**PHP TD1**

***1. Introduction***

1.1 Essai préliminaire

L’affichage du lien HTML donne une page entièrement blanche tandis que l’affichage du lien PHP montre l’écriture du mot « Bonjour ». Le fait d’ouvrir le fichier tst.php ne donne pas le résultat directement car on ne passe pas par un interpréteur PHP.

1.2 Message d’erreur

Le message d’erreur est :

**Parse error**: syntax error, unexpected '’' (T\_STRING), expecting ')' in **/home/ravelli1u/PHP\_TD1/tst.php** on line **4**

C’est l’interpréteur qui envoie ce message.

Ce message indique qu’il y a une erreur à la ligne 4, car c’est un String qui est attendu pour la variable $texte.

Une fois l’erreur de syntaxe corrigé, on peut conclure qu’il y a trois manières d’appeler une fonction : Soit en donnant des valeurs à chacun des arguments de la fonction, soit en donnant une valeur aux arguments vides, soit en en donnant aucuns.

1.3 Portée des variables

$var1 qui est défini dans l’entête du code est toujours accessible dans le corps du code. Il en est de même pour $var2.

Lorsque l’on appelle la fonction test pour la première fois, celle-ci modifie les valeurs des var1 et var2 pour ensuite les afficher. Cependant lorsque l’on fait à nouveau appel au $var1 et $var2 dans le corps du code, la $var1 à garder la valeur que le premier appel à la fonction test lui a affecter (ligne 11 :  *global $var1*), mais la $var2 a repris sa valeur initiale.

Lorsque l’on appelle la fonction test pour la deuxième fois, les valeurs des deux variables reprennent les mêmes valeurs que précédemment.

Mais appeler la variable $nb en dehors de la fonction test ne fonctionne pas car cette variable est indéfinie en dehors de la fonction test.

***2. Boucles***

2.1 Chiffres

Avec while :

<html>

<head>

<title> Exercice 2 </title>

</head>

<body>

<ul>

<?php

$i = 0;

while ($i <10)

{

echo '<li>';

echo $i++;

echo'</li>';

}

?>

</ul>

</body>

</html>

Avec for :

<html>

<head>

<title> Exercice 2 </title>

</head>

<body>

<ul>

<?php

for ($i=0; $i <10 ; $i++)

{

echo '<li>';

echo $i;

echo'</li>';

}

?>

</ul>

</body>

</html>

2.2 Triangle

<html>

<head>

<title>Exercice 2.2 </title>

</head>

<body>

<?php echo '<center>' ;

function triangle ($taille) {

$etoile ="\*";

for ($i=0 ; $i <$taille ; $i++) {

echo $etoile;

echo ' <br /> ';

$etoile = $etoile."\*";

}

echo '</center)>';

}

triangle (12);

?>

</body>

</html>

2.3 Tableau

<html>

<head>

<title>Exercice 2.3</title>

</head>

<body>

<?php

function tableau ($nb=10)

{

echo '<table>';

$elt=0;

for ($i=0;$i<$nb;$i++)

{

echo '<tr>';

for ($j=0; $j<$nb; $j++)

{

$elt++;

echo "<td> $elt </td>";

}

echo '</tr>';

}

echo '</table>';

}

tableau();

?>

</body>

</html>

**3. Manipulation de chaînes de caractères**

3.1 Syntaxes de spécifications de chaînes de caractères

<html>

<head>

<title>exercice 3.1</title>

</head>

<body>

<?php

echo '" guillemets " \' quotes \' et \ antislash \ ';

echo '<br />';

echo"\" guillemets \" ' quotes ' et \ antislash \ ";

echo '<br />';

echo '" guillemets "' . " ' quotes ' " . " et \ antislash \ ";

?>

</body>

</html>

3.2 Suppression des blancs initiaux et finaux

<html>

<head>

<title>exercice 3.2</title>

</head>

<body>

<?php

echo trim($phrase = " une phrase avec trop de blancs au début");

?>

</body>

</html>

3.3 Récupérer les mots d’une chaîne

<html>

<head>

<title>exercice </title>

</head>

<body>

<?php

?>

</body>

</html>