

TD1: PHP(Corrigé)

Exercice 1 Variables et conversion

Question 1.1 : Qu'affiche le code suivant ?

```
1  var_dump(1 + 2)    ; echo '<br/>';
2  var_dump('1' + '2'); echo '<br/>';
3  var_dump(1 . 2)    ; echo '<br/>';
4  var_dump('1' . '2'); echo '<br/>';
5  var_dump("3 fruits" + '1 légume' . 'donnent'); echo '<br/>';
6  var_dump("3 fruits" . '1 légume' + 'donnent'); echo '<br/>';
7  var_dump(1/2 + true - "-2"); echo '<br/>';
```

Correction :

```
1  int(3)
2  int(3)
3  string(2) "12"
4  string(2) "12"
5  string(8) "4donnent"
6  int(3)
7  float(3.5)
```



Question 1.2 : Considérons les variables `$voitures = 3`; `$motos = 6`; Afficher, à l'aide d'une seule instruction PHP, le paragraphe : *J'ai 3 voitures et 6 motos, j'ai donc 9 véhicules.*

Les nombres 3, 6 et 9 dépendent des valeurs des variables données précédemment. Le faire une fois en utilisant uniquement les apostrophes et une fois en utilisant uniquement les guillemets.

Correction :

```
1  echo '<p> J\'ai ' . $voitures . ' voitures et ' . $motos . ', j\'ai donc ' .
2    ($voitures + $motos) . ' véhicules </p>';
3
4  echo "<p> J'ai $voitures voitures et $motos, j'ai donc ".
5    ($voitures + $motos) . "véhicules </p>";
```



Exercice 2 Boucles

Écrire la table de multiplication de 7 (de 1 à 10). Cette table sera présentée d'abord sous forme d'une liste non ordonnée puis sous la forme d'une table HTML.

Correction :

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4    <meta charset="UTF-8">
5    <title>Table de multiplication</title>
6  </head>
7  <body>
8    <?php
9    echo '<ul>';
10   for($i = 1 ; $i <= 10 ; $i++)
11     echo '<li>' . $i . ' * 7 = ' . ($i * 7) . '</li>';
12   echo '</ul>';
13
14
15   echo '<table><tr><td></td><td> * 7 </td></tr>';
16   $i = 1;
17   while ($i <= 10) {
18     echo '<tr><td>' . $i . '</td><td>' . ($i * 7) . '</td></tr>';
19     $i++;
20   }
21   echo '</table>';
22   ?>
23 </body>
24 </html>

```



Exercice 3 Tableaux

Question 3.1 : Soit le tableau `$t = ['a'=>'z', 10=>20, 'toto'=>'titi', 'php']`. Quelles sont les clés du tableau ? Quelles sont les valeurs ?

Correction :

- Clés : 'a', 10, 'toto', 11
- Valeurs : 'z', 20, 'titi', 'php'



Question 3.2 : Définir une variable de type tableau. Ce tableau devra contenir tous les types de données PHP possibles pour les clés et pour les valeurs.

Correction :

- Types de données possibles pour les clés : entier, string
- Types de données possibles pour les valeurs : entier, float, string, booléens, tableau (array).

```
1 $tab = [ 'a' => 'b', 23, 23.5, 18 => false, [ 1,2,3 ] ];
```

Il existe également le type object pour les valeurs. Nous verrons ce type plus tard.



Question 3.3 : Ajouter à ce tableau deux valeurs 'jour' et 'nuit'. La clé associée à la première valeur sera 'bon'. La deuxième clé sera donnée par l'interpréteur PHP. Quelle clé donnera-t-il ?

Correction :

```

1 $tab['bon'] = 'jour';
2 $tab[] = 'nuit'; // La clé sera 20.

```



Exercice 4 Parcours de tableaux

Le tableau suivant donne les noms des étudiants et leur note en PHP.

```

1 $tabE1 = [
2     'Frédérique' => 12,
3     'Silvia' => 3,
4     'Julien' => 19,
5     'Lionel' => 12,
6     'Mario' => 12,
7     'Andrea' => 9,
8     'Gérard' => 3
9 ];

```

Question 4.1 : Afficher sous forme d'une liste ordonnée les noms des différents étudiants.

Correction :

```

1 echo '<ol>';
2 foreach ($tabE1 as $c => $v)
3     echo '<li>' . $c . '</li>';
4 echo '</ol>';

```

◇

Question 4.2 : Afficher sous forme d'une table HTML les noms et les notes des étudiants.

Correction :

```

1 echo '<table><tr><td>Nom</td><td>Note</td></tr>';
2 foreach ($tabE1 as $c => $v) {
3     echo '<tr><td>' . $c . '</td><td>' . $v . '</td></tr>';
4 }
5 echo '</table>';

```

◇

Question 4.3 : Créer un autre tableau `$tabE2` dans lequel les notes sont maintenant les clés.

Correction :

```

1 $tabE2 = [
2     3 => ['Silvia', 'Gérard'],
3     9 => ['Andrea'],
4     12 => ['Frédérique', 'Lionel', 'Mario'],
5     19 => ['Julien']
6 ];

```

◇

Question 4.4 : Refaire les deux premières questions avec le tableau `$tabE2`.

Correction :

```

1 echo '<ol>';
2 foreach ($tabE2 as $vTab)
3     foreach ($vTab as $v) {
4         echo '<li>' . $v . '</li>';
5     }

```

```

6  echo '</ol>';
7
8  echo '<table><tr><td>Nom</td><td>Note</td></tr>';
9  foreach ($tabE2 as $c => $vTab)
10     foreach($vTab as $v)
11         echo '<tr><td>' . $v . '</td><td>' . $c . '</td></tr>';
12 echo '</table>';

```



Exercice 5 Fonctions sur les tableaux I

Considérons le tableau suivant.

```

1  $anneeScolaire = [
2      "Rentrée" => [2,9,2019],
3      "Vacances Toussaint" => [
4          "début" => [19,10,2019],
5          "fin" => [4,11,2019]
6      ],
7      "Vacances Noël" =>[
8          "début" => [21,12,2019],
9          "fin" => [6,1,2020]
10     ],
11     "Vacances d'hiver" =>[
12         "début" => [8,2,2020],
13         "fin" => [24,2,2020]
14     ],
15     "Vacances de printemps" =>[
16         "début" => [4,4,2020],
17         "fin" => [20,4,2020]
18     ],
19     "Fin des cours" => [4,7,2020]
20 ];

```

Écrire le code permettant d'obtenir l'affichage suivant.

```

1  - Rentrée : 2/9/2019
2  - Vacances Toussaint : du 19/10/2019 au 4/11/2019
3  - Vacances Noël : du 21/12/2019 au 6/1/2020
4  - Vacances d'hiver : du 8/2/2020 au 24/2/2020
5  - Vacances de printemps : du 4/4/2020 au 20/4/2020
6  - Fin des cours : 4/7/2020

```



On utilisera uniquement deux boucles **foreach** imbriquées ainsi que les fonctions **count** et **implode**.

Correction :

```

1  echo '<ul>';
2  foreach($anneeScolaire as $c => $vTab) {
3      echo '<li>' . $c . ' : ' ;
4      if(count($vTab)==3)
5          echo implode('/', $vTab) . '</li>';
6      else
7          echo 'du ' . implode('/', $vTab['début']) . ' au '
8              . implode('/', $vTab['fin']) . '</li>';
9  }
10 echo '</ul>';

```



Exercice 6 Fonctions sur les tableaux II

Question 6.1 : Qu'affiche le code suivant ?

```

1 $tab = [ 2 => 'test', 0 => 'nombre' ];
2
3 if (in_array(2, $tab))
4     echo '<p> in_array : 2 est dans le tableau </p>';
5
6 if(in_array('nombre', $tab))
7     echo '<p> in_array : nombre est dans le tableau </p>';

```

Correction :

in_array : nombre est dans le tableau



Question 6.2 : Dans le `if`, peut-on remplacer `in_array` par `array_search` ?

Correction :

Il est possible de remplacer le deuxième test par `array_search('nombre', $tab) !== false` (pour le premier il faut changer 'nombre' par 2). Par contre, il faut impérativement `!== false` pour que l'interpréteur fasse une comparaison stricte avec `false`. Autrement, comme la valeur retournée est 0, elle est automatiquement castée en booléen donc en `false`.



Exercice 7 Fonctions

Considérons le tableau suivant :

```

1 $joueurs = [
2     ['nom' => 'Mehwish', 'score' => 150],
3     ['nom' => 'Laurent', 'score' => 120],
4     ['nom' => 'Ines', 'score' => 98],
5     ['nom' => 'Sondes', 'score' => 153],
6     ['nom' => 'Davide', 'score' => 118]
7 ];

```

Écrire une fonction `meilleur_joueur()` prenant en paramètre un tableau ayant la même structure que le tableau `$joueurs` et retournant un tableau contenant le nom et le score du meilleur joueur.

Correction :

```

1 function meilleur_joueur($t) {
2     $max = $t[0];
3     foreach ($t as $v) {
4         if($v['score'] > $max['score'])
5             $max = $v;
6     }
7     return $max;
8 }
9 $res = meilleur_joueur($joueurs);
10 echo '<p> Meilleur joueur : ' . $res['nom'] . ' avec un score de '
11     . $res['score'] . '</p>';

```



Exercice 8 Objets

Définir une classe `Compte` représentant un compte bancaire. Cette classe contiendra deux attributs privés, `montant` et `interet` (annuel), un constructeur prenant en paramètre un montant initial et un intérêt, une méthode `get_montant` retournant le montant du compte, ainsi qu'une méthode `un_an` modifiant le montant du compte au bout d'un an en fonction de l'intérêt.

Créer deux comptes, un d'un montant de 200 euros à 20% et l'autre d'un montant de 1000 euros à 2%. Afficher le montant de chaque compte au bout de 10 ans.

Correction :

```
1  class Compte {
2      private $montant;
3      private $interet; //En pourcentage
4
5      public function __construct($interet, $montantInitial) {
6          $this->montant = $montantInitial;
7          $this->interet = $interet;
8      }
9
10     public function un_an() {
11         $this->montant *= 1+ $this->interet / 100;
12     }
13
14     public function get_montant() {
15         return $this->montant;
16     }
17 }
18
19 $c1 = new Compte(20,200);
20 $c2 = new Compte(2,1000);
21 for($i = 1 ; $i <= 10 ; $i++)
22 {
23     $c1->un_an();
24     $c2->un_an();
25 }
26 echo '<p> Valeur du compte = ' . $c1->get_montant() . ' euros </p>';
27 echo '<p> Valeur du compte = ' . $c2->get_montant() . ' euros </p>';
```

◇

Exercice 9 Expressions régulières

Question 9.1 : Donner l'expression régulière correspondant à un nombre entier. Il est possible d'avoir un nombre négatif.

Correction :

```
'#^-?\d+$#'
```

◇

Question 9.2 : Donner l'expression régulière correspondant à un nombre décimal. On utilisera le symbole `.` comme séparateur décimal. Pour simplifier, on supposera qu'il y a forcément un chiffre après le séparateur décimal si celui-ci est donné. Les valeurs 10.0, 3.14, -42, 0.001, .001 doivent satisfaire l'ER mais 10. non.

Correction :

```
'#^-?\d*\.?\d+$#'
```

◇

Question 9.3 : Donner l'expression régulière correspondant aux dates au format JJ/MM/AAAA. Il est possible de ne spécifier qu'un chiffre pour le jour et le mois.

Correction :

`'#^(\\d{1,2}/){2}\\d{4}$#'`

◇

Question 9.4 : Parmi les différents ensembles de chaînes de caractères, lequel contient uniquement des chaînes satisfaisant l'expression régulière `'#[bon]?j+(our)?#'` ?

- ☐ 'bonjour', 'jour', 'j'
- ☐ 'bjour', 'jour', 'jr'
- ☐ 'our', 'jour', 'j'

Correction :

`'bjour', 'jour', 'jr'`

◇

Question 9.5 : Parmi les différents ensembles de chaînes de caractères, lequel contient uniquement des chaînes satisfaisant l'expression régulière `'#^a.b(ba{3}|c)$#'` ?

- ☐ 'a.bc', 'abc', 'anbc'
- ☐ 'a.bc', 'aabbaaa', 'anbc'
- ☐ 'aabc', 'aabbababa', 'anbc'

Correction :

`'a.bc', 'aabbaaa', 'anbc'`

◇

Exercice 10 Passage de paramètres dans l'url

Question 10.1 : Créer un script `nombre.php` qui teste s'il existe un paramètre de nom `nombre`. Si c'est le cas, alors le script teste si sa valeur est un nombre ou pas et affiche cette information.

Question 10.2 : Donner l'url appelant le script `nombre.php` en lui passant le nombre 25. On suppose que `nombre.php` se trouve sur le serveur `serveurWeb` à la racine.

Correction :

```

1 //Une valeur a été soumise
2 if(isset($_GET['nombre'])) {
3     //Si c'est bien un nombre flottant
4     if( preg_match("#^~?\\d*\\.?\\d+?$#", $_GET['nombre']) )
5         echo '<p> La valeur saisie est un nombre </p>';
6     else
7         echo '<p> La valeur saisie n\\'est pas un nombre </p>';
8 }
9 //Appel : http://serveurWeb/nombre.php?nombre=25

```

◇

Exercice 11 Formulaire

Créer un formulaire avec un champ de type texte (pour saisir un nombre) et un bouton `submit`. La soumission du formulaire appellera le fichier `nombre.php` défini dans l'exercice précédent.

Correction :

```

1 <form action="nombre.php">
2     <p>
3         Nombre :<input type="text" name="nombre"/>
4         <input type="submit" value="Multiplier"/>

```

```
5     </p>
6 </form>
```



Exercice 12 Sessions et cookies

On suppose que le formulaire précédent est dans le fichier `nombre.php` défini dans l'exercice 10.

Question 12.1 : Utiliser les sessions pour faire la multiplication des nombres saisis dans le formulaire.

Correction :

```
1  session_start();
2
3  //On crée le compteur s'il n'existe pas
4  if(! isset($_SESSION['produit']))
5      $_SESSION['produit'] = 1;
6
7  //Une valeur a été soumise
8  if(isset($_GET['nombre'])) {
9      //Si c'est bien un nombre flottant
10     if( preg_match("#^-?\d*\.\d+$#", $_GET['nombre']) ) {
11         $_SESSION['produit'] *= $_GET['nombre'];
12     }
13     else
14         echo '<p> La valeur saisie n\'est pas un nombre </p>';
15 }
16 echo '<p> Total = ' . $_SESSION['produit'] . '</p>';
```



Question 12.2 : Pourquoi est-ce important de vérifier que le nombre saisi correspond à un nombre ? Que se passerait-il si aucune vérification n'était faite et que l'utilisateur soumettait le formulaire sans avoir saisi de nombre ?

Correction :

Si aucun contrôle sur la saisie n'est fait, alors le script va multiplier le total par la valeur saisie convertie en nombre dans tous les cas. Si cette valeur ne correspond pas à un nombre, la conversion va s'arrêter à la première erreur et le total sera multiplié par une valeur imprévue. Dans l'exemple, la valeur de `$_GET['nombre']` sera la chaîne vide qui sera convertie en 0 et le produit vaudra 0. Il faut alors recommencer toute la saisie !



Question 12.3 : Modifier le code de manière à pouvoir réinitialiser le produit des nombres en appuyant sur un autre bouton de type `submit`.

Correction :

```
1  <?php
2  //On crée le compteur s'il n'existe pas ou si on souhaite le réinitialiser
3  if(! isset($_SESSION['produit']) or isset($_GET['reinit']) )
4      $_SESSION['produit'] = 1;
5
6  //Une valeur a été soumise et on a cliqué sur le bouton Multiplier
7  if(isset($_GET['nombre']) and isset($_GET['mult'])) {
8      //Si c'est bien un nombre flottant
9      if( preg_match("#^-?\d*\.\d+$#", $_GET['nombre']) )
10         $_SESSION['produit'] *= $_GET['nombre'];
```



```

11     else
12         echo '<p> La valeur saisie n\'est pas un nombre </p>';
13     }
14     ?>
15     <form action="nombre.php">
16         <p>
17             Nombre :<input type="text" name="nombre"/>
18             <input type="submit" name="mult" value="Multiplier"/>
19         </p>
20         <p>
21             <input type="submit" name="reinit" value="Réinitialiser">
22         </p>
23     </form>
24
25     <?php echo '<p> Total = ' . $_SESSION['produit'] . '</p>'; ?>

```



Question 12.4 : À l'aide des cookies, modifier le script pour qu'à chaque fois que l'on réinitialise un produit, la valeur de ce produit (s'il est différent de 1) soit stocké dans un cookie. Afficher à la fin du script la valeur du dernier produit calculé avant réinitialisation si cette information est disponible.

Correction :

```

1  //////////////////////////////////////
2  //  AU DEBUT DU FICHIER  //
3  //////////////////////////////////////
4  //Si l'on a bien demandé une réinitialisation et que l'on a déjà effectué
5  //un produit
6  if(isset($_GET['reinit']) and isset($_SESSION['produit']))
7      and $_SESSION['produit'] != 1) {
8      //Envoi du cookie
9      setcookie('produitNb', $_SESSION['produit'],time()+3600,
10                null, null, false, true);
11      //Mise à jour de la variable $_COOKIE pour cette exécution du script
12      $_COOKIE['produitNb'] = $_SESSION['produit'];
13  }
14
15  //////////////////////////////////////
16  //  EN FIN DE FICHIER  //
17  //////////////////////////////////////
18  if(isset($_COOKIE["produitNb"]))
19      echo '<p> Dernier produit calculé = ' . $_COOKIE['produitNb'] . '</p>';
20  else
21      echo '<p> Premier calcul </p>';

```

