

Array



COURS

4

**TRAVAUX
PRATIQUES**



Un **tableau** est une "**SUPER VARIABLE**" qui va permettre de **stocker un ensemble de valeurs** (une **liste** de valeurs)



Les **tableaux indexés**



**Tableaux
unidimensionnels**



Les **tableaux associatifs**





Un tableau unidimensionnel est un tableau dont chaque élément contient une valeur de type simple (string, int, float, boolean)



Un **tableau multidimensionnel** est un **tableau** dont **chaque élément** contient une **valeur** de **type tableau**

Dimension = 2



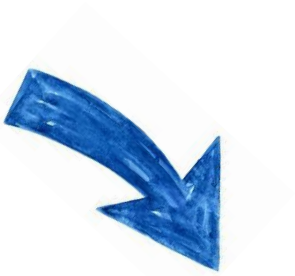
Tableau bidimensionnel



Déclaration d'un **tableau indexé 2D**



```
$tableau = [
  0 [val1,val2,val3],
  1 [val4,val5,val6],
  2 [val7,val8,val9]
]
```



Chaque **élément** est un **tableau indexé**

0			1			2		
0	1	2	0	1	2	0	1	2
[val1,val2,val3]			[val4,val5,val6]			[val7,val8,val9]		

index

\$tableau



Affichage d'un **tableau indexé 2D**

```
$tableau = [  
    [val1,val2,val3],  
    [val4,val5,val6],  
    [val7,val8,val9]  
]
```

```
print_r($tableau);
```





Accès aux éléments d'un tableau indexé 2D

```
$tableau = [  
  0 [val1,val2,val3],  
  1 [val4,val5,val6],  
  2 [val7,val8,val9]  
]
```

index

\$tableau[0]



⁰ ¹ ²
[val1, val2, val3]



 **Tableau**

\$tableau[0][1]



val2



Représentation d'un **tableau indexé 2D**

\$tableau

	0	1	2
0	012 [val1,val2,val3]	012 [val4,val5,val6]	012 [val7,val8,val9]

	Colonne 0	Colonne 1	Colonne 2
Ligne 0	tab [0][0]	tab [0][1]	tab [0][2]
Ligne 1	tab [1][0]	tab [1][1]	tab [1][2]
Ligne 2	tab [2][0]	tab [2][1]	tab [2][2]



Parcours d'un **tableau indexé 2D**

```
$tableau = [
    [val1,val2,val3],
    [val4,val5,val6],
    [val7,val8,val9]
]
```

	Colonne 0	Colonne 1	Colonne 2	Colonne 3
Ligne 0	tab [0][0]	tab [0][1]	tab [0][2]	tab [0][3]
Ligne 1	tab [1][0]	tab [1][1]	tab [1][2]	tab [1][3]
Ligne 2	tab [2][0]	tab [2][1]	tab [2][2]	tab [2][3]

// Parcours de chaque ligne

```
for($i=0; $i<count($tableau); $i++) {
```

// Parcours de chaque colonne

```
for($j=0; $j<count($tableau[$i]); $j++) {
```

```
    echo $tableau[$i][$j];
```

```
}
```

```
}
```



**Boucle
imbriquée**



for



Parcours d'un **tableau indexé 2D**

```
$tableau = [  
    [val1,val2,val3],  
    [val4,val5,val6],  
    [val7,val8,val9]  
]
```

	Colonne 0	Colonne 1	Colonne 2	Colonne 3
Ligne 0	tab [0][0]	tab [0][1]	tab [0][2]	tab [0][3]
Ligne 1	tab [1][0]	tab [1][1]	tab [1][2]	tab [1][3]
Ligne 2	tab [2][0]	tab [2][1]	tab [2][2]	tab [2][3]

```
// Parcours de chaque ligne  
foreach($tableau as $ligne) {  
    // Parcours de chaque colonne  
    foreach($ligne as $valeur) {  
        echo $valeur;  
    }  
}
```



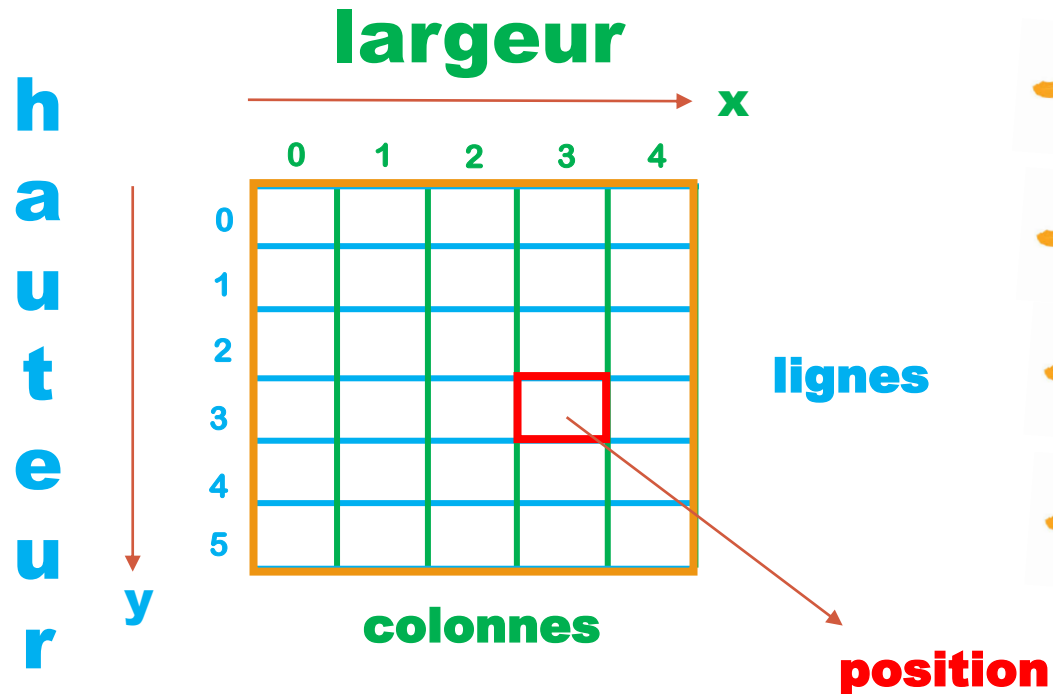
**Boucle
imbriquée**



foreach



Array



La **grille** est délimitée par une **largeur** et une **hauteur** : dimensions de la **grille**

La **largeur** : le nombre de **colonnes** de la **grille** , axe des **x**

La **hauteur** : le nombre de **lignes** de la grille, axe des **y**

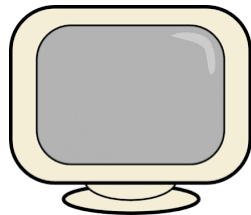
Intersection d'une **ligne** et d'une **colonne** : une **position** ☐



L'ÉNONCÉ

Ecrire un programme **grille.php** qui :

- Demande à l'utilisateur de saisir les **dimensions** de la **grille**
- Initialise la **grille** : chaque **position** devra contenir le caractère '-'
- Affiche la **grille**



```
Saisir une largeur : 6
Saisir une hauteur : 6

- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
```

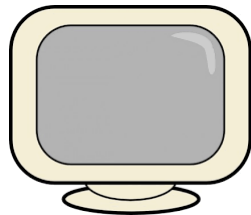


grille.php

L'ÉNONCÉ

Modifier le programme **grille.php** afin de modifier l'affichage de la **grille**. Vous devrez :

- Afficher le numéro de chaque ligne en **bleu**
- Encadrer chaque position avec le séparateur '|'. De plus chaque position devra être affichée en **jaune**



php

sprintf



```
Saisir une largeur : 6
Saisir une hauteur : 6
00 | - | - | - | - | - |
01 | - | - | - | - | - |
02 | - | - | - | - | - |
03 | - | - | - | - | - |
04 | - | - | - | - | - |
05 | - | - | - | - | - |
```

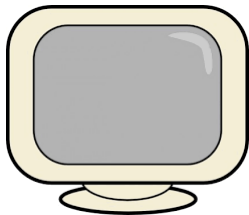


grille.php

L'ÉNONCÉ

Modifier le programme **grille.php** afin de modifier l'affichage de la **grille**. Vous devrez :

- Afficher le numéro de chaque colonne en **bleu**



```
Saisir une largeur : 6
Saisir une hauteur : 6

  00  01  02  03  04  05
00 | - | - | - | - | - |
01 | - | - | - | - | - |
02 | - | - | - | - | - |
03 | - | - | - | - | - |
04 | - | - | - | - | - |
05 | - | - | - | - | - |
```



Déclaration d'un **tableau indexé 2D**

```
$tableau = [  
  0 ['cle1'=>val1, 'cle2'=>val2],  
  1 ['cle1'=>val3, 'cle2'=>val4],  
  2 ['cle1'=>val5, 'cle2'=>val6]  
]
```

index



Chaque **élément**
est un **tableau**
associatif



Accès aux éléments d'un tableau indexé 2D

```
$tableau = [
    0 ['cle1'=>val1, 'cle2'=>val2],
    1 ['cle1'=>val3, 'cle2'=>val4],
    2 ['cle1'=>val5, 'cle2'=>val6]
]
```



index

```
$tableau[0]
```



['cle1'=>val1, 'cle2'=>val1]

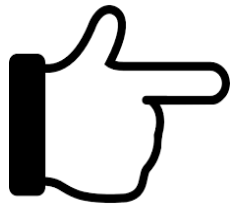
```
$tableau[0]['cle1']
```



val1

**resultatsBTS.php**

L'ÉNONCÉ



Ecrire un programme **resultatsBTS.php qui :**

- **Crée un **tableau bidimensionnel** permettant de stocker le **nom** et la **moyenne** de chaque **étudiant****
- **Affiche le **nom** et la **moyenne** de chaque **étudiant**.**

Pour ce faire vous proposerez une version avec une **boucle for et une version avec une **boucle foreach****



resultatsBTS.php

L'ÉNONCÉ

Modifier le programme `resultatsBTS.php` afin de :

- **Afficher le `nom` et la `moyenne` de tous les `étudiants reçus` au BTS**
- **Afficher le `nom` et la `moyenne` du `major de promotion` (étudiant ayant obtenu la `moyenne la plus haute`)**



Déclaration d'un **tableau associatif 2D**

```
$tableau = [  
    'cleA' => ['cle1'=>val1, 'cle2'=>val2],  
    'cleB' => ['cle1'=>val3, 'cle2'=>val4],  
    'cleC' => ['cle1'=>val5, 'cle2'=>val6]  
]
```

clés



Chaque **élément** est un **tableau associatif**



Accès aux éléments d'un tableau associatif 2D

```
$tableau = [
    'cleA' => ['cle1'=>val1, 'cle2'=>val2],
    'cleB' => ['cle1'=>val3, 'cle2'=>val4],
    'cleC' => ['cle1'=>val5, 'cle2'=>val6]
]
```

\$tableau['cleA']

➡ **['cle1'=>val1, 'cle2'=>val1]**

\$tableau['cleA']['cle1']

➡ **val1**



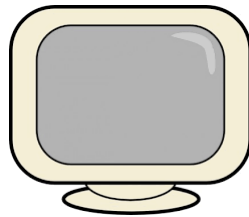
L'entreprise **Truc** vend des produits sur internet. Voici les différents produits :

Référence	Désignation	Prix	Stock
P0001	Produit 1	50.99	10
P0002	Produit 2	99;99	20
P0003	Produit 3	25.50	6
P0004	Produit 4	130.99	4



produits.php

L'ÉNONCÉ



Ecrire un programme **produits.php** qui :

- Créé un **tableau associatif bidimensionnel** permettant de stocker des **produits**. Un produit est **identifié** (la clé) par une **référence** et est **caractérisé** par une **désignation**, un **prix** et une **quantité en stock**.
- Affiche la **référence** et la **désignation** de chaque **produit**.

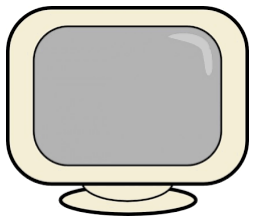
```
P0001 : Produit 1
P0002 : Produit 2
P0003 : Produit 3
P0004 : Produit 4
```



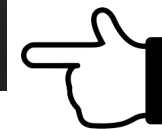
L'ÉNONCÉ

Modifier le programme `produits.php` afin de :

- **Calculer et afficher la *valeur du stock***



```
P0001 : Produit 1
P0002 : Produit 2
P0003 : Produit 3
P0004 : Produit 4
La valeur du stock est de 3 186,66
```

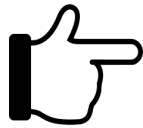


Formatage de la valeur



produits.php

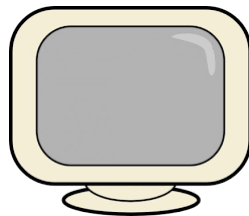
L'ÉNONCÉ



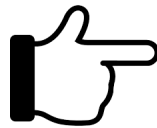
Modifier le programme **produits.php** afin de :

- Récupérer dans un tableau les **références des produits à réapprovisionner**
- Afficher les **références à réapprovisionner**

Un **produit à réapprovisionner** est un produit dont la **quantité en stock est inférieure à 7**



Références à réapprovisionner



```
P0001 : Produit 1
P0002 : Produit 2
P0003 : Produit 3
P0004 : Produit 4
La valeur du stock est de 3 186,66
```

```
Array
```

```
(
    [0] => P0003
    [1] => P0004
)
```

print_r



produits.php

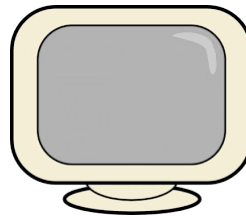
L'ÉNONCÉ



Modifier le programme `produits.php` afin de :

- **Demander à l'utilisateur la **référence** d'un produit réapprovisionné**
- **Demander à l'utilisateur la **quantité reçue****
- ****Mettre à jour le stock** du produit dans le **tableau****

```
Saisir une référence : P0003
Saisir une quantité reçue : 20
La quantité a bien été mise à jour !
```



```
Saisir une référence : P0005
La référence n'existe pas !
```

