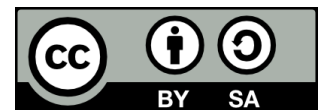


# Les tableaux en PHP



# Contenu

- **Mise en situation**
- **Le concept de tableau**
- **Utiliser les tableaux**
- **Retour sur la mise en situation**
- **Les tableaux à plusieurs dimensions**

# Mise en situation

# Mise en situation

- **Supposons que j'ai trois variables contenant des nombres entiers**
  - `$nombre1`, `$nombre2`, `$nombre3`
- **Je veux m'assurer que les trois nombres sont uniques**

# Mise en situation

```
if ($nombre1 != $nombre2 && $nombre1 != $nombre3 &&
$nombre2 != $nombre3) {
    echo 'Les trois nombres sont uniques';
}
```

# Mise en situation

- **Supposons maintenant que j'ai plutôt 100 variables contenant des nombres entiers**
  - `$nombre1, $nombre2, ..., $nombre100`
- **Je veux vérifier si ces 100 nombres sont uniques**



# Mise en situation

- **Si on devait vraiment écrire une telle condition, le résultat le plus probable serait celui-ci:**



« Laptop Toss » par Just Us 3 (CC BY-NC 2.0)



# Le concept de tableau

# Un tableau, c'est quoi?

- Un **tableau** (*array* en anglais), c'est une variable qui contient une **collection de valeurs** plutôt qu'une seule valeur

Un nombre entier

42

Un tableau de nombres entiers

42

26

18

37

# Ce qu'on peut faire avec un tableau

## • On peut...

- Déclarer et **initialiser** un tableau (comme n'importe quelle autre variable)
- Accéder au X-ième **élément** d'un tableau
- Remplacer le X-ième élément d'un tableau
- Ajouter un élément à la fin d'un tableau
- **Itérer** sur les éléments d'un tableau
  - C'est-à-dire parcourir les éléments pour pouvoir effectuer des opérations avec chacun de ceux-ci (ex: additionner tous les éléments d'un tableau)

# Utiliser les tableaux

# Déclarer un tableau

- **Déclarer un tableau avec un contenu initial**

- `$monTableau = array(17, 32, 45);`

17	32	45
----	----	----

- **Déclarer un tableau vide (pour lui ajouter des éléments plus tard)**

- `$monTableau = array();`

# Autre syntaxe possible

- **Déclarer un tableau avec un contenu initial**

- `$monTableau = [17, 32, 45];`

<b>17</b>	<b>32</b>	<b>45</b>
-----------	-----------	-----------

- **Déclarer un tableau vide (pour lui ajouter des éléments plus tard)**

- `$monTableau = [];`

# Accéder au X-ième élément d'un tableau

- **Chaque élément d'un tableau possède un indice**
  - Indice = « numéro » de l'élément dans le tableau (ex: le premier élément, le sixième, etc)
- **Important: on commence à compter à zéro!**
  - Le dernier indice du tableau est donc un de moins que son nombre d'éléments

<b>Indice</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>...</b>	<b>99</b>
<b>Valeur</b>	42	17	35	...	76

# Accéder au X-ième élément d'un tableau

- **\$nomDuTableau[indice]**

- Ex:

```
$monTableau = array(17, 42, 26, 55);  
echo $monTableau[2]; // Affiche « 26 »
```



# Remplacer le X-ième élément d'un tableau

- **\$nomDuTableau[indice] = nouvelle valeur**

- Ex:

```
$monTableau = array(17, 42, 26, 55);  
echo $monTableau[2]; // Affiche « 26 »
```

<b>17</b>	<b>42</b>	<b>26</b>	<b>55</b>
-----------	-----------	-----------	-----------

```
$monTableau[2] = 34;  
echo $monTableau[2]; // Affiche « 34 »
```

<b>17</b>	<b>42</b>	<b>34</b>	<b>55</b>
-----------	-----------	-----------	-----------

# Ajouter un élément à un tableau

- **array\_push(\$nomDuTableau, valeur);**

- Ex:

```
$monTableau = array(12, 9, 8);
```

```
array_push($monTableau, 22);
```

```
echo $monTableau[3]; // Affiche 22
```

12	9	8
----	---	---

12	9	8	22
----	---	---	----

# Itérer sur les éléments d'un tableau

- **On itère avec:**

- Une boucle
- Une variable désignant l'indice courant

- **Exemple avec une boucle For:**

```
/* Multiplier par 2 tous les éléments du tableau $monTableau  
sachant que le tableau a 10 éléments */  
for ($i = 0; $i < 10; $i++) {  
    $monTableau[$i] *= 2;  
}
```

# Itérer sur les éléments d'un tableau

```
/* Multiplier par 2 tous les éléments du tableau $monTableau  
   dont on ne connaît pas le nombre d'éléments */  
for ($i = 0; $i < count($monTableau); $i++) {  
    $monTableau[$i] *= 2;  
}
```

# La boucle For-Each

- **Nouveau type de boucle: la boucle For-Each**

```
// Afficher tous les éléments du tableau $monTableau  
foreach ($monTableau as $element) {  
    echo "$element <br />";  
}
```

# La boucle For-Each

- Pour pouvoir modifier les éléments d'un tableau à l'aide d'une boucle For-Each, il faut ajouter le symbole **&** devant la variable d'itération

```
// Multiplier par 2 tous les éléments du tableau $monTableau  
foreach ($monTableau as &$element) {  
    $element *= 2;  
}
```

# Retour sur la mise en situation

- Je veux vérifier si les 100 éléments du tableau `$nombres` sont uniques

# Solution

```
$sontTousUniques = true;
for ($i = 0; $i < 99; $i++) {
    for ($j = $i + 1; $j < 100; $j++) {
        if ($nombres[$i] === $nombres[$j]) {
            $sontTousUniques = false;
        }
    }
}
```



# Solution générale

```
$sontTousUniques = true;
for ($i = 0; $i < count($nombres) - 1; $i++) {
    for ($j = $i + 1; $j < count($nombres); $j++) {
        if ($nombres[$i] === $nombres[$j]) {
            $sontTousUniques = false;
        }
    }
}
```

# Solution générale plus efficace

```
$sontTousUniques = true;
for ($i = 0; $i < count($nombres) - 1; $i++) {
    for ($j = $i + 1; $j < count(nombres); $j++) {
        if ($nombres[$i] === $nombres[$j]) {
            $sontTousUniques = false;
            break;
        }
    }
}
if (!$sontTousUniques) {
    break;
}
}
```

# Solution générale plus efficace... en plus court!

```
$sontTousUniques = true;
for ($i = 0; $i < count($nombres) - 1; $i++) {
    for ($j = $i + 1; $j < count(nombres); $j++) {
        if ($nombres[$i] === $nombres[$j]) {
            $sontTousUniques = false;
            break 2;
        }
    }
}
```

# Les tableaux à plusieurs dimensions

# Les tableaux à une dimension

- **Les tableaux que nous avons vus jusqu'à maintenant sont à une dimension**
  - On les appelle aussi des **vecteurs**
- **Un tableau à une dimension est un tableau à une seule « ligne »**

Une seule ligne

<b>42</b>	<b>26</b>	<b>18</b>	<b>37</b>
-----------	-----------	-----------	-----------

# Les tableaux à deux dimensions

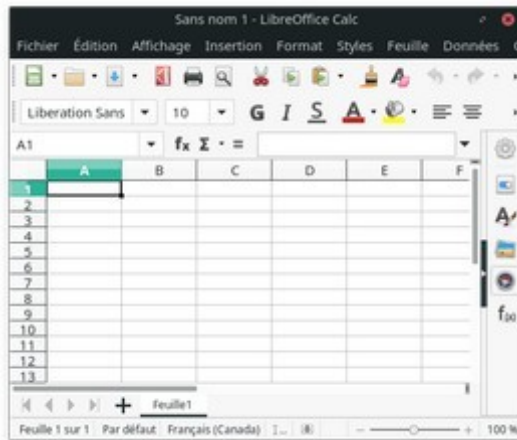
- **Il existe des tableaux à deux dimensions**
  - On les appelle aussi des **matrices**
  - Ils ont des **lignes** et des **colonnes**

<b>Ligne / Colonne</b>	<b>0</b>	<b>1</b>	<b>2</b>
<b>0</b>	[0, 0]	[0, 1]	[0, 2]
<b>1</b>	[1, 0]	[1, 1]	[1, 2]

# Exemples de matrices



Photo par Chase Clark sur Unsplash



	MINI (8 po)	PETITE (10 po)	MOYENNE (12 po)	LARGE (14 po)
<b>Pepperoni et fromage</b>	11,99\$	14,79\$	18,99\$	24,49\$
<b>Bacon et fromage</b>	11,99\$	14,79\$	18,99\$	24,49\$
<b>Toute garnie</b> Pepperoni, poivrons verts, champignons et fromage	12,49\$	15,29\$	20,49\$	26,49\$
<b>Végétarienne</b> Champignons, poivrons verts, fromage, oignons, tomates et olives vertes	12,49\$	15,29\$	20,49\$	26,49\$
<b>Jardinière</b> Champignons, poivrons verts, fromage, oignons, tomates, brocoli et chou-fleur	13,49\$	17,29\$	23,49\$	29,99\$

Tiré du menu du restaurant Freddy Pizzeria de Sherbrooke

# Représentation des matrices

- **Dans la pratique, une matrice est un tableau de tableaux!**

<b>Tableaux de tableaux</b>	<b>0</b>	<b>1</b>	<b>2</b>
<b>Tableau 0</b>	[0][0]	[0][1]	[0][2]
<b>Tableau 1</b>	[1][0]	[1][1]	[1][2]



# Déclarer une matrice en PHP

- **Une matrice est un tableau de tableaux. Donc c'est exactement ce qu'on fait!**

```
// Déclaration d'une matrice de 2 lignes x 3 colonnes  
$maMatrice = array(  
    array(3, 1, 7),  
    array(4, 28, 15)  
);
```

Tableaux de tableaux	0	1	2
Tableau 0	3	1	7
Tableau 1	4	28	15

# Accéder aux valeurs d'une matrice

Tableaux de tableaux	0	1	2
Tableau 0	3	1	7
Tableau 1	4	28	15

- `$maMatrice[0][0]` vaut 3
- `$maMatrice[0][1]` vaut 1
- `$maMatrice[1][2]` vaut 15
- Etc.

# Itérer sur une matrice

```
for ($i = 0; $i < count($maMatrice), $i++) {  
    for $j = 0; $j < count($maMatrice[$i]), $j++) {  
        echo $maMatrice[$i][$j] . ' ';  
    }  
    echo '<br />';  
}
```

# Les tableaux à plus de deux dimensions

- **On peut aussi créer des tableaux à trois dimensions et plus**

- Un tableau à trois dimensions est donc un tableau de tableaux de tableaux!
- Plus rare
- Exemple d'utilisation: représenter les points d'un espace en 3D ( $x, y, z$ )

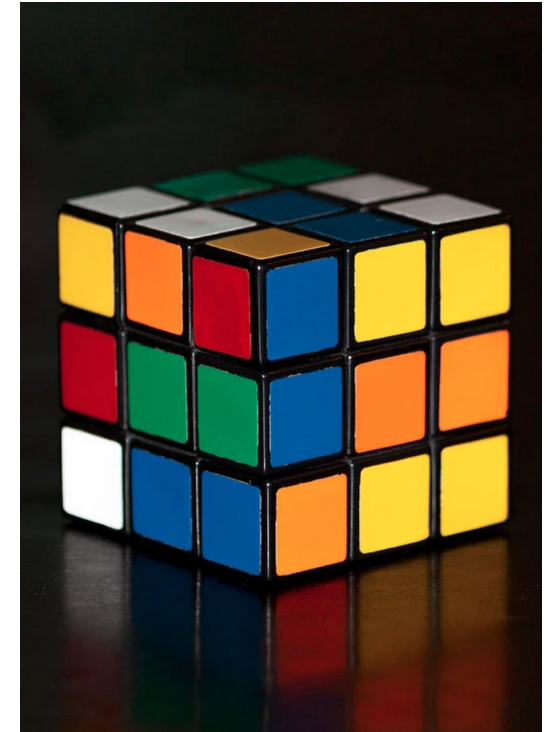


Photo par Elijah Ekdahl sur Unsplash

# Fin de la présentation

Des questions?



Photo par Emily Morter sur Unsplash