

Apprendre les tableaux PHP

Tableaux d'ajout PHP

En PHP, les éléments peuvent être ajoutés à la fin d'un tableau en attachant des crochets ([]) à la fin du nom du tableau, puis en tapant l'opérateur d'affectation (=), puis enfin en tapant l'élément à ajouter à le tableau.

```
$string_array = ["first element", "second element"];  
$string_array[] = "third element";  
// $string_array is now: ["first element", "second element", "third element"]
```

Tableaux ordonnés PHP

En PHP, un tableau ordonné est une structure de données représentant une liste de données ordonnées et stockées. L'emplacement d'un élément dans le tableau est appelé son index. Les éléments d'un tableau ordonné sont classés par ordre numérique croissant en commençant par zéro.

```
$array_of_integers = array(3, 2, 1);  
$array_of_strings = ["one", "2", "twenty"];  
$array_of_mixed_content = ["two", 0, 1.0];  
echo $array_of_integers[0]; // 3  
echo $array_of_strings[2]; // "twenty"  
echo $array_of_mixed_content[1] // 0
```

Fonction de tableau PHP

En PHP, un tableau ordonné peut être construit avec la fonction PHP intégrée : `array()`. La fonction `array()` renvoie un tableau. Chacun des arguments avec lesquels la fonction a été invoquée devient un élément du tableau (dans l'ordre dans lequel ils ont été passés). Transmettez les arguments en les incluant, séparés par des virgules, entre parenthèses.

```
$array_of_strings = array("Ab", "C#", "D", "Fm");  
$mixed_content = array("Shelby", 5, "GTO", 1964.5);
```

Tableaux imbriqués PHP

En PHP, les tableaux imbriqués sont des tableaux qui contiennent d'autres tableaux en tant qu'éléments. Les opérations chaînées peuvent être utilisées pour accéder et/ou modifier des éléments dans un tableau imbriqué.

```
$array_of_arrays = ["a", "Not that lucky", 2, ["smoke detectors", "d", ["boring street in", "U.S."], "nothing happens"], "."];  
echo $array_of_arrays[3][2][1] //Returns  
"U.S."  
$array_of_arrays[3][2][1] = "America"; //  
"U.S." changed to "America"  
echo $array_of_arrays[3][2][1] //Returns  
"America"
```

Fonction PHP print_r

La `print_r()` fonction intégrée génère des tableaux dans un format lisible par l'homme.

Pour utiliser la fonction, placez un tableau ou une variable avec un tableau comme valeur entre les parenthèses.

```
$my_array = array(25, "or", 6, "to", 4);  
print_r($my_array);  
/*  
Array  
(  
    [0] => 25  
    [1] => or  
    [2] => 6  
    [3] => to  
    [4] => 4  
)  
*/
```

Fonction PHP array_push

La fonction intégrée `array_push()` de PHP prend un tableau comme premier argument. Les arguments qui suivent sont des éléments à ajouter au tableau.

La fonction ajoute chacun des éléments à la fin du tableau et renvoie le nouveau nombre d'éléments dans le tableau.

Pour utiliser la fonction, placez un tableau ou une variable avec un tableau comme valeur entre parenthèses.

```
$some_numbers = ["1", 2, 3.0];  
echo array_push($some_numbers, "four");  
//Returns: 4  
// $some_numbers is now: ["1", 2, 3.0,  
"four"]
```

Fonction de comptage PHP

La fonction PHP `count()` intégrée prend un tableau comme argument et renvoie le nombre d'éléments dans ce tableau.

Pour utiliser la fonction, placez un tableau ou une variable avec un tableau comme valeur entre parenthèses.

```
$devices = array("watch", "phone",  
"tablet", "laptop");  
$number_of_devices = count($devices);  
echo $number_of_devices; //Returns: 4
```

Tableaux de crochets PHP

En PHP, les tableaux peuvent également être construits en utilisant la syntaxe de tableau court. Pour utiliser cette syntaxe, placez simplement les éléments séparés par des virgules dans un ensemble de crochets.

Fonction PHP `array_pop`

La fonction intégrée PHP `array_pop()` prend un tableau comme argument. Il supprime définitivement le dernier élément d'un tableau et renvoie l'élément supprimé.

Pour utiliser la fonction, placez un tableau ou une variable avec un tableau comme valeur entre parenthèses.

```
$one_to_three = [1, 2, 3];  
$three_to_five = ["3", "four", 5.0];
```

```
$some_numbers = ["1", 2, "three"];  
echo array_pop($some_numbers); // "three"  
// $some_numbers is now: ["1", 2]
```

Fonction de décalage PHP

La fonction intégrée PHP `array_unshift()` prend un tableau comme premier argument. Les arguments qui suivent sont des éléments à ajouter au tableau.

La fonction ajoute chacun des éléments au début du tableau et renvoie le nouveau nombre d'éléments dans le tableau.

Pour utiliser la fonction, placez un tableau ou une variable avec un tableau comme valeur entre les parenthèses.

```
$joke = [7, "ate", "nine!"];  
$scared = array_unshift($joke, "Why is  
6", "afraid of 7?", "Because");  
echo $scared; // 6  
// $joke is now: ["Why is 6", "afraid of  
7?", "Because", 7, "ate", "nine!"]
```

Fonction d'implosion PHP

La `implode()` fonction intégrée crée une chaîne à partir d'un tableau. La fonction prend deux arguments : une chaîne à placer entre chaque élément d'un tableau et le tableau à joindre.

La fonction renvoie une seule chaîne avec l'argument de chaîne, appelé `$glue`, inséré entre chaque élément du tableau, appelé `$pieces`.

```
$pieces = [1, "2", "three", 4.0];  
$glue = " and a ";  
echo implode($glue, $pieces); // 1 and  
a 2 and a three and a 4
```

//Note - the arrays do not have to always be labelled "\$pieces" & "\$glue"

PHP Accéder aux valeurs du tableau

En PHP, les éléments individuels d'un tableau sont accessibles à l'aide du nom de la variable du tableau et de l'index d'emplacement entouré de crochets `[]`.

N'oubliez pas qu'en PHP, l'index de localisation commence à zéro.

```
$suit_up = ["top hat", "&", "tails"];  
echo $suit_up[0]; // top hat
```

```
$index_value = 2;  
$odd_choices = [99, "red", "balloons"];  
echo $odd_choices[$index_value]; //  
balloons
```

La fonction intégrée PHP `array_shift()` prend un tableau comme argument, supprime définitivement le premier élément du tableau et renvoie l'élément supprimé. Tous les éléments du tableau seront décalés vers un index un plus petit que leur index précédent. Pour utiliser la fonction, placez un tableau ou une variable avec un tableau comme valeur entre les parenthèses.

Réaffecter l'élément de tableau

En PHP, les éléments individuels d'un tableau peuvent facilement être réaffectés. Pour ce faire, nous accédons à l'emplacement d'un élément dans le tableau à l'aide de crochets (`[]`), puis affectons une nouvelle valeur à l'aide de l'opérateur d'affectation (`=`).

Accès et ajout d'éléments

En PHP, nous pouvons accéder à la valeur vers laquelle pointe une clé donnée en utilisant des crochets (`[]`) et la clé. Pour ajouter de nouveaux éléments à un *tableau associatif*, nous pouvons utiliser l'opérateur d'affectation (`=`). Nous pouvons également modifier les éléments existants en utilisant la même syntaxe qui ajoute de nouveaux éléments de tableau.

Attribuer par valeur ou par référence

Les tableaux PHP peuvent être assignés ou passés par valeur ou par référence. Les tableaux PHP qui sont passés par référence utilisent le sigil de référence (`&`). Voici quelques-unes des différences importantes entre les deux :

Passé par valeur : cela crée deux variables qui contiennent des copies de la même valeur mais restent des entités indépendantes.

Passé par référence : cela crée deux noms de variables (alias) qui pointent vers le même espace en mémoire. Ils ne peuvent pas être modifiés séparément !

```
$some_numbers = ["1", 2, "three"];
echo array_shift($some_numbers); //
Returns "1"
// $some_numbers is now: [2, "three"]
```

```
$num_array = [1, 2, 3];
```

```
$num_array[1] = 4; // Reassign value of
second element
```

```
echo $num_array[1]; // Prints: 4
```

```
$nums = ["2" => 2, "4" => 3, "6" => 6];
```

```
echo $nums["2"]; // Accesses value and
Prints: 2
```

```
$nums["8"] = 8; // Adds new value 8
echo $nums["8"]; // Prints: 8
```

```
$nums["4"] = 4; // Changes value of key
"4"
echo $nums["4"]; // Prints: 4
```

```
$favorites = ["food"=>"pizza",
"person"=>"myself", "dog"=>"Tadpole"];
$copy = $favorites;
$alias =& $favorites;
$favorites["food"] = "NEW!";
```

```
echo $favorites["food"]; // Prints: NEW!
echo $copy["food"]; // Prints: pizza
echo $alias["food"]; // Prints: NEW!
```

Tableau associatif PHP

En PHP, les *tableaux associatifs* sont des structures semblables à des cartes, où les clés sont associées à des valeurs. Lorsque nous avons besoin d'accéder à une valeur spécifique, nous pouvons utiliser la clé associée pour la trouver.

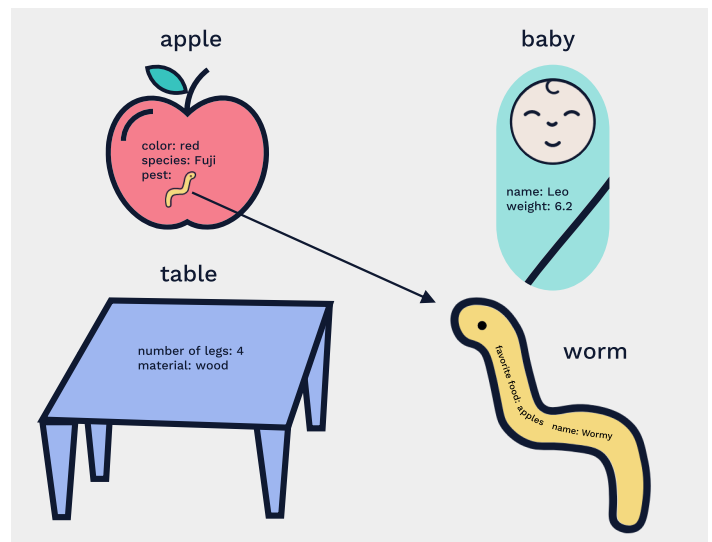
Dans un tableau ordonné PHP, les emplacements d'index sont les clés. Cependant, le type de tableau PHP nous permet également d'attribuer des clés significatives aux valeurs. Ces structures de données sont appelées *tableaux associatifs*.

Par exemple, dans le tableau associatif suivant `table`, la clé `"num_legs"` pointe vers la valeur `4` et la clé `"material"` pointe vers la valeur `"wood"`:

```
$table = ["num_legs" => 4,  
"material" => "wood"];
```

Joindre des tableaux en PHP

En PHP, l'opérateur union (`+`) prend deux opérandes de tableau et renvoie un nouveau tableau avec toutes les clés uniques du second tableau ajoutées au premier tableau.



```
$my_array = ["panda" => "very cute",  
"lizard" => "cute", "cockroach" => "not  
very cute"];
```

```
$more_rankings = ["capybara" => "cutest",  
"lizard" => "not cute", "dog" => "max  
cuteness"];
```

```
$animal_rankings = $my_array  
+ $more_rankings;
```

```
/*  
$animal_rankings will have each of the  
key=>value pairs from $my_array and the  
key=>value pairs from $more_rankings.
```

```
$my_array will retain the value of "cute"  
for $animal_rankings["lizard"] since  
"lizard is not a unique key"  
*/
```

Une `key=>value` paire dans un tableau associatif PHP peut être entièrement supprimée à l'aide de la `unset()` fonction PHP.

Si la clé utilisée n'existe pas dans le tableau, rien ne se passe.

```
$nums = ["one" => 1, "two" => 2];
```

```
unset($nums["one"]); // Removes the "one"
=> 1 pair.
```

```
echo implode(", ", $nums); // Prints: 2
```

Clés numériques dans les tableaux associatifs

Les tableaux associatifs peuvent utiliser des entiers ou des chaînes comme clés.

En PHP, les tableaux ordonnés ne sont que des tableaux dans lesquels des clés entières ont été automatiquement attribuées aux valeurs. Par exemple, le premier élément est associé à `key 0`, le second à `1`, etc. Même si les tableaux ordonnés ont la même structure que les tableaux associatifs, il est recommandé de traiter les deux séparément.

Lors de l'ajout d'un élément à un tableau sans spécifier de clé, PHP l'associe à la clé "next". Si aucune clé entière n'a été utilisée, il l'associera à la clé `0`, sinon il l'associera un de plus que le plus grand entier utilisé jusqu'à présent.

```
$num_array = [1000 => "one thousand", 100
=> "one hundred", 600 => "six hundred"];
echo $num_array[1000]; // Prints: one
thousand
```

```
$ordered = [99, 1, 7, 8];
$ordered["special"] = "Cool!";
echo $ordered[3]; // Prints: 8
echo $ordered["special"]; // Prints:
Cool!
```

```
$num_array = [1000 => "one thousand", 100
=> "one hundred", 600 => "six hundred"];
$num_array[] = "New Element in
$num_array";
echo $num_array[1001]; // Prints: New
Element in $num_array
```

```
$animals_array = ["panda"=>"very cute",
"lizard"=>"cute", "cockroach"=>"not very
cute"];
array_push($animals_array, "New Element
in $animals_array");
echo $animals_array[0]; // Prints: New
Element in $animals_array
```

Créer un tableau associatif

En PHP, les *tableaux associatifs* sont des collections de paires *clé => valeur*, où la clé doit être une chaîne ou un entier et la valeur peut être de n'importe quel type. L' `=>` opérateur permet d'associer une clé à sa valeur. Par exemple:

```
$grades = ["John" => 91, "Sally" => 94];
```

Le tableau associatif `grades` est défini à l'aide de la syntaxe de tableau court. La clé `"John"` pointe vers la valeur `91` et la clé `"Sally"` pointe vers la valeur `94`. Nous pouvons également définir ce tableau associatif en utilisant `array()`. Par exemple:

```
$grades = array(  
    "John" => 91,  
    "Sally" => 94  
);
```

