

# Les tableaux en PHP

# C'est quoi un tableau en PHP?

- Un tableau appelé aussi **array**, est une variable, qui peut contenir plusieurs valeurs **pas obligatoirement de même type**. On peut avoir des chaînes de caractères, des nombres,... dans le même tableau.

Il existe deux types de tableaux:

1. Les tableaux numérotés
2. Les tableaux associatifs

# Les tableaux numérotés

- Soit un tableau **liste** qui contient les noms de personnes. Chaque valeur est identifiée par un numéro qu'on appelle **clé** :

clé	Valeur
0	Mohamed
1	Karim
2	Ali
...	...

- La clé c'est l'équivalent de indice (index)
- Dans un tableau numéroté, les **clés** sont **numériques**.

# Déclarer une variable tableau

- Si on veut déclarer un tableau **vide**, on écrit:

```
$MonTableau=array();
```

- La déclaration d'un tableau **n'est pas obligatoire**.
- Il existe plusieurs manière de remplir un tableau en PHP.

# Initialiser un tableau numéroté

- Si on veut utiliser un tableau et l'initialiser avec des valeurs, on peut le faire de manière suivante:

```
<?php  
  
$liste1=array('Mohamed','Karim','Ali',...) ;  
$liste2=array(30, 12, -3, 10, 543) ;  
$liste3=[15,20,11,-7]  
  
?>
```

# Initialiser un tableau numéroté

- Il existe une autre manière d'initialiser les éléments du tableau:

```
<?php  
$liste[0]='Mohamed';  
$liste[1]='Karim';  
$liste[2]='Ali';  
...  
?>
```

# Initialiser un tableau numéroté

- On peut laisser PHP ajouter les clés de chaque valeur.

```
<?php  
  
$liste[]='Mohamed';  
$liste[]='Karilm';  
$liste[]='Ali';  
...  
?>
```

C'est-à-dire que le premier élément qui est affecté au tableau possède l'indice **0**, le deuxième d'indice **1** et ainsi de suite.

# Donner une clé à une valeur dans un tableau numéroté

- Les indices dans un tableau numéroté commence par défaut avec la valeur **0**, mais on a la possibilité de définir nous même la **valeur initiale de la clé**, de la manière suivante:

```
$liste=array(10=>'Mohammed' , 'Karim' , 'Ali');  
$liste=array(10=>'Mohammed' , 23=>'Karim' , 25=>'Ali');
```



# Accéder à un élément d'un tableau numéroté

- Pour référencer une valeur dans un tableau numéroté, il suffit d'indiquer **la valeur de la clé** entre parenthèses.

```
echo $liste[2];
```

# Initialiser un tableau associatif

- Un tableau associatif est un tableau qui utilise des **clés textuels** au lieu de clés numériques.
- *Exemple:*

```
$liste=array('Nom' => 'Talbi' , 'Prenom' => 'Mohamed' , 'Ville' => 'Oujda');
```

# Initialiser un tableau associatif

- On peut initialiser un tableau comme suit:

```
<?php  
  
$liste['Nom']='Talbi';  
$liste['Prenom']='Mohamed';  
$liste['Ville']='Oujda';  
  
?>
```

```
<?php  
  
$liste=[ 'Nom'=>"Talbi" , 'Prenom'=>"Mouhamed" , 'Ville'=>'Oujda' ]  
  
?>
```

# Accéder à une valeur d'un tableau associatif

- Pour afficher la valeur d'un tableau associatif, on met le nom de la clé entre []:

```
echo $liste['Nom'];
```

# Parcourir un tableau

- En utilisant la boucle for:

```
<?php
$liste=array('Mohamed','Karilm','Ali');

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

echo $liste[$i];

}

?>
```

- La fonction **count()** récupère le nombre des éléments d'un tableau passé en paramètre.

# Parcourir un tableau

- En utilisant la boucle **foreach**:

```
<?php
$liste=array('Mohamed' , 'Karim' , 'Ali');

foreach( $liste as $element ) {

echo $element ;

}
?>
```

# Parcourir un tableau

- S'il s'agit d'un tableau associatif:

```
<?php
$liste=array( 'Nom'=>'Talbi',
              'Prenom'=>'Mohamed',
              'Ville'=>'Oujda' );

foreach($liste as $element){
    echo $element;
}
?>
```

# Parcourir un tableau

- Pour le tableau associatif on peut utiliser la syntaxe suivante également:

```
<?php
$liste=array('Nom'=>'Talbi',
             'Prenom'=>'Mohamed',
             'Ville'=>'Oujda');

foreach($liste as $Macle => $element){
echo '['.$Macle.'] vaut :'. $element;
}
?>
```



# Quelques fonction utiles

- **array\_push(\$tab,\$val):** permet d'ajouter une ou plusieurs valeurs à la fin d'un tableau.
- **array\_unshift(\$tab,\$val):** permet d'ajouter une ou plusieurs valeurs au début d'un tableau.
- **array\_pop(\$ab):** permet de supprimer le dernier élément du tableau
- **array\_shift(\$tab):** permet de supprimer le premier élément du tableau
- **print\_r(\$tab):** permet d'afficher le contenu d'un tableau
- **in\_array(\$val,\$tab):** permet de vérifier si \$val existe dans le tableau \$tab
- **sort(\$tab):** permet de trier les **valeurs** du tableau dans le sens croissant (tableau numérique)
- **asort(\$tab):** permet de trier les valeurs du tableau dans le sens croissant (associatifs)
- **rsort(\$tab) :** permet de trier les valeurs du tableau dans le sens décroissant.(numérique)
- **arsort(\$tab) :** permet de trier les valeurs du tableau dans le sens décroissant (associatif)
- **ksort(\$tab):** permet de trier les clés du tableau dans le sens croissant
- **krsort(\$tab):** permet de trier les clés du tableau dans le sens décroissant

# La fonction `str_split()`

- **`str_split()`**: permet de convertir une chaîne en un tableau de caractères
  - **Exemple 1:**  
`$t=str_split("Bonjour"); // $t est un tableau`  
`$t` sera comme suit `['B','o','n','j','o','u','r']`
  - **Exemple 2:**  
`$t=str_split("Bonjour" , 2);`  
`$t` sera comme suit `['Bo','nj','ou','r']`

# La fonction explode()

- ***Explode()***: permet de découper une chaîne selon un caractère et retourne un tableau de chaînes
- **syntaxe**:

```
$t=explode(separateur , chaîne, limite)
```

- **Exemple**:

```
$str = 'Bonjour ; les ; amis';  
$t=explode(';', $str);
```

# La fonction array\_slice()

- Cette fonction permet de retourner un ou plusieurs éléments d'un tableau, selon l'indice fournit.
- **Exemple 1:**

```
$T=array("un", "deux" , "trois" , "quatre" );  
$v=array_slice($T,2,1) ;  
echo $v ;           //resultat : "trois"
```

- **Exemple 2:**

```
$T=array("un", "deux" , "trois" , "quatre" );  
$v=array_slice($T,2) ;  
echo $v ;           //resultat : "trois" et "quatre"
```

# La fonction array\_splice()

- La fonction **array\_splice()** **supprime** les éléments sélectionnés d'un tableau et les remplace par de nouveaux éléments. La fonction renvoie aussi un tableau des éléments supprimés.
- Si la fonction ne supprimer aucun élément (longueur=0) , le tableau remplacé **sera inséré** à partir de la position du paramètre indice.
- **Syntaxe:**

```
array_splice(tableau1 , indice , longueur , tableau2 )
```

- **Tableau1:** paramètre obligatoire, c'est le tableau duquel on supprimer
- **Indice :** paramètre obligatoire, d'est l'indice d'où on va commencer la suppression.
- **Longueur:** pas obligatoire. C'est le nombre des éléments à supprimer. Si il n'est pas donnée il va supprimer de indice jusqu'à la fin du tableau
- **Tableau2 :** paramètre non obligatoire. C'est le tableau des éléments qui vont être insérés dans le tableau.

# La fonction array\_splice()

- **Exemple 1:**

```
$T=[15, 18, 10, -7, 23];  
array_splice($T, 1, 2);  
  
foreach($T as $element){  
    echo $element ;  
}  
// 15 -7 23
```

- **Exemple 2 :**

```
$T=[15, 18, 10, -7, 23];  
array_splice($T, 1, 0, 500);  
  
foreach($T as $element){  
    echo $element ;  
}  
// 15 500 18 10 -7 23
```

# Les tableaux à deux dimensions

- Tableau multidimensionnel numéroté stockant des tableau numérotés :

```
$M1 = [ [1, 2, 4, 8, 16], [1, 3, 9, 27, 81] ]; //2X5
$M2= array( //3X3
    array(1,8,23) ,
    array(7,4,6) ,
    array(-8,12,3)
);
```

- Parcourir un tableau à deux dimensions:

```
for ($i=0 ;$i<3;$i++){
    for($j=0 ;$j<3 ;$j++){
        echo $M2[$i][$j];
    }
    echo "<br>" ;
}
```

# Les tableaux à deux dimensions

- Tableau multidimensionnel numéroté stockant des tableaux associatifs et une valeur simple.

```
$utilisateurs = [  
    [ 'nom' => 'Mathilde', 'mail' => 'math@gmail.com' ] ,  
    [ 'nom' => 'Pierre', 'mail' => 'pierre.giraud@edhec.com' ] ,  
    [ 'nom' => 'Amandine', 'mail' => 'amandine@lp.fr' ]  
];
```

- Parcourir ce tableau :

```
for($i=0 ; $i<3 ; $i++){  
    foreach ($M3[$i] as $selem ) {  
        echo $selem;  
    }  
    echo "<br>" ;  
}
```



# Les tableaux à deux dimensions

- Tableau multidimensionnel numéroté stockant des tableaux associatifs et une valeur simple.

```
$produits = [  
    'Livre' => [ 'poids' => 200, 'quantite' => 10, 'prix' => 15 ],  
    'Stickers' => [ 'poids' => 10, 'quantite' => 100, 'prix' => 1.5 ]  
];
```

```
$produits['Livre']['poids'];  
$produits['Livre']['quantite'];  
$produits['Livre']['prix'];  
  
$produits['Stickers']['poids'];  
$produits['Stickers']['quantite'];  
$produits['Stickers']['prix'];
```

```
foreach ($produits as $ele){  
    foreach($ele as $x){  
        echo $x." ";  
    }  
    echo "<br>";  
}
```