

LE PHP

```
<?PHP ECHO " COUR 3 "; ?>
```

Ferrandez Benjamin

C'EST QUOI UN TABLEAU EN PHP

Un tableau en PHP est une structure de données très pratique, qui permet de stocker plusieurs valeurs sous une seule variable. Comme une liste où chaque élément a une position spécifique (appelée indice / clé) et une valeur associée.

Il existe 3 types de tableaux :

- Tableau **indexé** : chaque élément est associé à un numéro commençant par **0**.
- Tableau **associatif** : chaque élément est associé à une clé nommée.
- Tableau **multidimensionnel** : un tableau qui contient d'autres tableaux.

COMMENT CRÉER UN TABLEAU

Un tableau se remarque par ses crochets : []. Il peut se construire de différentes façon :

- Avec les crochets [] (méthode moderne).

```
$fruits = ["Pomme", "Banane", "Cerise"];
```

- Avec la fonction array() (ancienne syntaxe).

```
$fruits = array("Pomme", "Banane", "Cerise");
```

Ces deux façon de faire créent le même tableau indexé, où chaque élément est associé à un numéro à partir de **0**.

LES TABLEAUX INDEXÉ

Chaque élément du tableau a une position (un **indice**) qui commence par 0.

```
3 // Tableau indexé
4
5 $fruits = ["Pomme", "Banane", "Cerise"];
6 echo $fruits[0]; // Resultat = "Pomme";
7 echo $fruits[2]; // Resultat = "Cerise";
8
9 // -----
```

LES TABLEAUX ASSOCIATIF

Au lieu d'utiliser des indices numériques, on utilise des clés nommées.

```
11 //Tableau associatif
12
13 ∨ $personne = [
14     "nom" => "Dupont",
15     "prenom" => "Maxime",
16     "age" => 36
17 ];
18 echo $personne["nom"]; // Resultat = "Dupont"
19 echo $personne["age"]; // Resultat = 36
20
21 // -----
```

LES TABLEAUX MULTIDIMENSIONNEL

Ce sont des tableaux qui contiennent d'autres tableaux.

```
23 //Tableau multidimensionnel
24
25 $etudiants = [
26     ["nom" => "Alice", "âge" => 25],
27     ["nom" => "Bob", "âge" => 22]
28 ];
29
30 echo $etudiants[0]["nom"]; // Resultat : "Alice"
31 echo $etudiants[1]["age"]; // Resultat : 22
32
33 // -----
```

AJOUTER/MODIFIER ÉLÉMENT DANS UN TABLEAU

Pour ajouter un élément dans un tableau indexé :

```
9  $fruits[] = "Mangue"; // ajoute "Mangue" à la fin du tableau
```

Pour ajouter un élément dans un tableau associatif :

```
23 $personne["pays"] = "France"; // Ajoute une clé "pays"
```

Modifier une valeur existante :

```
$personne["age"] = 26
```

PARCOURIR UN TABLEAU

Pourquoi parcourir un tableau ?

- Parcourir un tableau permet d'accéder à des éléments sans connaître à l'avance sa taille ou son contenu. Cela est particulièrement utile lorsque l'on travaille avec des listes de données

Il existe plusieurs façon de parcourir un tableau, via ce qu'on appelle : des boucles.

- La boucle **For**
- La boucle **Foreach**
- La boucle **While**
- La fonction **array_keys** (particulier).

LA BOUCLE FOR

La boucle **For** est une structure de contrôle qui permet d'exécuter un bloc de code un certain nombre de fois. Elle est utile pour parcourir les tableaux indexés où l'on accède aux éléments grâce à leur indices.

Exemple (j'ai ajouté un saut de ligne qui n'est pas nécessaire, c'est pour l'affichage ici de l'exemple) :

Fonctionnement : La variable `$i` s'incrémente de 1 après chaque itération grâce à `$i++`. Chaque itération va donc afficher le fruit correspondant à la position du tableau selon la valeur de `$i`.

```
71 // Boucle for :  
72 for($i=0; $i<count($fruits);$i++)  
73 {  
74     echo $fruits[$i] . "<br>";  
75 }  
76
```



BOUCLE FOR :

Pomme
Banane
Cerise
Mangue

LA BOUCLE FOREACH

La boucle **Foreach** est spécialement conçue pour parcourir facilement les tableaux en PHP. Elle permet de récupérer directement les valeurs d'un tableau sans avoir à gérer les indices manuellement (sans variable `$i` pour aller chercher la position par exemple)

Exemple (j'ai ajouté un saut de ligne qui n'est pas nécessaire, c'est pour l'affichage ici de l'exemple) :

Fonctionnement : A chaque tour de boucle, `$fruit` prend donc la valeur suivante du tableau. La boucle continue jusqu'à ce que tous les éléments aient été parcourus.

```
84 foreach($fruits as $fruit)
85 {
86     echo $fruit . "<br>";
87 }
```



BOUCLE FOREACH :

Pomme
Banane
Cerise
Mangue

LA BOUCLE WHILE

La boucle **While** (Tant que) permet d'exécuter un bloc de code tant qu'une condition est vraie.

Exemple (j'ai ajouté un saut de ligne qui n'est pas nécessaire, c'est pour l'affichage ici de l'exemple) :

Fonctionnement : à chaque tour de boucle, la condition est vérifiée. Si elle est vraie, le code à l'intérieur de la boucle s'exécute, sinon nous sortons de la boucle, et donc elle s'arrête.

```
92  $i=0;  
93  while ($i < count($fruits)) {  
94      echo $fruits[$i] . "<br>";  
95      $i++;  
96  }  
97
```



BOUCLE WHILE :

Pomme
Banane
Cerise
Mangue