

Array



COURS

1

**TRAVAUX
PRATIQUES**



Une **variable** est un **élément** défini un **nom** permettant de **stocker une seule valeur**



Une **variable** est **associée** à un **emplacement mémoire** (endroit où est mémorisé la valeur)

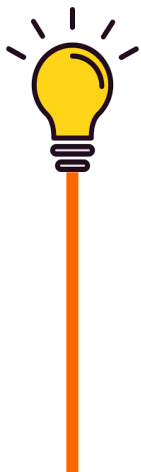


La **valeur** contenue dans une **variable** peut-être de **type** entier, réel, chaîne de caractères ou booléen

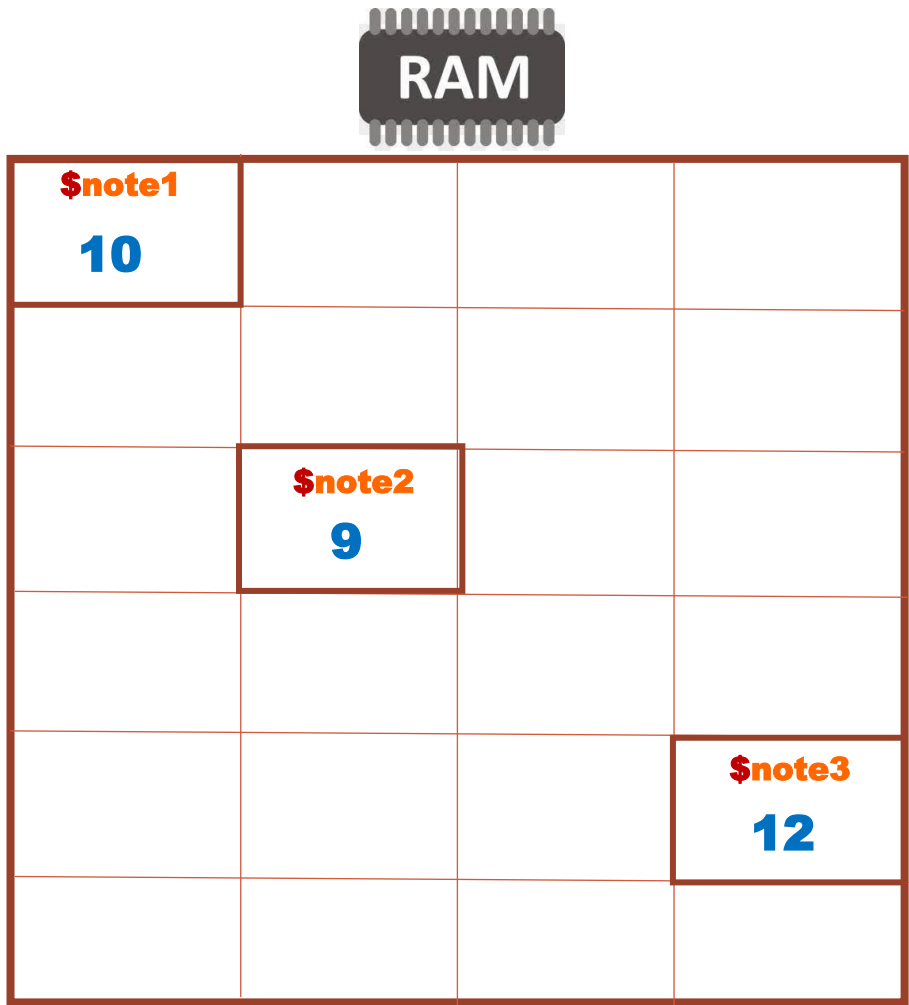


Le **contenu** d'une **variable** peut-être **modifié en cours d'exécution** du programme

```
$note1 = 10;
$note2 = 12;
$note3 = 9;
```



Chaque variable est stockée à un emplacement mémoire alloué aléatoirement : les variables ne sont pas contiguës en mémoire



```
$note1 = 10;  
$note2 = 12;  
$note3 = 9;  
...
```



Comment faire si l'on doit gérer un nombre significatif de notes (+ de 20 notes)



Créer une variable pour chaque note



c'est pas possible



c'est pas possible



20

notes



1 \$note1 = 10;

\$note2 = 12;

\$note3 = 9;

\$note4 = 9;

\$note5 = 12;

...

20 \$note20 = 14;



Afficher l'ensemble des notes.



Afficher la moyenne des notes



Afficher uniquement les notes en dessous de la moyenne



Afficher le nombre de notes au-dessus de la moyenne



Les **tableaux** vont permettre d'apporter une solution au problème



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



Déclaration et initialisation d'un tableau

```
$nomTableau = array(liste_valeurs);
```

OU

```
$nomTableau = [liste_valeurs];
```

PREFERRED



**Nom donné au
tableau**



**Liste des valeurs
initiales du tableau**

RAPPEL

```
$note1 = 10;  
$note2 = 12;  
$note3 = 9;  
...
```

EXAMPLE

```
$notes = [10,12,9];
```



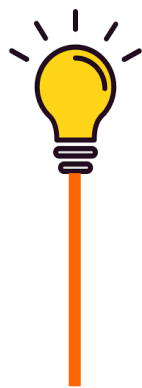
Valeur
=
Élément



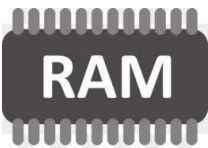
Déclaration d'un tableau
\$notes initialisé avec **3**
valeurs

\$notes = [10,12,9];

\$notes



Les **éléments** du tableau sont stockés à des **emplacements contigus en mémoire**



10	12	9	



Chaque **élément** du tableau est **identifié** par un **numéro** que l'on appelle un **index**



Accès à un **élément** du **tableau** : il faut **connaître** son **index**

index

\$notes

<u>0</u>	<u>1</u>	<u>2</u>
10	12	9



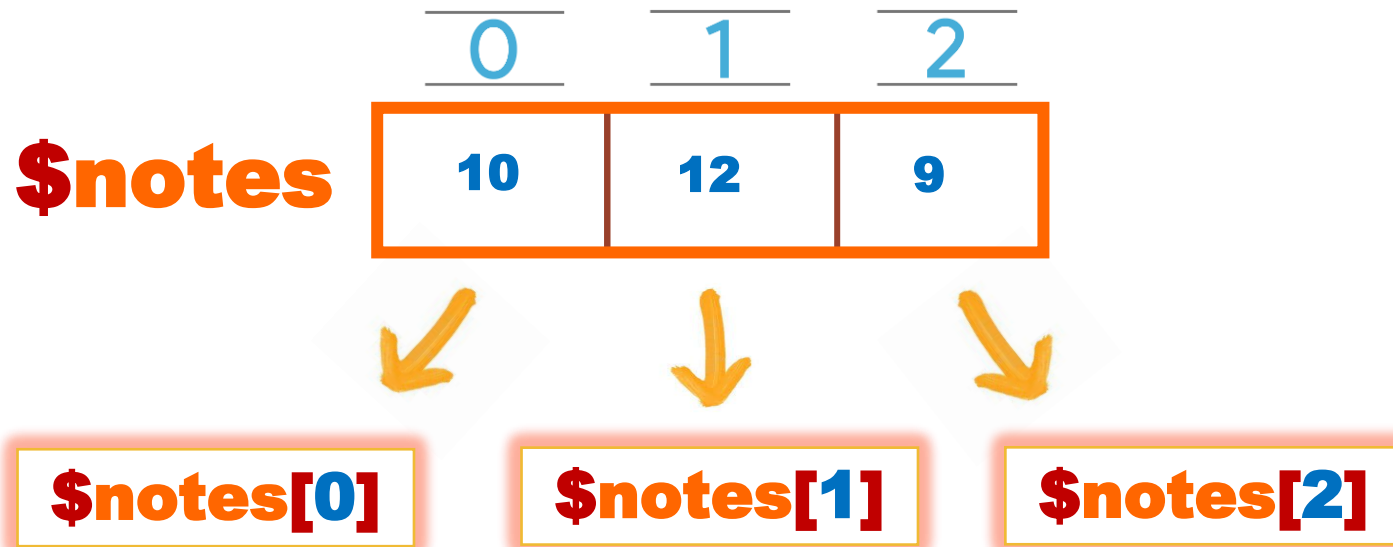
Les **index** commencent à partir de **0**



Accès à un **élément** du **tableau** se fait à partir de son **index**

\$nomTableau[index]

EXAMPLE





Affichage des éléments du tableau

```
$notes = [10,12,9];  
echo $notes[0];  
echo $notes[1];  
echo $notes[2];
```



Accès à chaque élément

c'est pas possible



Ne pourrait-on pas trouver une meilleure solution ?



Affichage des éléments du tableau

```
$notes = [10,12,9];  
echo $notes[0];  
echo $notes[1];  
echo $notes[2];
```

Les **index**
varient de **0** à **2**



Utilisation d'une boucle **for**



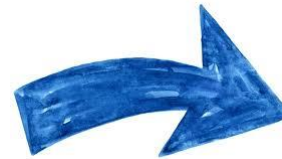
```
$notes = [10,12,9];  
for($i=0;$i<=2;$i++) {  
    echo $notes[$i];  
}
```



Affichage des éléments du tableau



```
$notes = [10,12,9];  
for($i=0;$i<=2;$i++) {  
    echo $notes[$i];  
}
```



Ici **2** représente l'**index** du **dernier élément**



On ne connaît pas forcément à l'avance l'index** du **dernier élément****



Taille d'un tableau : **nombre** d'**éléments** du tableau

	<u>0</u>	<u>1</u>	<u>2</u>
\$notes	10	12	9



Taille = 3



Index dernier élément

=

Taille - 1



Fonction PHP **count()**

count(\$nomTableau)



```
$notes = [10,12,9];  
for($i=0;$i<count($notes);$i++) {  
    echo $notes[$i];  
}
```



```
$notes = [10,12,9];  
for($i=0;$i<count($notes);$i++) {  
    echo $notes[$i];  
}
```



Parcourir le
tableau



Parcourir un tableau consiste à **traiter**
chaque **élément** du tableau **tant qu'il y en a**
UN à traiter



Il existe une **autre méthode** afin de **parcourir** un **tableau**



La boucle **foreach**

```
foreach($nomTableau as $element) {  
    // instructions  
}
```



A chaque **itération**, la variable **\$element** va contenir la **valeur** de l'**élément** à traiter

```
$notes = [10,12,9];  
for($i=0;$i<count($notes);$i++) {  
    echo $notes[$i];  
}
```

VS

```
$notes = [10,12,9];  
foreach($notes as $note) {  
    echo $note;  
}
```

+Simple





PHP propose un **ensemble de fonctions** permettant de manipuler les tableaux



<https://www.php.net/manual/fr/ref.array.php>



Array



moyenne-notes.php

L'ÉNONCÉ

Ecrire un programme **moyenne-notes.php** permettant de calculer et afficher la **moyenne** des **notes** d'un étudiant de BTS SIO à son examen final.

Les notes sont les suivantes **10 12,5 14 7 15 9,5 et 8**



moyenne-notes.php

L'ÉNONCÉ

Modifier le programme `moyenne-notes.php` afin d'afficher une **appréciation** en fonction des règles suivantes :

Moyenne	Appréciation
< 10	Vous n'avez pas votre diplôme
Entre 10 et 12 (exclu)	Mention passable
Entre 12 et 14 (exclu)	Mention assez bien
Entre 14 et 16 (exclu)	Mention bien
>=16	Mention très bien

**meilleure-note.php**

Les notes sont les suivantes **10 12,5 14 7 15 9,5** et **8**

L'ÉNONCÉ

Ecrire un programme `meilleure-note.php` permettant de trouver et afficher la **note la plus haute obtenue par l'élève**

Vous proposerez 2 versions différentes du programme :

- **Une 1^{ère} version utilisant une boucle**
- **Une 2^{nde} version utilisant une fonction proposée par PHP**



recherche-produits.php



```
Saisir une équipe :france
----- Version boucle -----
France sera présente à la prochaine coupe du monde !
----- Version fonction PHP -----
France sera présente à la prochaine coupe du monde !
```

L'ÉNONCÉ

Ecrire un programme **recherche-equipe.php** permettant de **rechercher** si une **équipe** sera **présente** à la prochaine coupe du monde de football

Votre programme doit demander à l'utilisateur de saisir une équipe



\$equipes = ['France','Brésil','Argentine','Espagne','Sénégal','Australie','Belgique'];



Une 1^{ère} version utilisant une boucle



Une 2nde version utilisant une fonction proposée par PHP



pourcentage-avancement.php



L'ÉNONCÉ

Ecrire un programme **pourcentage-avancement.php** qui :

- **Déclare et remplit** un tableau avec **100 000 éléments** (chaque élément aura la **même valeur** pour simplifier)
- **Parcours** le tableau et **affiche le pourcentage d'avancement** lors du parcours

```
debut traitement
```

```
debut traitement
En cours 15%
```

```
debut traitement
100 %
fin traitement

100000 valeurs ont été traitées !
```