Soumettre">
      </form>
    </p>
  </body>
</html>
```

Exercice 11 :

Le but de cet exercice est de mettre au point un mini générateur de formulaire. Vous avez déjà à votre disposition le formulaire HTML suivant (*Create_Formulaire.html*) :

```

<html>
<body>
<h3> Mon générateur de formulaires </h3>

<form method="post" action="Create_Formulaire.php">
<fieldset>
Nombre de champs <i>input type="text"</i> <input type="text"
name="nbtext"/><br/>
Nombre de champs <i>select</i> <input type="text" name="nbselect"/>
</fieldset>
<fieldset>
Pour chaque <i>select</i>, ses options doivent être sur une même ligne et
séparés par des ";". <br/>
Les options de deux <i>select</i> différents doivent être sur des lignes
différentes.<br/>

<textarea name="optionselect" rows="8" cols="70">
</textarea>
</fieldset>
<input type="submit" name="submit" value="valider"/>
</form>
</body>
</html>

```

Le rendu visuel de ce formulaire dans un navigateur est le suivant :

Mon générateur de formulaires

Nombre de champs *input type="text"*

Nombre de champs *select*

Pour chaque *select*, ses options doivent être sur une même ligne et séparés par des ";".
 Les options de deux *select* différents doivent être sur des lignes différentes.

En fait, ce formulaire invite un client à rentrer le nombre de champs *text* et le nombre de champs *select* souhaités. Ce formulaire invite également le client (via le champs *textarea*) à renseigner les *options* qu'il veut voir apparaître dans les *selects*. Un exemple de la façon dont ces champs doivent être renseignés est indiqué dans la première copie d'écran ci-dessous.

Mon générateur de formulaires

Nombre de champs *input type="text"*

Nombre de champs *select*

Pour chaque *select*, ses options doivent être sur une même ligne et séparés par des ";".
Les options de deux *select* différents doivent être sur des lignes différentes.

thème1; thème2

champs1; champs2; champs3; champs4

valider

Lorsque le client clique sur le bouton valider, les données renseignées sont envoyées au serveur, puis traitées par le fichier *Create_Formulaire.php*.

Pour cet exercice, vous devez donc écrire le contenu du fichier

Create_Formulaire.php. Ce dernier devra renvoyer au navigateur du client un formulaire HTML contenant le nombre de champs *text* et *select* choisis par le client dans le formulaire *Create_Formulaire.html*. Par ailleurs, les options des *selects* devront correspondre à ceux mentionnés dans le champs *textarea*.

On supposera que le client renseigne toujours correctement les champs du formulaire *Create_Formulaire.html*. En particulier, le nombre de lignes insérées dans le *textarea* est toujours égale au nombre de *select* indiqué par le client. En fait, la première ligne correspondra aux options du premier *select*, la seconde ligne aux options du second *select* etc... Par ailleurs, si le nombre de *select* souhaités n'est pas égal à 0, l'utilisateur n'insère aucune ligne vide dans le *textarea*.

Ci-dessous, deux figures illustrant le rendu visuel que doit avoir le formulaire généré via *Create_Formulaire.php* avec les valeurs indiquées dans la figure située juste au-dessus. Dans le code html du formulaire généré, il n'est pas nécessaire d'avoir des attributs *name* pour les champs ni d'avoir des attributs pour la balise *form*.

Mon Formulaire

thème1 ▾

champs1 ▾

Submit Query

Mon Formulaire

thème1 ▾

champs1 ▾

champs1Query

champs2

champs3

champs4