

Exercice

Tableaux

Consignes de l'exercice

- Compléter la première ligne de code pour utiliser la fonction appropriée qui retournera le nombre total d'éléments dans le tableau
- A la suite, afficher la seconde entrée du tableau \$fruits
- Compléter la deuxième ligne pour créer un tableau associatif
- A la suite compléter l'instruction pour afficher l'âge de Ben
- A la suite parcourir le tableau associatif et afficher les index et les valeurs de ce tableau
- Compléter la troisième ligne pour trier les valeurs du tableau \$colors en fonction de l'ordre croissant des lettres dans l'alphabet
- A la suite trier les valeurs du tableau \$colors en fonction de l'ordre décroissant des lettres de l'alphabet
- Compléter la quatrième ligne pour trier les valeurs du tableau de la plus petite à la plus grande

Théorie du cours

Comme en Javascript, PHP permet de créer des tableaux, pour rappel, les tableaux permettent de stocker plusieurs valeurs dans une seule variable.

Exemple :

```
$cars = array("Volvo", "BMW", "Toyota");  
echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] . " .";
```

En php, on utilise array() pour créer un nouveau tableau.

Contrairement à Javascript qui ne supporte que des tableaux indexés (des tableaux avec un index numérique), PHP permet de créer également des tableaux associatifs (des tableaux avec un index alpha-numérique) et des tableaux multi-dimensionnels (des tableaux contenant plusieurs tableaux)

- Pour créer des tableaux ayant un index numérique, il existe deux solutions :

```
$cars = array("Volvo", "BMW", "Toyota");
```

Ici, le tableau est créé en une seule fois, \$cars[0] renverra Volvo, \$cars[2] renverra Toyota (l'index commence à 0, comme en javascript)

```
$cars[0] = "Volvo";  
$cars[1] = "BMW";  
$cars[2] = "Toyota";
```

Ici, le tableau est créé progressivement, PHP crée un tableau automatiquement lorsque le premier index est défini (\$cars[0] = "Volvo")

- Pour connaître la longueur d'un tableau, on utilisera la fonction count() :

```
$cars = array("Volvo", "BMW", "Toyota");  
echo count($cars);
```

Ici, notre code retournera 3 car il y a trois valeur dans le tableau \$cars

En javascript, on utilisera .length sur le tableau pour connaître sa taille, en php l'approche est différente, le tableau ne comporte que les valeurs qu'on lui a assigné, on utilise des fonctions spécifiques pour retourner les informations.

- Parcourir les éléments d'un tableau indexé :

```
$cars = array("Volvo", "BMW", "Toyota");  
$arlength = count($cars);  
  
for($x = 0; $x < $arlength; $x++) {  
    echo $cars[$x];  
    echo "<br>";  
}
```

Nous utilisons for pour parcourir les valeurs de notre tableau, \$arlength est égal à 3 ici , la boucle for parcourt donc tout notre tableau.

- Tableaux associatifs :

Les tableaux associatifs permettent de définir un index sous forme de texte, on peut les créer de deux façons différentes :

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
```

Ici, notre tableau associatif est créé en une seule fois, \$age["Peter"] retournera 35

```
$age['Peter'] = "35";  
$age['Ben'] = "37";  
$age['Joe'] = "43";
```

Ici, on crée le tableau progressivement.

Si vous déclarez un tableau associatif, vous ne pourrez pas utiliser d'index numérique pour retourner les valeurs du tableau, \$age[0] retournera une erreur.

- Parcourir un tableau associatif :

Pour parcourir un tableau associatif, on peut utiliser foreach.

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");  
  
foreach($age as $x => $x_value) {  
    echo "Key=" . $x . ", Value=" . $x_value;  
    echo "<br>";  
}
```

L'instruction foreach parcourt le tableau \$age et retourne deux valeurs : \$x est l'index, \$x_value est la valeur de chaque entrée du tableau.

- Tableau multi-dimensionnels :

Il est parfois nécessaire de stocker des valeurs sur plusieurs dimensions, un cas concret serait un jeu d'échec, les pions peuvent être placés sur deux dimensions : la première est la largeur du plateau, la deuxième est la longueur du plateau.

Par exemple, si on souhaite indiquer qu'un pion se trouve sur la troisième case en largeur et la première case en longueur, on pourrait écrire ceci : `$echiquier[3][1] = "pion"`

Il n'y a pas de limite technique sur le nombre de dimensions d'un tableau mais il est difficile pour un être humain de représenter mentalement plus de trois dimensions (sauf exceptions)

Pour parcourir un tableau multi dimensionnel, on utilisera des boucles dans des boucles dans des boucles etc... (bref plus vous avez de dimensions plus vous avez de boucles)

- Fonctions permettant de réorganiser les tableaux :

Il existe des fonctions permettant de trier les tableaux, certaines sont propres aux tableaux associatifs

. Trier un tableau du plus petit au plus grand (ou par ordre alphabétique de la première lettre de l'alphabet vers la dernière)

```
$cars = array("Volvo", "BMW", "Toyota");  
sort($cars);
```

la fonction `sort` va trier les éléments du tableau du plus petit vers le plus grand, dans cet exemple, le nouvel ordre sera :

```
"BMW","Toyota","Volvo"
```

. Trier un tableau du plus grand vers le plus petit

```
$cars = array("Volvo", "BMW", "Toyota");  
rsort($cars);
```

La fonction `rsort` va trier les éléments du tableau du plus grand vers le plus petit, le nouvel ordre sera :

```
"Volvo","Toyota","BMW"
```

. Trier un tableau associatif de la plus petite VALEUR à la plus grande

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");  
asort($age);
```

Ici l'ordre du tableau ne changera pas car nos valeurs sont déjà classé de la plus petite à la plus grande

. Trier un tableau associatif de la plus grande VALEUR à la plus petite

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");  
arsort($age);
```

arsort va trier notre tableau de la plus grande valeur à la plus petite, notre tableau deviendra donc :

```
"Joe"=>"43", "Ben"=>"37","Peter"=>"35"
```

. Trier un tableau associatif de la plus petite CLEF à la plus grande

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");  
ksort($age);
```

Ici notre tableau va être trié de cette façon, ksort utilise la valeur des index pour faire le tri :

```
"Ben"=>"37", "Joe"=>"43", "Peter"=>"35"
```

. Trier un tableau associatif de la plus petite CLEF à la plus grande

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");  
ksort($age);
```

Ici notre tableau va être trié de cette façon, ksort utilise la valeur des index pour faire le tri :

```
"Ben"=>"37", "Joe"=>"43", "Peter"=>"35"
```

. Trier un tableau associatif de la plus grande CLEF à la plus petite

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");  
krsort($age);
```

Ici notre tableau va être trié de cette façon, krsort utilise la valeur des index pour faire le tri :

```
"Peter"=>"35", "Joe"=>"43", "Ben"=>"37"
```