

TP2 PWS :

Manipulation de tableaux, Fonctions PHP et Gestion d'erreurs

Table des matières

Introduction :.....	3
Manipulation de Tableaux.....	4
Exercice 1.....	4
Exercice 2.....	4
Exercice 3.....	4
Exercice 4.....	5
Exercice 5.....	5
Exception.....	6
Conclusion :	7

Introduction :

L'objectif de ce TP est de voir comment utiliser des **tableaux multidimensionnels associatifs**, les **fonctions** et les **exceptions** en *Php*. Puis comprendre leur rendu.

Pour ce faire , nous utiliserons MobaXterm.

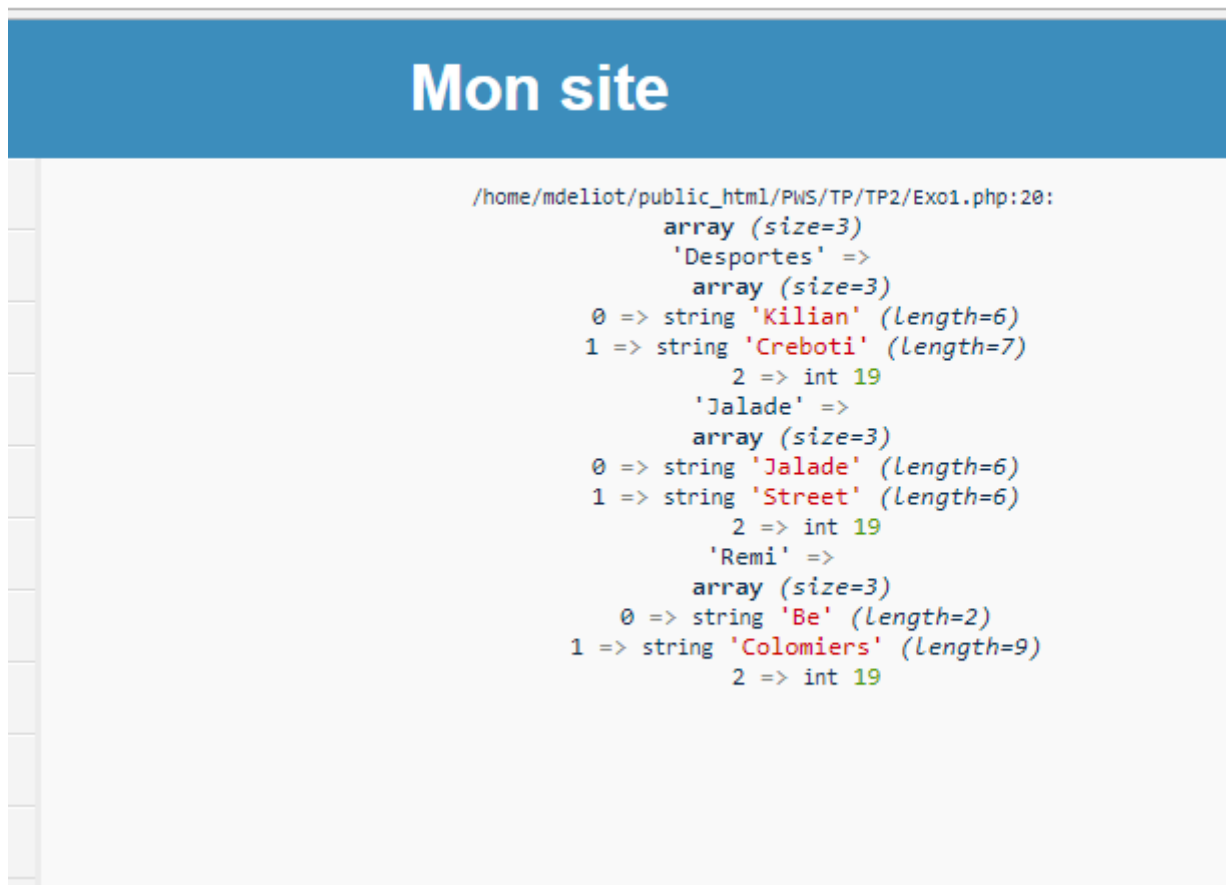
Manipulation de Tableaux

Exercice 1

Définir et afficher un tableau multidimensionnel associatif dont les clés sont des noms de personne et les valeurs des tableaux indicés contenant le prénom, la ville de résidence et l'âge de la personne.

Rendu

<ps://webdev.iut-blagnac.fr/~mdeliot/PWS/TP/TP2/Exo1.php>



```
/home/mdeliot/public_html/PWS/TP/TP2/Exo1.php:20:
    array (size=3)
        'Desportes' =>
            array (size=3)
                0 => string 'Kilian' (length=6)
                1 => string 'Creboti' (length=7)
                2 => int 19
        'Jalade' =>
            array (size=3)
                0 => string 'Jalade' (length=6)
                1 => string 'Street' (length=6)
                2 => int 19
        'Remi' =>
            array (size=3)
                0 => string 'Be' (length=2)
                1 => string 'Colomiers' (length=9)
                2 => int 19
```

Code commenté

```
<?php
$stab= array('Desportes'=>array(0=>'Kilian',1=>'Creboti',2=>19),
            'Jalade'=>array(0=>'Jalade',1=>'Street',2=>19),
            'Remi'=>array(0=>'Be',1=>'Colomiers',2=>19));
var_dump($stab);

?>
```

Ce code permet de créer le tableau multidimensionnel demandé. Pour créer un tableau, on utilise **array()** dans lequel se trouve ici une clé vers un autre tableau simple.

Par exemple pour le premier la clé "principal" est : 'Desportes' et sa valeur sera le tableau qui contient les valeurs des clés 0,1 et 2.

Le **var_dump()** permet d'afficher le tableau en entier rapidement (une seule instruction).

Exercice 2

Définir et afficher un tableau multidimensionnel associatif dont les clés sont des noms de personne et les valeurs des tableaux associatifs dont les clés sont le prénom, la ville de résidence et l'âge de la personne avec une série de valeurs associées.

Rendu

```
/home/mdeliot/public_html/PWS/TP/TP2/Exo2.php:20:
array (size=3)
  'Desportes' =>
    array (size=3)
      'prenom' => string 'Kilian' (length=6)
      'ville' => string 'Creboti' (length=7)
      'age' => int 19
  'Jalade' =>
    array (size=3)
      'prenom' => string 'Jalade' (length=6)
      'ville' => string 'Street' (length=6)
      'age' => int 19
  'Remi' =>
    array (size=3)
      'prenom' => string 'Be' (length=2)
      'ville' => string 'Colomiers' (length=9)
      'age' => int 19
```

Code commenté

```
<?php
    $tab= array('Desportes'=>array('prenom'=>'Kilian','ville'=>'Creboti','age'=>19),
                'Jalade'=>array('prenom'=>'Jalade','ville'=>'Street','age'=>19),
                'Remi'=>array('prenom'=>'Be','ville'=>'Colomiers','age'=>19));
    var_dump($tab);
?>
```

Ici, c'est le même principe de construction, sauf que les clés 0,1 et 2 du sous-tableau ont été remplacé par 'prenom' , 'ville' et 'age'.

Exercice 3

Utiliser une boucle foreach pour lire les tableaux des exercices 1 et 2.

Rendu

<http://webdev.iut-blagnac.fr/~mdeliot/PWS/TP/TP2/Exo3.php>

Mon site

Element Desportes
Element 0 : Kilian
Element 1 : Creboti
Element 2 : 19

Element Jalade
Element 0 : Jalade
Element 1 : Street
Element 2 : 19

Element Remi
Element 0 : Be
Element 1 : Colomiers
Element 2 : 19

Element Desportes
prenom : Kilian
ville : Creboti
age : 19

Element Jalade
prenom : Jalade
ville : Street
age : 19

Element Remi
prenom : Be
ville : Colomiers
age : 19

Code commenté

```

<?php
$tab= array('Desportes'=>array(0=>'Kilian',1=>'Creboti',2=>19),
            'Jalade'=>array(0=>'Jalade',1=>'Street',2=>19),
            'Remi'=>array(0=>'Be',1=>'Colomiers',2=>19));
foreach($tab as $personne=>$val){
    echo '<b>Element '.$personne.'<br/></b>';
    foreach($val as $champ=>$val2){
        echo 'Element '.$champ.' : '.$val2.'<br/>';
    }
}
echo '<BR/>';

$tab2= array('Desportes'=>array('prenom'=>'Kilian','ville'=>'Creboti','age'=>19),
            'Jalade'=>array('prenom'=>'Jalade','ville'=>'Street','age'=>19),
            'Remi'=>array('prenom'=>'Be','ville'=>'Colomiers','age'=>19));
foreach($tab2 as $personne=>$val){
    echo '<b>Element '.$personne.'<br/></b>';
    foreach($val as $champ=>$val2){
        echo $champ.' : '.$val2.'<br/>';
    }
}
echo '<BR/>';
?>

```

Pour les deux tableaux, le **foreach** doit s'effectuer de la même manière. Pour tous les couples clés valeur, où la clé correspond à 'Desportes', 'Jalade' et 'Remi' et la valeur est leur tableau "personnel", il exécutera un autre for pour chaque couple clé valeur du tableau "personnel" dont les clés seront 0,1 et 2 pour le premier et 'nom', 'ville' et 'age' pour le second. Pour chacune de ces clés on affichera sa valeur.

Exercice 4

En utilisant la documentation de PHP, trouver et tester les fonctions permettant :

- a°) de trier les éléments du tableau de l'exercice 1 par prénom croissant
- b°) de tirer un élément au hasard du tableau de l'exercice 2
- c°) de mélanger les éléments d'un tableau indexé contenant quelques couleurs (blanc, bleu, jaune, rouge...)

Rendu

```
Mon site

Trace a
Avant tri
/home/mdeliot/public_html/PWS/TP/TP2/Exo4.php:22:
array (size=3)
  'Desportes' =>
    array (size=3)
      0 => string 'Kilian' (length=6)
      1 => string 'Creboti' (length=7)
      2 => int 19
  'Jalade' =>
    array (size=3)
      0 => string 'Jalade' (length=6)
      1 => string 'Street' (length=6)
      2 => int 19
  'Remi' =>
    array (size=3)
      0 => string 'Be' (length=2)
      1 => string 'Colomiers' (length=9)
      2 => int 19
Après tri
/home/mdeliot/public_html/PWS/TP/TP2/Exo4.php:25:
array (size=3)
  'Remi' =>
    array (size=3)
      0 => string 'Be' (length=2)
      1 => string 'Colomiers' (length=9)
      2 => int 19
  'Jalade' =>
    array (size=3)
      0 => string 'Jalade' (length=6)
      1 => string 'Street' (length=6)
      2 => int 19
  'Desportes' =>
    array (size=3)
      0 => string 'Kilian' (length=6)
      1 => string 'Creboti' (length=7)
      2 => int 19
```

Trace b

Tirage element hasard : Desportes

Trace c

Avant melange

```
/home/mdeliot/public_html/PWS/TP/TP2/Exo4.php:35:
array (size=3)
  0 => string 'blanc' (length=5)
  1 => string 'bleu' (length=4)
  2 => string 'rouge' (length=5)
```

Après melange

```
/home/mdeliot/public_html/PWS/TP/TP2/Exo4.php:38:
array (size=3)
  0 => string 'bleu' (length=4)
  1 => string 'blanc' (length=5)
  2 => string 'rouge' (length=5)
```

Après second melange

```
/home/mdeliot/public_html/PWS/TP/TP2/Exo4.php:41:
array (size=3)
  0 => string 'rouge' (length=5)
  1 => string 'bleu' (length=4)
  2 => string 'blanc' (length=5)
```

Contact : maxence.deliot@etu.univ-tlse2.fr

Code commenté

Trace a :

```
<?php
echo '<H2>Trace a</H2>';
echo 'Avant tri';
$tab= array('Desportes'=>array(0=>'Kilian',1=>'Creboti',2=>19),
            'Jalade'=>array(0=>'Jalade',1=>'Street',2=>19),
            'Remi'=>array(0=>'Be',1=>'Colomiers',2=>19));
var_dump($tab);
echo 'Après tri';
array_multisort($tab,SORT_ASC);
var_dump($tab);
```

Pour trier un tableau, on utilise la fonction **array_multisort()** avec en paramètre le tableau et l'ordre de tri.

Trace b :

```
echo '<H2>Trace b</H2>';
$tab2= array('Desportes'=>array('prenom'=>'Kilian','ville'=>'Creboti','age'=>19),
            'Jalade'=>array('prenom'=>'Jalade','ville'=>'Street','age'=>19),
            'Remi'=>array('prenom'=>'Be','ville'=>'Colomiers','age'=>19));
$hasard=array_rand($tab2);
echo'<BR/> Tirage element hasard : '.$hasard.'<BR/>';
```

Pour tirer au hasard, une valeur d'un tableau, on utilise la fonction ***array_rand()*** avec le tableau en paramètre.

Trace c :

```
echo '<H2>Trace c</H2>';
echo 'Avant melange <BR/>';
$tab3= array('0'=>'blanc','1'=>'bleu','2'=>'rouge');
var_dump($tab3);
echo 'Apres melange <BR/>';
shuffle($tab3);
var_dump($tab3);
echo 'Apres second melange <BR/>';
shuffle($tab3);
var_dump($tab3);
```

Pour mélanger un tableau, on utilise la fonction ***shuffle()*** qui prend le tableau en paramètre.

Exercice 5

Créez un tableau contenant une liste d'adresses e-mail. Extrayez le nom de serveur de ces données (free.fr, gmail.com...) en trouvant et utilisant la bonne fonction de PHP pour l'extraction dans une chaîne de caractères. Afficher le tableau contenant la liste de ces serveurs de données.

Rendu

Mon site

Le tableau des emails

```
/home/mdeliot/public_html/PWS/TP/TP2/Exo5.php:19:  
array (size=3)  
0 => string 'maxencedeliot@gmail.com' (length=23)  
1 => string '1@ggg.fr' (length=8)  
2 => string '2@aol.eu' (length=8)
```

Le tableau des serveurs

```
/home/mdeliot/public_html/PWS/TP/TP2/Exo5.php:25:  
array (size=3)  
0 =>  
array (size=1)  
0 => string 'gmail.com' (length=9)  
1 =>  
array (size=1)  
1 => string 'ggg.fr' (length=6)  
2 =>  
array (size=1)  
2 => string 'aol.eu' (length=6)
```

Code commenté

```
<?php
    $tab= array(0=>'maxencedeliot@gmail.com',1=>'1@ggg.fr',2=>'2@aol.eu');
    echo "Le tableau des emails<BR/>";
    var_dump($tab);
    echo "<BR/>Le tableau des serveurs<BR/>";
    foreach($tab as $cle=>$val){
        $serveur=explode('@',$val);
        $tabSer[$cle]=array($cle=>$serveur[1]);
    }
    var_dump($tabSer);
?>
```

Le tableau est simple donc on peut le parcourir simplement avec un **foreach()**. Pour récupérer le contenu après le '@' de chaque adresse mails ce qui correspond au nom de serveur, on utilisera la fonction **explode()**. Elle prend en paramètre :

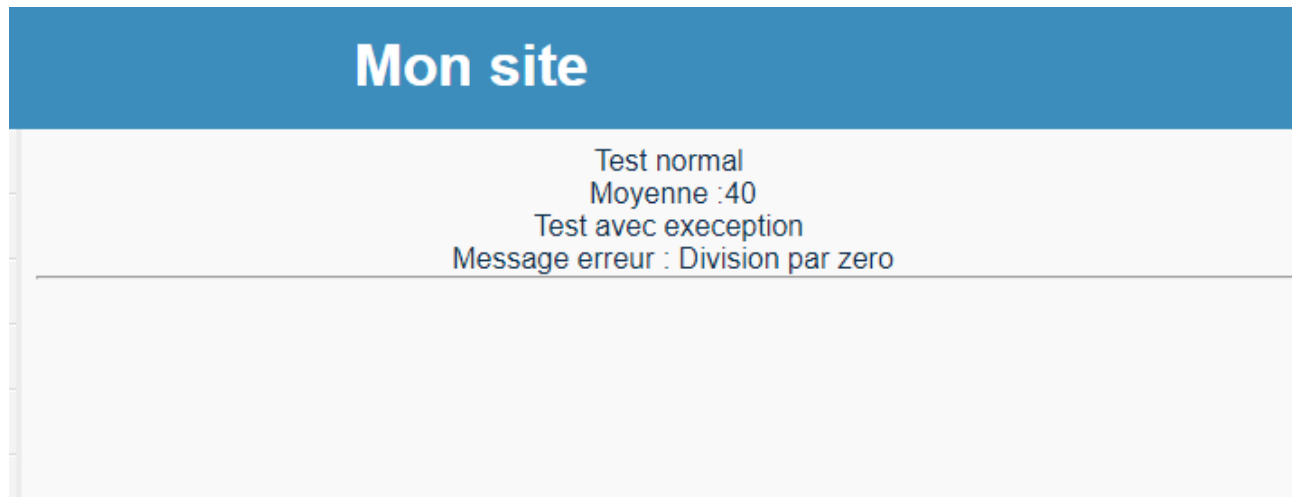
- Le caractère qui à chaque fois qu'il sera trouvé coupera le mot en deux et affectera ces deux mots à un tableau dans lequel tab[0] sera la partie gauche du caractère et tab[1] la partie droite.
- La chaîne de caractère à séparer. Ici, ce sera la valeur de chaque couple.

On affectera à un nouveau tableau la partie droite tab[1] pour n'avoir que le nom de serveur puis on l'affichera.

Exception

Coder et tester la fonction avec exception de l'exercice 4 du TD2.

Rendu



Code commenté

```
<?php
function moyAge($tab) {
    $somme=0;
    $cpt=0;

    foreach($tab as $cle=>$val) {
        if(array_key_exists("age", $val)) {
            $somme+=$val["age"];
            $cpt++;
        }
    }

    if($cpt==0) {
        throw new Exception("Division par zero");
    }else{
        return $somme/$cpt;
    }
}
```

```

require("moyAge.php");
$stab2= array('Desportes'=>array('prenom'=>'Kilian','ville'=>'Creboti','age'=>50),
              'Jalade'=>array('prenom'=>'Jalade','ville'=>'Street','age'=>50),
              'Remi'=>array('prenom'=>'Be','ville'=>'Colomiers','age'=>20));
echo 'Test normal <BR/>';
$moy=moyAge($stab2);
echo 'Moyenne :'.$moy;
echo '<BR/>Test avec exeception <BR/>';
$stab3= array('Desportes'=>array('prenom'=>'Kilian','ville'=>'Creboti'),
              'Jalade'=>array('prenom'=>'Jalade','ville'=>'Street'),
              'Remi'=>array('prenom'=>'Be','ville'=>'Colomiers'));
try{
    $moy2=moyAge($stab3);
    echo 'Moyenne :'.$moy2;
}catch(Exception $e){
    echo "Message erreur : ".$e->getMessage().'\n';
}
?>

```

Tout d'abord pour créer une fonction, on utilise la structure : `function nomdelafunction([paramètres possibles]){}.` Ensuite l'algorithme est simple, il suffit de parcourir le tableau, de regarder si chaque tableau "personnel" contient la clé 'age'. On effectue ce teste grace à `array_key_exists()`. Si oui alors on la recupère et les additionne avec l'age des autres personnes `$somme` et on ajoute +1 au compteur de personne `$cpt`. Après avoir parcouru toutes les clés, on regarde si le compteur de personne est égal à zero alors on soulève une exception grace à `throws new Exception(..)`. Sinon retourne la moyenne qui est `$somme/$cpt`.

Ensuite pour l'appeler dans notre page il faut :

- L'importer grâce à `require()`
- Récupérer sa valeur quand on l'appelle et faire cet appel dans un `try{} catch(){}.`


Coder et tester l'exemple d'Error_reporting vu en cours.

Rendu

Mon site

Test normal
Moyenne : 40
Test avec exeception
Message erreur : Division par zero

Test avec error_reporting

 Warning: Division by zero in /home/mdeliot/public_html/PWS/TP/TP2/moyAge2.php on line 13

Call Stack				
#	Time	Memory	Function	Location
1	0.0003	241480	{main}()	.../Part2.php:0
2	0.0013	252984	moyAge2()	.../Part2.php:36

Moyenne :

Code commenté

```
<?php
function moyAge2 ($tab) {
    error_reporting(2);
    $somme=0;
    $cpt=0;

    foreach($tab as $cle=>$val){
        if(array_key_exists("age", $val)){
            $somme+=$val["age"];
            $cpt++;
        }
    }
    return $somme/$cpt;
}
?>

require("moyAge2.php");
echo '<BR/>Test avec error_reporting <BR/>';
$moy2=moyAge2($tab3);
echo '<BR/>Moyenne : '.$moy2;
```


La fonction MoyAge2 est identique à MoyAge sauf que cette fois on utilise ***error_reporting()*** pour gérer l'exception du zéro. Avec ***error_reporting(2)*** on affiche ce qui ne va pas mais on n'arrete pas le script, c'est ce qui est le mieux adapté à ce que l'on veut.

Pour l'appeler plus besoin de ***try{} catch(){}.***

Conclusion :

A l'issue de ce TP, nous savons donc comment utiliser un tableau multidimensionnel associatif.
Mais aussi les fonctions *Php* et comment y lever une exception spécifique.