

Les array

Présentation des array

Les tableaux en PHP sont des variables spéciales qui peuvent stocker plusieurs valeurs en même temps.

Dans un tableau, chaque valeur va être associée à une clef unique. Cette clef va nous permettre notamment de récupérer la valeur associée. Nous allons pouvoir définir les différentes clefs ou laisser le PHP générer automatiquement les clefs pour les différentes valeurs d'un tableau.

On va pouvoir créer trois types de tableaux différents en PHP :

- Des tableaux numérotés ou indexés (les clefs vont être des nombres) ;
- Des tableaux associatifs (nous allons définir la valeur que l'on souhaite pour chaque clef) ;
- Des tableaux multidimensionnels (tableaux qui stockent d'autres tableaux en valeur).

Nous allons bien évidemment étudier chacun de ces trois types de tableaux dans la suite de ce cours et présenter leurs spécificités.

Pour créer un tableau, on peut soit utiliser la structure de langage `array()`, soit la nouvelle syntaxe plus courte `[]`.

On va pouvoir passer autant d'argument qu'on souhaite stocker de valeurs dans notre tableau à `array()` ou dans `[]`. Les arguments vont pouvoir être soit des valeurs simples (auquel cas les clefs seront des entiers générés automatiquement), soit des paires clef => valeur.

Création d'un tableau indicé ou indexé

Les tableaux numérotés sont le type de tableaux le plus simple à créer en PHP puisque les clefs vont être générées automatiquement par le PHP.

Pour créer un tableau numéroté en PHP, il suffit en fait d'indiquer une série de valeurs et le PHP associera automatiquement une clef unique à chaque valeur, en commençant avec la clef 0 pour la première valeur, la clef 1 pour la deuxième valeur, la clef 2 pour la troisième valeur, etc.

Regardez plutôt le code ci-dessous. On crée ici deux tableaux numérotés \$prenoms et \$ages en utilisant les deux syntaxes vues précédemment.

```
1.  <!DOCTYPE html>
2.  <html>
3.      <head>
4.          <title>Cours PHP & MySQL</title>
5.          <meta charset="utf-8">
6.          <link rel="stylesheet" href="cours.css">
7.      </head>
8.
9.      <body>
10.         <h1>Titre principal</h1>
11.         <?php
12.             $prenoms = array('Mathilde', 'Pierre', 'Amandine', 'Florian');
13.             $ages = [27, 29, 21, 29];
14.         ?>
15.         <p>Un paragraphe</p>
16.     </body>
17. </html>
```

Ici, on crée deux variables tableau \$prenoms et \$ages. Notez bien qu'on va toujours stocker nos tableaux dans des variables. Notre premier tableau stocke 4 valeurs qui sont ici des chaînes de caractères. Notre deuxième tableau stocke 4 chiffres.

On peut tout à fait créer des tableaux qui vont stocker des chaînes de caractères, des nombres, des booléens, etc. Comme d'habitude, seules les chaînes de caractères doivent être entourées d'apostrophes ou de guillemets droits.

Ici, le PHP va automatiquement créer les clefs qu'on appelle également indices ou index et les associer aux différentes valeurs. La valeur Mathilde de notre premier tableau va avoir la clef 0, la valeur Pierre la clef 1, etc. De même, la clef 0 va être associée à la valeur 27 de notre deuxième tableau, la deuxième valeur (le premier 29) va être associée à la clef 1 et etc.

On va également pouvoir créer un tableau indice par indice et valeur par valeur en précisant ici les indices à associer à chaque valeur. Dans ce cas-là, il faudra utiliser la syntaxe suivante :

```
1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <title>Cours PHP & MySQL</title>
5.         <meta charset="utf-8">
6.         <link rel="stylesheet" href="cours.css">
7.     </head>
8.
9.     <body>
10.        <h1>Titre principal</h1>
11.        <?php
12.            $prenoms[0] = 'Mathilde';
13.            $prenoms[1] = 'Pierre';
14.            $prenoms[2] = 'Amandine';
15.            $prenoms[3] = 'Florian';
16.
17.            $ages[0] = 27;
18.            $ages[1] = 29;
19.            $ages[2] = 21;
20.            $ages[3] = 29;
21.        ?>
22.        <p>Un paragraphe</p>
23.    </body>
24. </html>
```

Afficher les valeur d'un tableau numéroté (indiqué)

Pour afficher les valeurs d'un tableau numéroté une à une, il suffit d'echo notre variable tableau en précisant l'indice (entre crochets) correspondant à la valeur que l'on souhaite afficher.

```
1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <title>Cours PHP & MySQL</title>
5.         <meta charset="utf-8">
6.         <link rel="stylesheet" href="cours.css">
7.     </head>
8.
9.     <body>
10.        <h1>Titre principal</h1>
11.        <?php
12.            $prenoms = ['Mathilde', 'Pierre', 'Amandine', 'Florian'];
13.
14.            echo $prenoms[0]. '<br>';
15.            echo $prenoms[2];
16.        ?>
17.        <p>Un paragraphe</p>
18.    </body>
19. </html>
```

Titre principal

Mathilde
Amandine

Un paragraphe

Pour afficher toutes les valeurs d'un tableau numéroté d'un coup, nous allons cette fois-ci devoir utiliser une boucle for.

Nous allons boucler sur les indices de notre tableau et nous allons récupérer la valeur associée à l'indice en question à chaque nouveau passage dans la boucle.

Il va donc avant tout falloir déterminer la taille de notre tableau, c'est à dire le nombre de valeurs contenues dans notre tableau. Bien évidemment, nous travaillons pour le moment avec des tableaux qu'on crée manuellement et on peut donc ici compter le nombre de valeurs à la main mais notez bien ici qu'en pratique on ne saura pas forcément à priori combien de valeurs sont stockées dans un tableau.

La façon la plus simple de déterminer la taille d'un tableau est d'utiliser la fonction count() qui va tout simplement prendre un tableau en argument et va retourner le nombre d'éléments de ce tableau.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Cours PHP & MySQL</title>
    <meta charset="utf-8">
    <meta name="viewport"
      content="width=device-width, initial-scale=1, user-scalable=no">
    <link rel="stylesheet" href="cours.css">
  </head>
  <body>
    <h1>Titre principal</h1>
    <?php
      $prenoms = ['Mathilde', 'Pierre', 'Amandine', 'Florian'];

      echo 'Le tableau $prenoms contient ' . count($prenoms) . ' valeurs';
    ?>
    <p>Un paragraphe</p>
  </body>
</html>
```

Titre principal

Le tableau \$prenoms contient 4 valeurs

Un paragraphe

Une fois qu'on a déterminé la taille de notre tableau, il va être très simple de créer notre boucle for.

```
1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <title>Cours PHP & MySQL</title>
5.         <meta charset="utf-8">
6.         <link rel="stylesheet" href="cours.css">
7.     </head>
8.
9.     <body>
10.        <h1>Titre principal</h1>
11.        <?php
12.            $prenoms = ['Mathilde', 'Pierre', 'Amandine', 'Florian'];
13.
14.            //On récupère la taille du tableau et on la stocke dans
15.            $taille
16.            $taille = count($prenoms);
17.
18.            //On peut soit parcourir le tableau et afficher les valeurs
19.            une à une
20.            for($i = 0; $i < $taille; $i++){
21.                echo $prenoms[$i]. ' , ' ;
22.            }
23.
24.            //...soit les stocker dans une autre variable et echo cette
25.            variable
26.            for($i = 0; $i < $taille; $i++){
27.                $p .= $prenoms[$i]. ' , ' ;
28.            }
29.            echo $p;
30.        ?>
31.        <p>Un paragraphe</p>
32.    </body>
33. </html>
```

Titre principal

Mathilde, Pierre, Amandine, Florian,

Mathilde, Pierre, Amandine, Florian,

Un paragraphe

Ici, on utilise la fonction `count()` pour déterminer le nombre de valeurs dans notre tableau puis on boucle sur les indices en partant de l'indice 0. Lors du premier passage dans la boucle, on va donc récupérer la valeur associée à l'indice 0 du tableau ; lors du deuxième passage, nous allons récupérer la valeur associée à l'indice 1 et etc.

Notre première boucle `for` va echo la valeur associée à la clef à chaque passage de boucle tandis que notre deuxième boucle `for` va stocker les valeurs liées à chaque clef dans la variable `$p` grâce à l'opérateur d'affectation `+=`.

Cette technique va très bien fonctionner tant que nos tableaux numérotés vont avoir des indices « naturels ». En effet, il est tout à fait possible d'attribuer les indices manuellement et de sauter certains indices pour stocker nos valeurs dans nos variables tableaux.

Dans ce cas-là, on ne pourra pas récupérer toutes les valeurs en bouclant sur les indices comme ci-dessus mais on utilisera plutôt une boucle `foreach` qui est une boucle spécialement créée pour les tableaux.

Nous allons utiliser cette syntaxe :

```
1.  <!DOCTYPE html>
2.  <html>
3.      <head>
4.          <title>Cours PHP & MySQL</title>
5.          <meta charset="utf-8">
6.          <link rel="stylesheet" href="cours.css">
7.      </head>
8.
9.      <body>
10.         <h1>Titre principal</h1>
11.         <?php
12.             $prenoms = ['Mathilde', 'Pierre', 'Amandine', 'Florian'];
13.
14.             //On rajoute une valeur à $prenoms et on lui associe la clef 8
15.             $prenoms[8] = 'Lisa';
16.
17.             $taille = count($prenoms);
18.
19.             /*On tente d'afficher les valeurs de notre tableau en
utilisant le
20.             *résultat de count() et en bouclant sur les indices*/
21.             for($i = 0; $i < $taille; $i++){
22.                 $p .= $prenoms[$i]. ' , ';
23.             }
24.             echo $p. '<br><br>';
25.
26.             //On utilise une boucle foreach ($tableau as $valeurs)
27.             foreach ($prenoms as $valeurs){
28.                 $resultat .= $valeurs. ' , ';
29.             }
30.             echo $resultat;
31.         ?>
32.         <p>Un paragraphe</p>
33.     </body>
34. </html>
```

Titre principal

Mathilde, Pierre, Amandine, Florian, ,

Mathilde, Pierre, Amandine, Florian, Lisa,

Un paragraphe

Ici, boucler sur les indices avec une boucle for ne permet pas de renvoyer toutes les valeurs du tableau tout simplement car notre boucle va s'arrêter avant d'atteindre l'indice 8. En effet, count() sert à compter le nombre d'éléments d'un tableau. Ici, notre fonction renvoie donc 5. On va donc ensuite boucler sur les indices de 0 à 4, ce qui ne nous apporte pas le résultat attendu.

La boucle foreach, au contraire, va nous permettre de parcourir des tableaux élément par élément. A chaque nouveau passage dans la boucle, la valeur d'un élément du tableau va être placée dans la variable qu'on a ici appelé \$valeurs et la boucle va aller chercher la valeur suivante jusqu'à arriver à la fin du tableau.

Les tableaux associatifs

Un tableau associatif est un tableau qui va utiliser des clefs textuelles qu'on va associer à chaque valeur.

Les tableaux associatifs vont s'avérer intéressant lorsqu'on voudra donner du sens à nos clefs, c'est-à-dire créer une association forte entre les clefs et les valeurs d'un tableau.

Imaginons par exemple qu'on souhaite stocker les âges de nos différents utilisateurs dans un tableau. Ici, plutôt que d'utiliser un tableau numéroté dans lequel il serait difficile de dire à qui appartient chaque âge, il serait judicieux d'utiliser un tableau associatif en utilisant par exemple les pseudonymes de nos membres comme clefs.

Créer un tableau associatif

Les tableaux associatifs vont être différents des tableaux numérotés au sens où nous allons devoir définir chacune des clefs : le PHP ne va pas ici pouvoir nommer automatiquement nos clefs.

Tout comme pour les tableaux numérotés, on va pouvoir créer notre tableau en une fois en utilisant la structure de langage `array()` ou la syntaxe `[]` ou le construire clef par clef et valeur par valeur.

```
1.  <!DOCTYPE html>
2.  <html>
3.      <head>
4.          <title>Cours PHP & MySQL</title>
5.          <meta charset="utf-8">
6.          <link rel="stylesheet" href="cours.css">
7.      </head>
8.
9.      <body>
10.         <h1>Titre principal</h1>
11.         <?php
12.             $ages = ['Mathilde' => 27, 'Pierre' => 29, 'Amandine' => 21];
13.
14.             $mails['Mathilde'] = 'math@gmail.com';
15.             $mails['Pierre'] = 'pierre.giraud@edhec.com';
16.             $mails['Amandine'] = 'amandine@lp.fr';
17.         ?>
18.         <p>Un paragraphe</p>
19.     </body>
20. </html>
```


On crée notre premier tableau \$ages d'un seul coup en utilisant la syntaxe []. Ici, « Mathilde », « Pierre » et « Amandine » sont les clefs ou indices du tableau et 27, 29 et 21 sont les valeurs associées. Notez bien l'utilisation du signe => qui sert à associer une clef à une valeur.

Récupérer et afficher les valeur d'un tableau associatif

On va pouvoir afficher une valeur en particulier d'un tableau associatif très simplement de la même façon que pour les tableaux numérotés :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Cours PHP & MySQL</title>
    <meta charset="utf-8">
    <meta name="viewport"
      content="width=device-width, initial-scale=1, user-scalable=no">
    <link rel="stylesheet" href="cours.css">
  </head>

  <body>
    <h1>Titre principal</h1>
    <?php
      $ages = ['Mathilde' => 27, 'Pierre' => 29, 'Amandine' => 21];

      /*Identique à
      *$ages = array('Mathilde' => 27, 'Pierre' => 29, 'Amandine' => 21);
      */

      $mails['Mathilde'] = 'math@gmail.com';
      $mails['Pierre'] = 'pierre.giraud@edhec.com';
      $mails['Amandine'] = 'amandine@lp.fr';

      echo 'Pierre a ' . $ages['Pierre']. ' ans';
    ?>
    <p>Un paragraphe</p>
  </body>
</html>
```

Titre principal

Pierre a 29 ans

Un paragraphe

Pour parcourir un tableau associatif et par exemple afficher les valeurs les unes après les autres, nous allons en revanche être obligés d'utiliser une boucle foreach qui est une boucle créée spécialement pour les tableaux.

Ici, nous allons utiliser la boucle foreach de la manière suivante :

```
1.  <!DOCTYPE html>
2.  <html>
3.      <head>
4.          <title>Cours PHP & MySQL</title>
5.          <meta charset="utf-8">
6.          <link rel="stylesheet" href="cours.css">
7.      </head>
8.
9.      <body>
10.         <h1>Titre principal</h1>
11.         <?php
12.             $ages = ['Mathilde' => 27, 'Pierre' => 29, 'Amandine' => 21];
13.
14.             /*Identique à
15.              *$ages = array('Mathilde' => 27, 'Pierre' => 29, 'Amandine'
=> 21);
16.              */
17.
18.             $mails['Mathilde'] = 'math@gmail.com';
19.             $mails['Pierre'] = 'pierre.giraud@edhec.com';
20.             $mails['Amandine'] = 'amandine@lp.fr';
21.
22.             foreach($ages as $clef => $valeur){
23.                 echo $clef. ' a ' . $valeur. ' ans<br>';
24.             }
25.         ?>
26.         <p>Un paragraphe</p>
27.     </body>
28. </html>
```

Titre principal

Mathilde a 27 ans
Pierre a 29 ans
Amandine a 21 ans

Un paragraphe

Ici, nous utilisons une syntaxe de type foreach(\$tableau as \$clef => \$valeur). Cette syntaxe nous permet de récupérer à la fois les valeurs du tableau qui vont être stockées dans la variable qu'on a ici appelée \$valeur ainsi que les clefs associées à chaque valeur.

Lors du premier passage dans la boucle, la première paire clef => valeur du tableau va être récupérée et affichée grâce à echo puis foreach va nous permettre de passer à la paire

suivante clef => valeur du tableau qu'on va afficher lors du deuxième passage dans la boucle et etc. jusqu'à la fin de notre tableau.

Les tableaux multidimensionnels

Un tableau multidimensionnel est un tableau qui va lui-même contenir d'autres tableaux en valeurs.

On appelle ainsi tableau à deux dimensions un tableau qui contient un ou plusieurs tableaux en valeurs, tableau à trois dimensions un tableau qui contient un ou plusieurs tableaux en valeurs qui contiennent eux-mêmes d'autres tableaux en valeurs et etc.

Les « sous » tableaux vont pouvoir être des tableaux numérotés ou des tableaux associatifs ou un mélange des deux.

Nous ne sommes pas limités dans le nombre de dimensions d'un tableau : le PHP sait tout à fait travailler avec des tableaux à 2, 3, 4, 5... dimensions. Cependant, il est généralement déconseillé de créer trop de dimensions de tableaux tout simplement car cela rend le code très vite très peu lisible et très peu compréhensible pour nous autres développeurs.

Vous pouvez déjà noter que le nombre de dimensions d'un tableau va indiquer le nombre d'indices nécessaires pour accéder à une valeur du tableau (nous allons illustrer cela par la suite).

Créer un tableau multidimensionnel

Un tableau multidimensionnel est un tableau dont les valeurs peuvent elles-mêmes être des tableaux qui vont à nouveau pouvoir contenir d'autres tableaux et etc.

Commençons déjà par créer des tableaux à deux dimensions. Vous pourrez ensuite créer des tableaux à 3, 4, 5... dimensions en suivant le même modèle.

```

1.  <!DOCTYPE html>
2.  <html>
3.      <head>
4.          <title>Cours PHP & MySQL</title>
5.          <meta charset="utf-8">
6.          <link rel="stylesheet" href="cours.css">
7.      </head>
8.
9.      <body>
10.         <h1>Titre principal</h1>
11.         <?php
12.             /*Tableau multidimensionnel numéroté stockant
13.              *des tableaux numérotés*/
14.             $suite = [
15.                 [1, 2, 4, 8, 16],
16.                 [1, 3, 9, 27, 81]
17.             ];
18.
19.             /*Tableau multidimensionnel numéroté stockant
20.              *des tableaux associatifs et une valeur simple*/
21.             $utilisateurs = [
22.                 ['nom' => 'Mathilde', 'mail' => 'math@gmail.com'],
23.                 ['nom' => 'Pierre', 'mail' => 'pierre.giraud@edhec.com'],
24.                 ['nom' => 'Amandine', 'mail' => 'amandine@lp.fr'],
25.                 'Florian'
26.             ];
27.
28.             /*Tableau multidimensionnel associatif stockant
29.              *des tableaux associatifs*/
30.             $produits = [
31.                 'Livre' => ['poids' => 200, 'quantite' => 10, 'prix' =>
15],
32.                 'Stickers' => ['poids' => 10, 'quantite' => 100, 'prix' =>
1.5]
33.             ]
34.         ?>
35.         <p>Un paragraphe</p>
36.     </body>
37. </html>

```

Ici, on a créé trois tableaux `$suite`, `$utilisateurs` et `$produits` à deux dimensions.

Notre premier tableau, `$suite`, est un tableau multidimensionnel numéroté qui contient deux valeurs qui vont elles-mêmes être des tableaux numérotés : les valeurs `[1, 2, 4, 8, 16]` et `[1, 3, 9, 27, 81]`.

Notre deuxième tableau est à nouveau un tableau multidimensionnel numéroté qui contient cette fois-ci 4 valeurs : trois tableaux associatifs et la valeur « florian ». En effet, chaque valeur d'un tableau multidimensionnel ne doit pas forcément être elle-même un tableau : il suffit au contraire qu'une valeur d'un tableau soit elle-même un tableau pour que le tableau de départ soit multidimensionnel.

Finalement, notre dernier tableau est un tableau multidimensionnel associatif qui stocke deux valeurs qui sont elles-mêmes des tableaux associatifs. Ici, les clefs de notre tableau multidimensionnel sont `Livre` et `Stickers` et les valeurs associées sont les tableaux `['poids' => 200, 'quantite' => 10, 'prix' => 15]` et `['poids' => 10, 'quantite' => 100, 'prix' => 1.5]`.

Récupérer ou afficher une valeur en particulier d'un tableau multidimensionnel

Pour récupérer une valeur en particulier dans un tableau numéroté ou associatif à une dimension, il suffisait simplement d'indiquer la clef associée à la valeur en question.

Dans le cas d'un tableau multidimensionnel qui ne contient que des tableaux en valeur, nous allons donc accéder aux différents sous tableaux si on ne précise qu'une seule clef.

Regardez plutôt l'exemple suivant :

```

1.  <!DOCTYPE html>
2.  <html>
3.      <head>
4.          <title>Cours PHP & MySQL</title>
5.          <meta charset="utf-8">
6.          <link rel="stylesheet" href="cours.css">
7.      </head>
8.
9.      <body>
10.         <h1>Titre principal</h1>
11.         <?php
12.             /*Tableau multidimensionnel numéroté stockant
13.              *des tableaux numérotés*/
14.             $suite = [
15.                 [1, 2, 4, 8, 16],
16.                 [1, 3, 9, 27, 81]
17.             ];
18.
19.             /*Tableau multidimensionnel numéroté stockant
20.              *des tableaux associatifs et une valeur simple*/
21.             $utilisateurs = [
22.                 ['nom' => 'Mathilde', 'mail' => 'math@gmail.com'],
23.                 ['nom' => 'Pierre', 'mail' => 'pierre.giraud@edhec.com'],
24.                 ['nom' => 'Amandine', 'mail' => 'amandine@lp.fr'],
25.                 'Florian'
26.             ];
27.
28.             /*Tableau multidimensionnel associatif stockant
29.              *des tableaux associatifs*/
30.             $produits = [
31.                 'Livre' => ['poids' => 200, 'quantite' => 10, 'prix' =>
32.                     15],
33.                 'Stickers' => ['poids' => 10, 'quantite' => 100, 'prix' =>
34.                     1.5]
35.             ];
36.
37.             // $sous_suite = [1, 2, 4, 8, 16]
38.             $sous_suite = $suite[0];
39.             echo $sous_suite[0]. ' <br> '. $sous_suite[2]. ' <br> ';
40.
41.             // $sous_util = ['nom' => 'Amandine', 'mail' =>
42.                 'amandine@lp.fr']
43.             $sous_util = $utilisateurs[2];
44.             echo $sous_util['nom']. ' <br> ';
45.
46.             // $sous_produits = ['poids' => 200, 'quantite' => 10, 'prix'
47.                 => 15]
48.             $sous_produits = $produits['Livre'];
49.             echo $sous_produits['prix'];
50.         ?>
51.         <p>Un paragraphe</p>
52.     </body>
53. </html>

```

Titre principal

1
4
Amandine
15

Un paragraphe

Ici, on procède en deux étapes à chaque fois pour bien comprendre ce qu'il se passe. Notre but est d'afficher certaines valeurs finales de nos tableaux multidimensionnels à deux dimensions.

A chaque fois, on commence par récupérer l'une des valeurs de nos tableaux multidimensionnels qui sont elles-mêmes des tableaux.

On récupère la valeur liée à l'indice 0 de notre tableau \$suite c'est-à-dire le tableau [1, 2, 4, 8, 16] qu'on place dans une variable \$sous_suite qui devient de fait une variable tableau. On affiche ensuite les valeurs liées aux clefs 0 et 2 de notre tableau sous-suite, c'est-à-dire 1 et 4.

On effectue le même type d'opérations avec nos deux autres tableaux multidimensionnels, en faisant bien attention à préciser les bonnes clefs textuelles lorsque nos tableaux et / ou sous tableaux sont des tableaux associatifs.

La chose à retenir ici est qu'il nous faut donc deux indices pour récupérer l'une des valeurs finales d'un tableau à deux dimensions qui ne contient que des tableaux en valeurs : un premier indice qui va nous permettre d'accéder à une valeur-tableau de notre tableau multidimensionnel et un deuxième indice qui va nous permettre d'accéder à une valeur effective dans notre valeur-tableau.

Nous allons alors pouvoir abréger l'écriture de la façon suivante :


```

1.  <!DOCTYPE html>
2.  <html>
3.      <head>
4.          <title>Cours PHP & MySQL</title>
5.          <meta charset="utf-8">
6.          <link rel="stylesheet" href="cours.css">
7.      </head>
8.
9.      <body>
10.         <h1>Titre principal</h1>
11.         <?php
12.             /*Tableau multidimensionnel numéroté stockant
13.              *des tableaux numérotés*/
14.             $suite = [
15.                 [1, 2, 4, 8, 16],
16.                 [1, 3, 9, 27, 81]
17.             ];
18.
19.             /*Tableau multidimensionnel numéroté stockant
20.              *des tableaux associatifs et une valeur simple*/
21.             $utilisateurs = [
22.                 ['nom' => 'Mathilde', 'mail' => 'math@gmail.com'],
23.                 ['nom' => 'Pierre', 'mail' => 'pierre.giraud@edhec.com'],
24.                 ['nom' => 'Amandine', 'mail' => 'amandine@lp.fr'],
25.                 'Florian'
26.             ];
27.
28.             /*Tableau multidimensionnel associatif stockant
29.              *des tableaux associatifs*/
30.             $produits = [
31.                 'Livre' => ['poids' => 200, 'quantite' => 10, 'prix' =>
32.                     15],
33.                 'Stickers' => ['poids' => 10, 'quantite' => 100, 'prix' =>
34.                     1.5]
35.             ];
36.             echo $suite[0][0]. ' <br>'. $suite[0][2]. ' <br>';
37.             echo $utilisateurs[2]['nom']. ' <br>';
38.             //Affichage d'une valeur simple contenue directement dans
39.             $utilisateurs
40.             echo $utilisateurs[3]. ' <br>';
41.             echo $produits['Livre']['prix'];
42.             ?>
43.         <p>Un paragraphe</p>
44.     </body>
45. </html>

```

Titre principal

1
4
Amandine
Florian
15

Un paragraphe

Pour accéder aux valeurs finales d'un tableau à 2 dimensions, nous allons devoir préciser deux indices : le premier indice permet d'accéder à une valeur (qui est un tableau) du tableau multidimensionnel et le deuxième indice sert à accéder à une valeur en particulier dans ce sous tableau.

Nous allons suivre exactement le même schéma pour les tableaux à 3, 4, 5... dimensions en précisant autant de clefs que notre tableau possède de dimensions.

Notez toutefois ici que dans le cas où notre tableau multidimensionnel contient à la fois des tableaux et des valeurs simples, alors on accèdera aux valeurs simples de manière « classique », c'est-à-dire en précisant seulement le nombre d'indices nous permettant d'accéder à la valeur en question. Il est cependant très rare d'avoir des tableaux multidimensionnels composés de valeurs-tableaux et de valeurs simples.

Parcourir et afficher les valeur d'un tableau multidimensionnel

Pour parcourir toutes les valeurs d'un tableau multidimensionnel (et éventuellement les afficher ou effectuer d'autres opérations dessus), la meilleure manière de faire va être d'utiliser plusieurs boucles foreach imbriquées.

On va ici utiliser autant de boucles foreach qu'on a de dimensions dans le tableau qu'on souhaite parcourir. La première boucle foreach va nous permettre de parcourir les valeurs de notre tableau multidimensionnel de base, puis la deuxième boucle foreach va nous permettre de parcourir les valeurs des tableaux contenus directement dans le tableau multidimensionnel et etc.

Essayons d'afficher toutes les valeurs de nos tableaux précédents (note : j'ai enlevé la valeur simple « Florian » de mon tableau utilisateur car elle aurait été complexe à traiter ici).

```

1. <!DOCTYPE html>
2. <html>
3.   <head>
4.     <title>Cours PHP & MySQL</title>
5.     <meta charset="utf-8">
6.     <link rel="stylesheet" href="cours.css">
7.   </head>
8.
9.   <body>
10.    <h1>Titre principal</h1>
11.    <?php
12.      $suite = [
13.        [1, 2, 4, 8, 16],
14.        [1, 3, 9, 27, 81]
15.      ];
16.
17.      $utilisateurs = [
18.        ['nom' => 'Mathilde', 'mail' => 'math@gmail.com'],
19.        ['nom' => 'Pierre', 'mail' => 'pierre.giraud@edhec.com'],
20.        ['nom' => 'Amandine', 'mail' => 'amandine@lp.fr'],
21.      ];
22.
23.      $produits = [
24.        'Livre' => ['poids' => 200, 'quantite' => 10, 'prix' =>
25.          15],
26.        'Stickers' => ['poids' => 10, 'quantite' => 100, 'prix' =>
27.          1.5]
28.      ];
29.      foreach ($suite as $suiteNb => $n){
30.        echo 'Suite ' . ($suiteNb + 1) . ' : ' ;
31.        foreach ($n as $ni => $nn){
32.          echo $nn. ' ' ;
33.        }
34.        echo '<br><br>';
35.      }
36.      foreach ($utilisateurs as $nb => $infos){
37.        echo 'Utilisateur n°' . ($nb + 1) . ' :<br>';
38.        foreach ($infos as $c => $v){
39.          echo $c. ' : ' . $v. '<br>';
40.        }
41.        echo '<br>';
42.      }
43.      foreach ($produits as $clef => $produit){
44.        echo 'Produit : ' . $clef. '<br>';
45.        foreach ($produit as $caracteristique => $valeur){
46.          echo $caracteristique. ' : ' . $valeur. '<br>';
47.        }
48.        echo '<br>';
49.      }
50.    ?>
51.    <p>Un paragraphe</p>
52.  </body>
53. </html>

```

Titre principal

Suite 1 : 1, 2, 4, 8, 16,

Suite 2 : 1, 3, 9, 27, 81,

Utilisateur n°1 :

nom : Mathilde

mail : math@gmail.com

Utilisateur n°2 :

nom : Pierre

mail : pierre.giraud@edhec.com

Utilisateur n°3 :

nom : Amandine

mail : amandine@lp.fr

Produit : Livre

poids : 200

quantite : 10

prix : 15

Produit : Stickers

poids : 10

quantite : 100

prix : 1.5

Un paragraphe

Ici, nos trois tableaux sont trois tableaux à deux dimensions. Nous allons donc utiliser deux boucles foreach à chaque fois. La première boucle foreach notre premier tableau \$suite va nous servir à accéder aux éléments de ce tableau multidimensionnel, c'est-à-dire aux deux clefs numérotées et aux deux valeurs qui sont des tableaux. On echo déjà à partir de cette première boucle le numéro de la suite qui va être affichée en rajoutant 1 à la valeur de son index (puisque les index numérotés commencent à 0).

Lors du premier passage dans cette première boucle foreach, on va donc accéder à notre première suite et on va echo « Suite 1 : » et rentrer dans notre deuxième boucle foreach. Cette deuxième boucle foreach va parcourir le sous tableau [1, 2, 4, 8, 16] et echo les différentes valeurs du tableau à chaque fois.

Dès qu'on arrive à la fin de ce premier sous tableau, on retourne dans notre première boucle foreach pour un deuxième passage et ce sont cette fois-ci les valeurs de notre deuxième sous tableau qui vont être affichées.

La chose à bien comprendre dans ce code est que notre boucle foreach interne ou imbriquée va renvoyer toutes les valeurs d'un sous tableau puis on va ensuite retourner dans notre première boucle pour effectuer un autre passage.

Afficher rapidement la structure d'un tableau multidimensionnel

Parfois, on voudra simplement afficher la structure d'un tableau PHP sans mise en forme pour vérifier ce qu'il contient ou pour des questions de débogage.

Le PHP nous fournit deux possibilités de faire cela : on va pouvoir soit utiliser la fonction `print_r()`, soit la fonction `var_dump()` que nous connaissons déjà pour afficher n'importe quel type de tableaux (numérotés, associatifs ou multidimensionnels).

Notez que `var_dump()` va nous fournir davantage d'informations que `print_r()`.

On va généralement utiliser cette fonction avec l'élément HTML `pre` pour avoir un meilleur affichage de la structure du tableau qu'on souhaite afficher (je vous rappelle que `pre` va permettre de conserver la mise en forme de notre code).

```
1.  <!DOCTYPE html>
2.  <html>
3.      <head>
4.          <title>Cours PHP & MySQL</title>
5.          <meta charset="utf-8">
6.          <link rel="stylesheet" href="cours.css">
7.      </head>
8.
9.      <body>
10.         <h1>Titre principal</h1>
11.         <?php
12.             $suite = [
13.                 [1, 2, 4, 8, 16],
14.                 [1, 3, 9, 27, 81]
15.             ];
16.
17.             $utilisateurs = [
18.                 ['nom' => 'Mathilde', 'mail' => 'math@gmail.com'],
19.                 ['nom' => 'Pierre', 'mail' => 'pierre.giraud@edhec.com'],
20.                 ['nom' => 'Amandine', 'mail' => 'amandine@lp.fr'],
21.             ];
22.
23.             $produits = [
24.                 'Livres' => ['poids' => 200, 'quantite' => 10, 'prix' =>
25.                 15],
26.                 'Stickers' => ['poids' => 10, 'quantite' => 100, 'prix' =>
27.                 1.5]
28.             ];
29.
30.             echo '<pre>';
31.             print_r($produits);
32.             var_dump($produits);
33.             echo '</pre>';
34.         ?>
35.         <p>Un paragraphe</p>
36.     </body>
37. </html>
```

Titre principal

```
Array
(
    [Livre] => Array
        (
            [poids] => 200
            [quantite] => 10
            [prix] => 15
        )

    [Stickers] => Array
        (
            [poids] => 10
            [quantite] => 100
            [prix] => 1.5
        )
)
array(2) {
    ["Livre"]=>
    array(3) {
        ["poids"]=>
        int(200)
        ["quantite"]=>
        int(10)
        ["prix"]=>
        int(15)
    }
    ["Stickers"]=>
    array(3) {
        ["poids"]=>
        int(10)
        ["quantite"]=>
        int(100)
        ["prix"]=>
        float(1.5)
    }
}
```

Un paragraphe