

Les tableaux et les dates

Table des matières



Objectifs	3
I - Les tableaux	4
1. Les tableaux indexés	4
2. Les tableaux associatifs	5
3. Les tableaux multidimensionnels	6
4. Travaux dirigés	7
II - Exercice : Test de connaissance No 1	11
III - Les dates	13
1. La fonction date()	13
2. La fonction checkdate()	14
3. Le timestamp avec les fonctions time, mktime() et strtotime()	14
4. La fonction getdate()	15
5. Travaux dirigés	17
IV - Exercice : Test de connaissance No 2	20

Objectifs

A la fin de cette leçon, l'étudiant sera capable de stocker et récupérer des informations au sein d'un tableau, puis de définir des dates

Objectifs d'apprentissage

À la fin de cette leçon, vous serez capable de :

- Créer un tableau et stocker et récupérer des informations
- Définir une date dans différents formats et l'utiliser

Les tableaux



Un tableau est une structure de donnée informatique permettant de stocker ou ranger plusieurs informations au sein d'une même variable. En PHP, nous avons :

- les tableaux numérotés ou indexés,
- les tableaux associatifs.

Les deux types de tableaux susmentionnés peuvent être *unidimensionnels* ou *multidimensionnels*. Commencerons par les tableaux unidimensionnels, puis nous verrons après comment créer un tableau multidimensionnel. Les tableaux en PHP sont de type *array*.

Liens utiles :

<https://www.pierre-giraud.com/php-mysql/cours-complet/php-variables-tableaux.php>

<https://openclassrooms.com/fr/courses/918836-concevez-votre-site-web-avec-php-et-mysql/4238931-les-tableaux>

1. Les tableaux indexés

Avec les tableaux indexés, chaque élément ou cellule du tableau est référencé par un numéro. C'est par ce numéro qu'on accède à la case pour stocker ou récupérer une information.

- Pour créer un tableau la syntaxe est : `variable = array()` ;

Exemple : \$fruits= array() ;

- Pour ajouter un élément dans un tableau, on fait : `variable[indice] = valeur ;`

Exemple :

```
$fruits[0]="orange";
```

```
$fruits[1]="mangue";
```

```
$fruits[2]="pomme";
```

- Il est possible de mettre des éléments dans un tableau pendant sa création, on fait : `variable = array(valeur1,...,valeurN)` ;

Exemple : `$fruits= array("orange", "mangue", "pomme") ;`

Après, on peut ajouter d'autres informations. *Ex*: `$fruits[3] = "banane";`

- Pour récupérer et afficher un élément, on peut procéder comme suit :
`echo variable[indice]` ; Ex : `echo $fruits[2]` ;

- Pour afficher en un seul coup les éléments du tableau, on utilise la fonction *print_r()*. Ex : `print_r($fruits)` ;
- Pour connaître le nombre d'éléments dans un tableau on utilise la fonction *sizeof()*
Exemple : `$nb = sizeof($fruits)` ;
- Pour parcourir le tableau on peut utiliser la boucle *for* ou *foreach*

```
<?php
for ($i = 0; $i < 4; $i++){
    echo $fruits[$i] ;
}
?>
```

ou bien

```
<?php
foreach ($fruits as $element){
    echo $element ;
}
?>
```

Exemple2 : parcours avec la boucle *foreach*

```
<?php
foreach ($fruits as $element){
    echo $element;
}
?>
```

La boucle affiche les éléments du tableau les un à la suite des autres.

2. Les tableaux associatifs

Les tableaux associatifs fonctionnent sur le même principe, sauf qu'au lieu de numérotter les cases, on va les étiqueter en leur donnant à chacune un nom différent. Les tableaux associatifs sont aussi appelés tableaux clés valeurs, c.-à-d. à une clé (la clé est une chaîne de caractère) on associe une valeur (la valeur peut être de n'importe quel type).

Les tableaux associatifs sont utilisés pour enregistrer des informations sur des personnes, objets ou éléments possédants des attributs ou propriétés.

- Pour créer un tableau la syntaxe est : *variable = array()* ;
Exemple : `$etudiant = array()` ;
- Pour ajouter un élément dans un tableau, on fait : *variable[cle] = valeur* ;
Exemple :
`$etudiant["nom"] = "Koffi Yao"` ;
`$etudiant["age"] = 17` ;
`$etudiant["taille"] = 1.67` ;
- Il est possible de mettre des éléments dans un tableau pendant sa création, on fait : *variable = array("cle1" =>valeur1,..., "cleN"=>valeurN)* ;
Exemple : `$etudiant = array("nom"=>"Koffi Yao", "age" =>17, "taille" =>1.67)` ;

Après, on peut ajouter d'autres informations. Ex : `$etudiant["tel"] = "45124264"` ;

- Pour récupérer et afficher un élément, on peut procéder comme suit :
`echo variable[cle] ; Ex : echo $etudiant["nom"] ;`
- Pour afficher en un seul coup les éléments du tableau, on utilise la fonction *print_r()*. Ex : `print_r($etudiant) ;`
- Pour connaître le nombre d'éléments dans un tableau on utilise la fonction *sizeof()*
Exemple : `$nb = sizeof($etudiant) ;`
- Pour parcourir le tableau on peut utiliser la boucle *foreach*
Exemple :

```
foreach ($etudiant as $cle=>valeur){
    echo $valeur;
}
```

 Ou bien

```
foreach ($etudiant as $cle=>valeur){
    echo $cle . 'vaut ' . $valeur . '<br />';
}
```

`
` est une balise permettant de créer un retour à la ligne.

3. Les tableaux multidimensionnels

Pour créer un tableau à plusieurs dimensions on procède comme précédemment, mais pour stocker et récupérer des informations nous avons plusieurs approches :

- *Approche 1 : pour les tableaux indexés seulement*
 - *Syntaxe* : `variable[indice1] [indice2]...[indiceN]`
 - *Exemple* :

```
< ?php
$matrice[0][0]=12 ;
$matrice[0][1]=8 ;
$matrice[1][0]=17 ;
$matrice[1][1]=30 ;
$matrice[2][0]=15 ;
.....
?>
```

 Pour afficher un élément on procède de la même manière :
Ex : `echo $matrice[2][0] ;`
- *Approche 2 : pour les tableaux indexés et associatifs*
 - *Syntaxe* pour tableau indexé : `variable[indice]= array(valeur1, ...,valeurN)`
`;`
 - *Syntaxe* pour tableau associatif : `variable[cle]= array(valeur1, ...,valeurN) ;`
 - *Exemple 1* : tableaux indexés

```
< ?php
$matrice[0]=array(12,8,4,25) ;
$matrice[1]=array(17,30,45,20) ;
$matrice[2]=array(15,17,48,50) ;
$matrice[3]= array('x'=>20, 'y'=>15,'xy'=>25) ;
```

```
?>
Pour afficher un élément on procède de la même manière :
Ex:echo $matrice[2][3] ;// ça affiche 50
- Exemple 2 : tableaux associatif
< ?php
$etudiant['notes']= array(12,17,9,16) ;
$etudiant['coordonnees']=          array('tel'=>'47856231',
'email'=>'toto@ya.fr');
?>
Pour afficher un élément on procède comme suit:
echo $etudiant['notes'][2] ;// ça affiche 9
echo $etudiant['coordonnees']['tel'] ;// ça affiche 47856231
```

Remarque :

PHP offre quelques fonctions très utiles pour manipuler les tableaux, on a :

- `in_array` : pour vérifier si une valeur existe dans le tableau. Sa syntaxe est : `array(valeur, tableau) ;`
Ex: `< ?php if (in_array('pomme', $fruits) { echo 'yes' ;} ?>`
- `array_key_exists` : pour vérifier si une clé existe dans le tableau ;
Sa syntaxe est : `array_key_exists('cle', $array);`
- `min` et `max` : pour avoir le maximum et le minimum des éléments d'un tableau ;
Sa syntaxe est : `$maximum = max(tableau) ; OU $minimum = min(tableau);`
- `array_sum` permet de sommer les éléments d'un tableau contenant des nombres. Ex : `$som = array_sum($tab) ;`

4. Travaux dirigés

Calcul de moyenne : tableau une dimension

Libellé :

On considère une liste de notes et leurs coefficients respectifs assignés dans deux tableaux notes et coefs. Écrire un script permettant de calculer la moyenne résultante. vous initialiserez au départ les deux tableaux par des valeurs quelconque.

Correction :

```
<?php
$notes=array(12.5, 17, 9, 15, 19, 14, 8.5);
$coefs=array(1,2,2,1,3,1,1);
$somN=0; // cette variable représente la somme des notes
for($i=0;$i<sizeof($notes);$i++){
    $somN += $notes[$i]*$coefs[$i];
}
$somC= array_sum($coefs); // on somme les coefs
```

```

$moy = $somN/$somC;

$moy= round($moy,2); // on arrondit la moyenne à l'ordre 2, cela est
facultatif

echo "La moyenne est : ".$moy;

?>

```

Calcul de moyenne : tableau deux dimensions

Libellé :

On considère une liste de notes et leurs coefficients respectifs assignés dans un tableau notes de deux colonnes, la première colonne contenant les notes et la deuxième les coefficients des notes. Écrire un script permettant de calculer la moyenne résultante. vous initialiserez au départ le tableau par des valeurs quelconque.

Correction :

```

<?php

$notes=array();

$notes[0][0]=12.5;
$notes[0][1]=1;
$notes[1][0]=17;
$notes[1][1]=2;
$notes[2][0]=9;
$notes[2][1]=2;
$notes[3][0]=15;
$notes[3][1]=1;
$notes[4][0]=19;
$notes[4][1]=3;
$notes[5][0]=14;
$notes[5][1]=1;
$notes[6][0]=8.5;
$notes[6][1]=1;

$somN=0; // cette variable représente la somme des notes
$somC=0; // représente la somme des coefs

for($i=0;$i<sizeof($notes);$i++){
    $somN += $notes[$i][0]*$notes[$i][1];
    $somC += $notes[$i][1];
}

$moy = $somN/$somC;

```



```
$moy= round($moy,2); // on arrondit la moyenne à l'ordre 2, cela est facultatif

echo "La moyenne est : ".$moy;

?>
```

NB :

On pouvait initialiser le tableau notes de la manière suivante :

```
$notes=          array(array(12.5,1),array(17,2),array(9,2),array(15,1),array
(19,3),array(14,1),array(8.5,1));
```

Calcul de moyenne : tableau associatif

Libellé :

On considère une liste de moyenne et leurs coefficients respectifs dans différentes matières assignés dans un tableau associatif moy de trois clés. Une clé matiere stocke les noms des matières, une clé moyenne stocke les moyennes des matières et une clé coef stocke les coefficients respectifs des moyennes. Écrire un script qui liste les matières, leurs moyennes et leurs coefficients dans un tableau HTML, puis affiche la moyenne la plus petite et la plus grande. Enfin, le script calcule la moyenne générale résultante et l'affiche. vous initialiserez au départ le tableau par des valeurs quelconque.

Correction :

```
<!DOCTYPE HTML>

<html lang="fr">

<head>

<meta http-equiv="content-type" content="text/html" />

<title>tp calcul moyenne</title>

<style type="text/css">
    table{
        border-collapse: collapse;
    }
    table td,table th{
        padding: 5px 15px;
        border: 1px solid rgb(64,64,64);
    }
</style>

</head>

<body>

<?php
$notes=array();
```

```

$notes["matiere"]=      array("maths","réseau","BD","sécurité","PHP","SE","
comptabilité");

$notes["moyenne"]= array(12.5, 17, 9, 15, 19, 14, 8.5);

$notes["coef"]= array(1,2,2,1,3,1,1);

echo "Le tableau des moyennes, matières et coefs est : <br/>";

echo "<table> <tr> <th>Matière</th> <th>Moyennes</th> <th>Coefs</th> <
/tr>";//Entête du tableau

//Affichage des différentes ligne du tableau en html
for($i=0;$i<sizeof($notes['matiere']);$i++){
    echo "<tr> <td>". $notes['matiere'][$i]."</td> <td>".$notes['moyenne']
[$i]."</td> <td>".$notes['coef'][$i]."</td> </tr>";
}

echo "</table>";//on ferme la balise <table>

echo "La plus petite moyenne est : ".min($notes['moyenne'])."<br/>";
echo "La plus grande moyenne est : ".max($notes['moyenne'])."<br/>";

//Calcul de la moyenne générale
$somN=0;
for($i=0;$i<sizeof($notes['moyenne']);$i++){
    $somN += $notes['moyenne'][$i]*$notes['coef'][$i];
}

$somC=array_sum($notes['coef']);
$moy = $somN/$somC;

$moy= round($moy,2); // on arrondit la moyenne à l'ordre 2
echo "La moyenne générale est : <strong>$moy</strong>";

?>

</body>

</html>

```

Exercice : Test de connaissance No 1

II

Exercice

Complétez le code suivant pour créer un tableau tabs et stocker les valeur 12 et 17, puis les afficher :

```
<?php
$tabs=      () ;
$tabs      0      =12 ;
$tabs      1      =17
echo "Les éléments du tableau sont : " $tabs      ." et " $tabs
      ;
?>
```

Exercice

Complétez le code suivant pour afficher le nombre d'éléments dans le tableau tabs :

```
<?php
$nb=      ($tabs) ;
echo      ;
?>
```

Exercice

Complétez le code suivant pour créer un tableau tabs, puis le parcourir pour afficher les éléments :

```
<?php
$tabs=      17,12,5,14.5,9,7,15      ;
      ($i=      ;$i<      ($tabs) ;$i+      )
echo $tabs[      ].'<br/>' ;
     
?>
```

Exercice

Complétez le code suivant pour enregistrer et afficher les informations concernant un étudiant :

```
<?php
```

```
$etudiant=(  
$etudiant["nom"]="Koffi";  
$etudiant["prenom"]="Jean Yves";  
$etudiant["ville"]="Abidjan";  
$etudiant["tel"]=44785213;  
$etudiant["age"]=22;  
$etudiant["ecole"]="UVCI";  
echo "Les infos de l'étudiant sont : <br/>";  
  
($cle $valeur)  
  
echo "$cle : <br/>";  
  
?>
```

Les dates



La gestion des dates est une tâche très délicate dans les langages de programmation. Pour manipuler les dates, PHP nous offre plusieurs fonctions.

1. La fonction `date()`

La fonction `date()` crée une chaîne de caractère contenant la date du jour au format défini par son paramètre, selon la syntaxe suivante : `date("format") ;`

`Format` est une suite de caractères permettant de définir le format de la date qu'on veut. Pour cela, on utilise plusieurs symboles comme suit :

<i>Caractère</i>	<i>Signification</i>
d	Représente le jour du mois (entre 01 et 31)
m	Représente un mois (entre 01 et 12)
n	Représente un mois (entre 1 et 12)
M	Représente un mois écrit en abrégé (« Jan » pour January, janvier)
Y	Représente une année sous forme de 4 chiffre(ex : 2015)
y	Affiche l'année sous forme de 2 chiffres
D	Représente le jour de la semaine (ex : Monday)
N	Représente un jour de la semaine (entre 1 et 7, Monday = 1, Sunday = 7)
w	Affiche le numéro du jour de la semaine (0 à 6)
W	Affiche le numéro de la semaine dans l'année
t	Affiche le nombre de jour dans le mois
z	Affiche le numéro du jour dans l'année

H	Représente les heures (entre 00 et 23)
h	Représente les heures (entre 00 et 12)
i	Représente les minutes (entre 00 et 59)
s	Représente les secondes (entre 00 et 59)

Exemple :

```
<?php
echo date("d-m-Y"); // affiche 27-07-2018
echo date("Y/m/d"); // affiche 2018/07/27
echo date("d-m-Y H:i:s"); // affiche 27-07-2018 20:30:50
echo 'Il est : '.date('H :i :s') ;

?>
```

2. La fonction `checkdate()`

La fonction `checkdate()` permet de vérifier si une date est au bon format. Sa syntaxe est : `checkdate(mois,jour,annee)`.

Exemple :

```
<?php
if (checkdate(2,29,2018)){
echo "La date 29-2-2018 est au bon format";
}else {
echo "le format de la date n'est pas correct";
}

?>
```

3. Le timestamp avec les fonctions `time`, `mktime()` et `strtotime()`

- La fonction `time()` permet de récupérer le timestamp UNIX actuel, c.-à-d., le nombre de seconds écoulées depuis 1970.

Exemple :

```
<?php
echo 'le nombre de seconds écoulés depuis 1970 est :';
echo time() ;

?>
```

- Pour récupérer le timestamp d'une date quelconque on utilise la fonction `mktime()`. Sa syntaxe est : `mktime(heure, minute, second, mois, jour, annee)` ;

Exemple :

```
<?php
echo 'le nombre de seconds écoulés de 1970 au 30-07-2018 à 12 :30 :00 est
:' ;

echo mktime(12,30,0,7,30,2018) ;

?>
```

- Pour connaître la date correspondant à un timestamp donné on utilise la fonction `date()`

Exemple :

```
<?php
echo 'le timestamp 2451010 correspond à la date ' . date('Y/m/d H :m :s',
2451010) ;

echo 'La date du timestamp 1251020 est ' . date('d-m-Y',1251020) ;

?>
```

- Il est possible de d'avoir le timestamp d'une date anglaise au format texte grâce à la fonction `strtotime()`. Sa syntaxe est `strtotime('date')` ;

Exemple :

```
<?php
echo 'le timestamp la date 25/07/2018 est ' . strtotime('2018/07/25') ;

?>
```

4. La fonction `getdate()`

La fonction `getdate()` retourne un tableau associatif contenant les informations de date et d'heure du timestamp lorsqu'il est fourni. Sa syntaxe est : `getdate()` ou `getdate(timestamp)`.

Nom des clés du tableau associatif retourné		
Clé	Description	Exemple de valeur retournée
"seconds"	Représentation numérique des secondes	0 à 59
"minutes"	Représentation numérique des minutes	0 à 59
"hours"	Représentation numérique des heures	0 à 23
"mday"	Représentation numérique du jour du mois courant	1 à 31
"wday"	Représentation numérique du jour de la semaine courante	0 (pour Dimanche) à 6 (pour Samedi)

"mon"	Représentation numérique du mois	1 à 12
"year"	Année, sur 4 chiffres	Exemples : 1999 ou 2018
"yday"	Représentation numérique du jour de l'année	0 à 365
"weekday"	Version texte du jour de la semaine	Sunday à Saturday
"month"	Version texte du mois, comme January ou March	January à December
0	Nombre de secondes depuis l'époque Unix, similaire à la valeur retournée par la fonction time() et utilisée par date().	Dépend du système, typiquement de -2147483648 à 2147483647.

Exemple :

```
<?php
$today = getdate();
print_r($today);
?>
```

L'exemple ci-dessus va afficher quelque chose de similaire à :

```
Array
(
    [seconds] => 40
    [minutes] => 58
    [hours] => 21
    [mday] => 17
    [wday] => 2
    [mon] => 6
    [year] => 2018
    [yday] => 167
    [weekday] => Tuesday
    [month] => June
    [0] => 1055901520
)
```

Pour afficher le jour en lettre on peut faire : `echo $today['weekday']`

Liens utiles :

<https://www.pierre-giraud.com/php-mysql/cours-complet/php-obtenir-formater-date.php>

https://fr.wikibooks.org/wiki/Programmation_PHP/Dates

<http://php.net/manual/fr/function.date.php>

5. Travaux dirigés

Jour de la semaine en Français

Libellé :

Écrire un programme qui affiche le jour de la semaine en Français

Correction :

```
<?php
$j= date('N');
switch($j){
    case 1: echo "Lundi";
    break;
    case 2: echo "Mardi";
    break;
    case 3: echo "Mercredi";
    break;
    case 4: echo "Jeudi";
    break;
    case 5: echo "Vendredi";
    break;
    case 6: echo "Samedi";
    break;
    case 7: echo "Dimanche";
    break;
    default: echo "Erreur";
}
?>
```

Le jour suivant :

Libellé :

Écrire un script qui affiche la date de demain

Correction :

```
<?php
    $t= strtotime("tomorrow");
    echo date("d-m-Y",$t);
?>
```

Ou bien

```
<?php
    $t= strtotime("+1 day");
    echo date("d-m-Y",$t);
?>
```

Pour afficher la date dans 3 jours on fait :

```
<?php
    $t= strtotime("+3 day");
    echo date("d-m-Y",$t);
?>
```

Libellé :

Écrire un script qui affiche le suivant d'une date donnée

Correction :

```
<?php

$t= strtotime("+1 day 2018-10-20"); //La date dans la fonction strtotime
doit être au format anglais

echo date("d-m-Y",$t);

?>
```

Semaine prochaine

Libellé :

Écrire un script qui affiche la date du prochain mardi

Correction :

```
<?php

    $t= strtotime("next Tuesday");

    echo date("d-m-Y",$t);

?>
```

Pour afficher par exemple le samedi qui vient après une date donnée (2018-10-21), on fait :

```
<?php

$t= strtotime("next saturday 2018-10-21");

echo date("d-m-Y",$t);
```

?>

Affichage Heure

Libellé :

Écrire un script pour afficher l'heure se trouvant dans la chaîne "17-10-2018 15:30:55"

Correction :

```
<?php
$t= strtotime("2018-10-17 15:30:55");
$ladate= getdate($t);
echo "L'heure est : ".$ladate['hours'].":".$ladate['minutes'].":".$ladate
['seconds'];
?>
```

Ou Bien

```
$t= strtotime("2018-10-17 15:30:55");
echo "L'heure est : ".date('H:i:s',$t);
?>
```

Exercice : Test de connaissance No 2

IV

Exercice

Complétez le code suivant pour afficher la date du jour sous forme de jj-mm-aaaa :

```
<?php
echo ( " - " );
?>
```

Exercice

Complétez le code suivant pour afficher l'heure du jour sous forme de h:m:s , h est compris entre 00 et 23, m les minutes entre 00 et 59 et s entre 00 et 59

```
<?php
echo ( ' : ' );
?>
```

Exercice

Complétez le code suivant pour afficher la date et l'heure du jour sous forme de j_lettre j_chiffre m_lettre annee, où j_lettre représente le jour en lettre (en anglais ex : Sun pour dimanche), j_chiffre le jour en chiffre, m_lettre le mois en lettre (en anglais ex Jan pour janvier ou january), et annee représente l'année. Ex Mon 15 Jan 2018 :

```
<?php
echo ( " " );
?>
```

Exercice

Complétez le code suivant pour afficher la date correspondant au timestamp 2547856320 au format jj-mm-aaaa

```
<?php
echo ( ' - ' , 2547856320 );
?>
```

Exercice

Complétez le code suivant pour afficher l'heure du timestamp 245789554

```
<?php
$ladate= (245789554) ;
$h= $ladate[' '];
$m= $ladate[' '];
$s= $ladate[' '];
echo "L'heure correspondant est : $h:$m:$s" ;
?>
```