

# Tri de tableaux PHP

Dans ce chapitre, vous apprendrez à trier les éléments ou les clés d'un tableau par ordre croissant ou décroissant en PHP.

# Fonctions PHP pour trier les tableaux

Dans le chapitre précédent, vous avez appris l'essentiel des tableaux PHP, c'est-à-dire ce que sont les tableaux, comment les créer, comment afficher leur structure, comment accéder à leurs éléments, etc. Vous pouvez faire encore plus de choses avec les tableaux comme trier les éléments dans n'importe quel commande que vous aimez.

PHP est livré avec un certain nombre de fonctions intégrées conçues spécifiquement pour trier les éléments du tableau de différentes manières, par exemple par ordre alphabétique ou numérique dans l'ordre croissant ou décroissant. Ici, nous allons explorer certaines de ces fonctions les plus couramment utilisées pour trier les tableaux.

- **sort()** et **rsort()**— Pour trier les tableaux indexés
- **asort()** et **arsort()**— Pour trier les tableaux associatifs par valeur
- **ksort()** et **krsort()**— Pour trier les tableaux associatifs par clé

# Tri des tableaux indexés par ordre croissant

La fonction **sort()** permet de trier les éléments du tableau indexé par ordre croissant (alphabétiquement pour les lettres et numériquement pour les nombres).

```
<?php
// Définir le tableau
$colors = array("Red", "Green", "Blue", "Yellow");
// Trier et afficher
sort($colors);
print_r($colors);
?>
```

Cette instruction print\_r() donne la sortie suivante :

Tableau ( [0] => Blue [1] => Green [2] => Red [3] => Yellow )

De même, vous pouvez trier les éléments numériques du tableau par ordre croissant.

```
<?php
// Créer le tableau
$numbers = array(1, 2, 2.5, 4, 7, 10);
// Trier et afficher
sort($numbers);
print_r($numbers);
?>
```

Cette instruction print\_r() donne la sortie suivante :

Tableau ( [0] => 1 [1] => 2 [2] => 2.5 [3] => 4 [4] => 7 [5] => 10 )

# Tri des tableaux indexés par ordre décroissant

La fonction `rsort()` permet de trier les éléments du tableau indexé par ordre décroissant (par ordre alphabétique pour les lettres et numérique pour les nombres).

```
<?php
// Définir tableau
$colors = array("Red", "Green", "Blue", "Yellow");
// Trier et afficher le tableau
rsort($colors);
print_r($colors);
?>
```

Cette instruction `print_r()` donne la sortie suivante :

```
Array ( [0] => Yellow [1] => Red [2] => Green [3] => Blue )
```

De même, vous pouvez trier les éléments numériques du tableau par ordre décroissant.

```
<?php
/ Créer le tableau
$numbers = array(1, 2, 2.5, 4, 7, 10);
// Trier et afficher
rsort($numbers);
print_r($numbers);
?>
```

Cette instruction `print_r()` donne la sortie suivante :

```
Array ( [0] => 10 [1] => 7 [2] => 4 [3] => 2.5 [4] => 2 [5] => 1 )
```

# Tri des tableaux associatifs dans l'ordre croissant par valeur

La fonction **asort()** trie les éléments d'un tableau associatif dans l'ordre croissant selon la valeur. Cela fonctionne exactement comme **sort()**, mais il préserve l'association entre les clés et ses valeurs lors du tri.

```
<?php
// Créer tableau
$age = array("Peter"=>20, "Harry"=>14, "John"=>45, "Clark"=>35);

// Trier et afficher
asort($age);
print_r($age);
?>
```

Cette instruction `print_r()` donne la sortie suivante :

```
Array ( [Harry] => 14 [Peter] => 20 [Clark] => 35 [John] => 45 )
```

# Tri des tableaux associatifs par ordre décroissant de valeur

La fonction **arsort()** trie les éléments d'un tableau associatif par ordre décroissant en fonction de la valeur. Cela fonctionne exactement comme **rsort()**, mais il préserve l'association entre les clés et ses valeurs lors du tri.

```
<?php
// Définir tableau
$age = array("Peter"=>20, "Harry"=>14, "John"=>45, "Clark"=>35);

// Trier et afficher
arsort($age);
print_r($age);
?>
```

Cette instruction `print_r()` donne la sortie suivante :

```
Array ( [John] => 45 [Clark] => 35 [Peter] => 20 [Harry] => 14 )
```



# Tri des tableaux associatifs dans l'ordre croissant par clé

La fonction **ksort()** trie les éléments d'un tableau associatif dans l'ordre croissant de leurs clés. Il préserve l'association entre les clés et ses valeurs lors du tri, comme la fonction `asort()`.

```
<?php
// création tableau
$age = array("Peter"=>20, "Harry"=>14, "John"=>45, "Clark"=>35);

// trier et afficher
ksort($age);
print_r($age);
?>
```

Cette instruction `print_r()` donne la sortie suivante :

```
Array ( [Clark] => 35 [Harry] => 14 [Jean] => 45 [Pierre] => 20 )
```

# Tri des tableaux associatifs par ordre décroissant par clé

La fonction **krsort()** trie les éléments d'un tableau associatif dans l'ordre décroissant de leurs clés. Il préserve l'association entre les clés et ses valeurs lors du tri, comme la fonction **arsort()**.

```
<?php
// création tableau
$age = array("Peter"=>20, "Harry"=>14, "John"=>45, "Clark"=>35);

// trier et afficher
krsort($age);
print_r($age);
?>
```

Cette instruction print\_r() donne la sortie suivante :

```
Array ( [Pierre] => 20 [Jean] => 45 [Harry] => 14 [Clark] => 35 )
```