



# PHP 5

---

## Tableaux - Formulaires



## Les tableaux



# Les tableaux

---

## ■ Généralités

- Déclaration : plusieurs manières :

```
<?php
```

```
// Déclaration d'un tableau vide
```

```
$fruits = array();
```

```
// Déclaration d'un tableau indexé numériquement
```

```
$legumes = array('carotte', 'poivron', 'aubergine', 'chou');
```

```
// Déclaration d'un tableau mélangeant les types entier et  
chaîne
```

```
$tab = array($variable, "texte", 153, 56);
```

```
?>
```



# Les tableaux à indices numériques

---

## ■ Déclaration

- En PHP, la déclaration est implicite, nul besoin de préciser à l'avance le nombre d'éléments du tableau...
- Par affectation

```
$t[0]="bonjour";  
$t[1]="bonsoir";  
$t[2]="bla bla bla";
```

- Utilisation

```
echo "case numéro 2 : ".$t[2]."<BR>\n";  
for ($i=2 ; $i<6 ; $i++) {  
    echo "case numéro $i : ".$t[$i]."<BR>\n";  
}
```



# Les tableaux

---

## ■ Exemple : array1.php

```
<?php
```

```
    $tab = array("bonjour", 3, " tout", -5.4, " monde");
```

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

```
        print("indice $i element = $tab[$i] <br>");
```

```
    print('nombre element = '.count($tab).'<br>');
```

```
    $tab[5] = "le 6-ieme";
```

```
    $tab[] = "encore"; //se met à l'indice 6
```

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

```
        print("indice $i element = $tab[$i] <br>");
```

```
?>
```



# Les tableaux associatifs

## ■ Création

- Pour créer un tableau associatif, il faut donner pour chaque élément, le couple : (clé => valeur)

```
<?php
    $tab = array(
        "prenom" => "Cyril";
        "ville" => "Paris";
        "travail" => "informatique"
    );
?>
```

Clé	Valeur
prenom	Cyril
ville	Paris
travail	informatique



# Les tableaux associatifs

---

## ■ Exemples de tableaux simples :

- clé => valeur

```
$fruits = array ("a"=>"orange", "b"=>"banane", "c"=>"pomme");
```

```
$trous = array (1=>"premier", 2 => "second", 3 => "troisième");
```

## ■ Exemple de tableau de tableaux :

- clé =N° de département => sous-tableau :

- Chaque sous- tableau est composé de 4 éléments : région, nom du département, et coordonnées (4 nombres) sur une carte de France

```
var $departement = array (  
    "01" => array ( "Rhône Alpes", "Ain", "236", "222", "255", "243" ),  
    "02" => array ( "Picardie", "Aisne", "192", "97", "209", "122" ),  
    "03" => array ( "Auvergne", "Allier", "176", "215", "201", "232" ),  
    etc.
```



# Les tableaux associatifs

---

## ■ Fonctions relatives : isset

- Pour tester l'existence d'un élément, on utilise la fonction `isset()`
- Exemple :

```
$calories["pommes"]= 300;  
$calories["banane"]= 130;  
$calories["litchi"]= 30;
```

```
if( isset( $calories["pommes"] ) ) {  
    echo "une pomme contient ", $calories["pommes"] ,  
    "caloriesn";  
}  
else{  
    echo "pas de calories définies pour la pomme n";  
}
```





# Les tableaux associatifs

---

## ■ Parcours :

- La méthode classique ne fonctionne pas. Il faut utiliser les fonctions : **foreach**, **list**...
- Exemple : `parcours-tableau-assoc.php`

```
<?php
    $tableau = array(
        0 => 'Chiffre 0',
        1 => 'Chiffre 1',
        2 => 'Chiffre 2'
    );
    foreach($tableau AS $cle => $valeur) {
        echo $cle.' | '.$valeur.'  

```

## ■ Autre exemple : array5.php

```
<HTML>
  <BODY>
    <TABLE BORDER>
      <?php
        $nom["Thècle"]="Assicmonpote";
        $nom["Édith"]="Avuleur";
        $nom["Kelly"]="Diocy";
        $nom["Mélusine"]="Enfaillite";
        $nom["Elmer"]="Hitmieux";
        foreach ($nom as $clef => $valeur) {
          echo "<TR><TD>$clef</TD><TD>$valeur</TD>
            </TR>\n";
        }
      ?>
    </TABLE>
  </BODY>
</HTML>
```

## ■ Parcours : parcous-tab-assoc3.php

```
<html>
  <body>
    <?php
      $annee_modif = "";
      $mois_modif = array("January" => "Janvier", "February" => "Février",
        "March" => "Mars", "April" => "Avril", "May" => "Mai",
        "June" => "Juin", "July" => "Juillet", "August" => "Août", "September" =>
        "Septembre", "October" => "Octobre", "November" => "Novembre",
        "December" => "Décembre");

      $date_modif = date( "d F Y", getlastmod());
      list($jour, $mois, $annee) = split( '[ ]', $date_modif);
      foreach($mois_modif as $cMMM => $MMM)
        if($cMMM == $mois) $mois = $MMM;
      $date_modif = "Dernière modification : $jour $mois $annee";
      echo $date_modif;
    ?>
  </body>
</html>
```



# Les tableaux associatifs

---

- Autre exemple : [parcours-tableau-assoc2.php](#)

- Si vous n'avez pas besoin de récupérer l'indice, vous pouvez également utiliser la fonction `foreach()` comme ceci:

```
<?php
    $tableau = array(0 => 'Chiffre 0',
                    1 => 'Chiffre 1',
                    2 => 'Chiffre 2');

    foreach($tableau AS $valeur)
    {
        echo $valeur.'<br />';
    }
?>
```



# Utilisation des tableaux

---

- Debogguer un tableau
  - PHP print\_r() et var\_dump()
    - Ces fonctions affichent la vraie structure du tableau : structure lisible)
  - Exemple : array6.php

```
<?php
    $colors = array('rouge', 'vert', 'bleu');
    echo '<pre>';
    print_r($colors);
    var_dump($colors);
    echo '</pre>';
?>
```



# Utilisation des tableaux

---

- **Debogguer un tableau**
  - résultat

- Avec `print_r` :

```
Array
(
    [0] => rouge
    [1] => vert
    [2] => bleu
)
```

- Avec `var_dump` :

```
array(3)
{
    [0]=> string(5) "rouge"
    [1]=> string(4) "vert"
    [2]=> string(4) "bleu"
}
```



# Utilisation des tableaux

---

## ■ Rechercher un élément

- Présence d'un élément :
  - `in_array(expression, tableau)`
- Exemple

```
<?php
    $colors = array('rouge', 'vert', 'bleu');
    if (in_array('vert', $colors)){
        echo '<br>Trouvé, vert';
    }
?>
```



# Utilisation des tableaux

---

- Calculer la clé :

- `Array_search(expression, tableau)`

- Exemple

```
<?php
```

```
$colors = array('rouge', 'vert', 'bleu');
```

```
$cle = array_search('vert', $colors);
```

```
echo "La valeur 'vert' est à la clé $cle";
```

```
//Affiche : la valeur 'vert' est à la clé 1
```

```
?>
```





# Utilisation des tableaux

---

## ■ Vérifier l'existence d'une clé

- `array_key_exists()`

### – Exemple

```
<?php
    $colors = array('ff0000' => 'rouge', '00ff00' => 'vert',
        '0000ff' => 'bleu');
    if(array_key_exists('00ff00', $colors)){
        echo 'La clé "00ff00" existe';
    }
?>
```



# Utilisation des tableaux

---

- Calculer le nombre d'occurrences d'un élément

- `array_count_values()`

- Exemple

```
<?php
```

```
$tab = array('Cyril', 'Christophe', 'Cyril', 'Thomas', 'Eric');
```

```
//tableau contenant les décomptes des éléments
```

```
$cpt = array_count_values($tab);
```

```
echo "L'élément 'Cyril' apparaît ", $cpt['Cyril'],  
"fois.<br>";
```

```
?>
```



# Utilisation des tableaux

---

## ■ Trier un tableau

– Par valeur : `sort()`

• Exemple :

```
<?php
    $stab = array('Cyril', 'Christophe', 'Benjamin', 'Thomas',
    'Eric');
    sort($stab);
    foreach($stab as $cle => $valeur){
        echo "$cle - $valeur, ";
    }
?>
```

➔ affiche : 0 - Benjamin, 1 - Christophe, 2 - Cyril, 3 - Eric, 4 - Thomas

– En ordre inverse

• `rsort()`



# Utilisation des tableaux

---

- Par la clé : ksort()

```
<?php
```

```
$tab = array('Cyril', 'Christophe', 'Benjamin' , 'Thomas',  
'Eric');
```

```
ksort($tab);
```

```
foreach($tab as $cle => $valeur){
```

```
    echo "$cle - $valeur, ";
```

```
}
```

- ➔ affiche : 0 - Cyril, 1 - Christophe, 2 - Benjamin, 3 - Thomas,  
4 - Eric,



# Utilisation des tableaux

---

## ■ Extraire et remplacer un élément

- Par utilisation de : list()
- Exemple

```
<?php
    $tab = array(1, 2, 3, 4);
    list($a, $b, $c, $d) = $tab;
    echo "$a-$b-$c-$d";
?>
```

➔ affiche : 1-2-3-4



# Utilisation des tableaux

---

## ■ Extraire des indices

- extract() permet de faire des clés des variables et de leur donner la valeur de leur indice

```
<?php
```

```
$tab = array('a'=>1, 'b'=> 2, 'c'=> 3, 'd'=> 4);
```

```
extract($tab);
```

```
echo "$a-$b-$c-$d";
```

```
?>
```

➔ affiche 1-2-3-4



# Utilisation des tableaux

---

- S rialiser un tableau
  - Il est possible de convertir un tableau en cha ne et inversement : implode() et explode()

- Exemple

```
<?php
$tab = array(1, 2, 3, 4);
$chaine = implode(*,$a, $tab);
echo $chaine
//Affiche 1*2*3*4
$tab=explode(*, $chaine);
//on r cup re le tableau d'origine
?>
```



# Utilisation des tableaux

---

- Extraire un sous-tableau

- `array_slice()`

- Exemple

```
<?php
```

```
$tab = array(1, 2, 3, 4,5,6,7);
```

```
$soustab = array_slice($tab,2,3)
```

```
echo implode(',',$soustab);
```

```
//Affiche 3,4,5
```

```
?>
```





# Utilisation des tableaux

---

## ■ Gérer les clés utilisées

– Liste des clés utilisées

- `array_keys()`

– Exemple

```
<?php
```

```
$tab = array('a'=>1, 'c'=> 5);
```

```
$cles = array_keys($tab);
```

```
echo implode('-', $cles);
```

```
//Affiche a-c
```

```
?>
```



# Utilisation des tableaux

## ■ Fusionner et séparer

- Fusion de plusieurs tableaux

- `array_merge()`

- Exemple

```
<?php
```

```
$result_2002 = array(12250, 12000, 21300, 25252, 20010, 8460);
```

```
$result_2003 = array(1520, 25000, 13530, 1052, 5010, 3680);
```

```
$result_2002_2003 = array_merge($result_2002, $result_2003);
```

```
print_r($result_2002_2003);
```

```
?>
```

- Affiche :

```
Array ( [0] => 12250 [1] => 12000 [2] => 21300 [3] => 25252 [4] => 20010 [5] => 8460 [6] => 1520 [7] => 25000 [8] => 13530 [9] => 1052 [10] => 5010 [11] => 3680 )
```



# Utilisation des tableaux

---

- Séparer

- `array_chunk($tab,n)`

- ❖ sépare `$tab` en tableaux de `n` éléments chacun

- Calculer des différences et des intersections

- Différence : `array_diff`

- Exemple

```
<?php
```

```
$tab1 = array(1, 2, 3, 4, 5, 6, 7);
```

```
$tab2 = array(1, 3, 5, 7);
```

```
$tab3 = array(1, 2, 3);
```

```
$diff = array_diff($tab1,$tab2,$tab3);
```

```
echo implode('-', $diff);
```

```
//Affiche 4-6
```

```
?>
```



# Utilisation des tableaux

---

- intersection : `array_intersect()`
- Exemple

```
<?php
```

```
$tab1 = array(1, 2, 3, 4, 5, 6, 7);
```

```
$tab2 = array(1, 3, 5, 7);
```

```
$tab3 = array(1, 2, 3);
```

```
$inter = array_intersect($tab1,$tab2,$tab3);
```

```
echo implode('-', $diff);
```

```
//Affiche 1-3
```

```
?>
```



# Utilisation des tableaux

---

- Enlever les doublons : array\_unique()
- Exemple

```
<?php
```

```
$tab = array(1, 2, 3, 4, 5, 2, 4);
```

```
$tab = array_unique($tab);
```

```
echo implode('-', $diff);
```

```
//Affiche 1-2-3-4-5
```

```
?>
```



# Utilisation des tableaux

---

- Gérer des piles et des files
  - Fonctions : array\_push et array\_pop

- array\_push

```
<?php
```

```
$tab = array();
```

```
array_push($tab, 1, 3, 5);
```

```
/*équivalent à */
```

```
$tab = array();
```

```
$tab[1]=1;
```

```
$tab[2]=3;
```

```
$tab[3]=5;
```

```
?>
```



# Utilisation des tableaux

---

– Fonctions : array\_push et array\_pop

- array\_pop

```
<?php
```

```
$tab = array();
```

```
array_push($tab, 1, 3, 5);
```

```
echo array_pop($tab); //Affiche 5;
```

```
echo array_pop($tab); //Affiche 3;
```

```
?>
```



# PHP 5

---

## Les formulaires





# Les formulaires

---

## ■ Intérêt

- Dans un contexte Web, les données échangées avec le système se font à travers des formulaires
- Les formulaires HTML sont la méthode la plus simple pour avoir des interactions avancées avec l'utilisateur
- Ils permettent, par exemple, de :
  - Créer un espace sécurisé
  - Donner aux clients la possibilité de modifier eux-mêmes leurs sites
  - Interagir avec le visiteur en lui demandant des informations complémentaires...



# Les formulaires

---

- Création : balise `<form>`

```
<form action='reception_formulaire.php' method='GET ou  
POST '>
```

```
<!-- différents champs -->
```

```
</form>
```

- action :
  - désigne la page vers laquelle seront envoyées les informations rentrées par l'utilisateur une fois le bouton d'envoi actionné
- method
  - définit le mode d'envoi des informations au serveur
  - Deux méthodes existent
    - ❖ GET et POST
  - PHP associe à ces deux variables deux tableaux `$_GET` et `$_POST` pour récupérer les données passées



# Les formulaires

---

- Création : méthode POST

- Exemple

```
<form action="valider.php" method="post">
  <select name="objet">
  ...
  <input name="genre" type="text" />
  ...
```

- Ce code HTML spécifie que les données du formulaire seront soumises à la page web "valider.php" en utilisant la **méthode POST**
    - Prenez soin de noter les noms (après **name**) des données du formulaire, car ils représentent les "**clés**" dans le tableau associatif "\$\_POST"
    - Exemple : **\$\_POST['genre']** permettra de récupérer genre
    - **Ces variables seront cachées pendant l'envoi**



# Les formulaires

---

- **Création : méthode GET (transmission par URL)**
  - La différence avec la méthode POST est qu'elle passe les variables à la page web "valider.php" en les ajoutant à la fin de l'URL
  - Après avoir cliqué soumettre, l'URL aura ceci ajouté à la fin :
    - "valider.php?objet=xxx&genre=xxx"
  - Le point d'interrogation "?" dit au navigateur que les **objets suivants sont des variables**
  - **On les récupérera en utilisant le tableau \$\_GET[]**
  - **Dans ce mode de transmission, les variables sont apparentes**
  - **Pour votre utilisation c'est égal**



# Les formulaires

## Les éléments du formulaire

---

### ■ Champ de texte

```
<input type="text" name="pseudo" value="M@teo21" />
```

#### – 2 attributs

- **name** : c'est le nom du champ de texte

- ❖ Choisissez-le bien, car c'est lui qui va produire une variable

- `$_POST['pseudo']`

- **value** : permet d'affecter une valeur à la zone de texte

- ❖ Par défaut, le champ de texte est vide

# Les formulaires

## Les éléments du formulaire

- Champ de texte : tester l'exemple : form-texte1.php

```
<?php
  if (isset($_POST['mon_champ'])) {
?>
  Votre champ contenait :
  <b> <?php echo $_POST['mon_champ']; ?></b>
  <br/><br/>
<?php
}
?>
<form method="POST">
  <input name="mon_champ" type="text"/>
  <input name="valider" type="submit" value="OK"/>
</form>
```

- La fonction isset() permet de savoir si la variable `$_POST['mon_champ']` existe

## ■ Champ de texte : autre manière : form-texte2.php

```
<?php
$mon_champ = isset($_POST['mon_champ']) ?
    $_POST['mon_champ'] : "";
    // ... dont la forme équivalente avec des if/else est :
    //if (isset($_POST['mon_champ'])) {
    //    $mon_champ = $_POST['mon_champ'];
    //} else {
    //    $mon_champ = "";
    //}
if ($mon_champ) { //ici, on a créé une variable pour le test
?>
    Votre champ contenait :
    <b><?php echo $mon_champ; ?></b>
    <br/><br/>
<?php
}
?>
<form method="POST">
    <input name="mon_champ" type="text" value=" <?php echo
    $mon_champ; ?>"/>
    <input type="submit" value="OK"/>
</form>
```



# Les formulaires

## Les éléments du formulaire

---

### ■ Zone de texte : textarea

```
<textarea name="message" rows="8" cols="45">
```

Votre message ici.

```
</textarea>
```

### – Remarques

- rows resp. cols détermine le nombre de lignes resp. le nombre de colonnes de la zone de texte
- Il n'y a pas d'attribut **value**
- En fait, le texte par défaut est ici écrit entre le `<textarea>` et le `</textarea>`



# Les formulaires

## Les éléments du formulaire

### ■ Exemple 1 : form-textarea1.php

- Afficher le contenu de la zone de texte si celle-ci n'est pas vide

```
<?php
$mon_champ = isset($_POST['mon_champ']) ?
    $_POST['mon_champ'] : '';
```

```
if ($mon_champ) {
?>
    Votre champ contenait :
    <b><?php echo $mon_champ; ?></b>
    <br/><br/>
```

```
<?php
}
?>
```

```
<form method="POST">
    <textarea name="mon_champ"></textarea>
    <input type="submit" value="OK"/>
</form>
```

# Les formulaires

## Les éléments du formulaire

### ■ Exemple 2 : form-textarea2.php

- Un peu plus compliqué maintenant, car nous allons réafficher dans le même contrôle, son contenu :

```
<?php
    $mon_champ = isset($_POST['mon_champ']) ?
    $_POST['mon_champ'] : "";
    if ($mon_champ) {
?>
    Votre champ contenait :
    <b><?php echo $mon_champ; ?></b>
    <br/><br/>
<?php
}
?>
<form method="POST">
    <textarea name="mon_champ">
        <?php echo $mon_champ; ?>
    </textarea>
    <input type="submit" value="OK"/>
</form>
```



# Les formulaires

## Les éléments du formulaire

---

### – Attention

- Lorsque vous affichez quelque chose dans un textarea, faites attention de bien coller la balise de début et de fin du textarea, car dans certains cas vous aurez par exemple dans votre zone de texte soit des espaces ou un saut de ligne qui traînent
- Vous pouvez dans ce cas utiliser la fonction [trim\(\)](#)

# Les formulaires

## Les éléments du formulaire

### ■ Les boutons d'options (radio)

- Permettent de faire des choix

Aimez-vous les frites ?

```
<input type="radio" name="frites" value="oui"
checked="checked" /> Oui
```

```
<input type="radio" name="frites" value="non" /> Non
```

- On utilise le même nom pour la gestion de tous les boutons, ici :  
« frites »
- C'est donc cette variable qui contiendra ce que l'utilisateur a choisi
- Dans la page cible, une variable `$_POST['frites']` sera créée
- Elle aura la valeur du bouton d'option choisi par le visiteur
- Si on aime les frites, alors on aura `$_POST['frites'] = 'oui'`
- Il faut bien penser à remplir l'attribut "value" du bouton d'option car c'est lui qui va déterminer la valeur de la variable

# Les formulaires

## Les éléments du formulaire

- Exemple : form-bouton-option1.php

```
<?php
    $mon_champ = isset($_POST['mon_champ']) ? $_POST['mon_champ'] : "";
    if ($mon_champ) {
?>
    Vous avez choisi :
    <b><?php echo $mon_champ; ?></b>
    <br/><br/>
<?php
}
?>
//on utilise la même variable mon_champ
<form method="POST">
    <input type="radio" name="mon_champ" value="Option 1"/>Option 1<br/>
    <input type="radio" name="mon_champ" value="Option 2"/>Option 2<br/>
    <input type="radio" name="mon_champ" value="Option 3"/>Option 3<br/>
    <input type="submit" value="OK"/>
</form>
```

## ■ Exemple 2 : form-bouton-option2.php

- Nous allons maintenant faire la même chose, mais l'option qui a été choisie sera sélectionnée à nouveau

```
<?php
$mon_champ = isset($_POST['mon_champ']) ? $_POST['mon_champ'] : "";
if ($mon_champ) {
?>
    Vous avez choisi :
    <b><?php echo $mon_champ; ?></b>
    <br/><br/>
<?php
}
?>
<form method="POST">
    <input type="radio" name="mon_champ" value="Option 1" <?php
    if($mon_champ == "Option 1") { echo 'checked'; } ?> />Option 1<br/>
    <input type="radio" name="mon_champ" value="Option 2" <?php
    if($mon_champ == "Option 2") { echo 'checked'; } ?> />Option 2<br/>
    <input type="radio" name="mon_champ" value="Option 3" <?php
    if($mon_champ == "Option 3") { echo 'checked'; } ?> />Option 3<br/>
    <input type="submit" value="OK"/>
</form>
```

## ■ Exemple 3 : form-bouton-option3.php

- On utilise un tableau pour simplifier le code. Dans ce cas, il suffit d'utiliser une boucle **foreach** pour faire une unique condition qui permettrait de re-sélectionner celle que l'utilisateur a choisie car ce code devient vite illisible, lorsque les options se font plus nombreuses :

```
<?php
$mon_champ = isset($_POST['mon_champ']) ? $_POST['mon_champ'] : "";
// Les options du formulaire
$options = array(
    'Option 1',
    'Option 2',
    'Option 3');
if($mon_champ) {
?>
    Vous avez choisi :
    <b><?php echo $mon_champ; ?></b>
    <br/><br/>
<?php}?>
<form method="POST">
    <?php
    foreach ($options as $v) {
        if ($v == $mon_champ) {
            echo '<input type="radio" name="mon_champ" value="" . $v . "' checked/>' . $v .
            '<br/>';}
            else {echo '<input type="radio" name="mon_champ" value="" . $v . "'/>' . $v . '<br/>';}
        }
    }
?>
    <input type="submit" value="OK"/>
</form>
```

# Les formulaires

## Les éléments du formulaire

### ■ Les cases à cocher

- Ici, on fait appel à un tableau pour retenir toutes les cases cochées
- Exemple : form\_case-a-cocher1.php

```
<?php
if (isset($_POST['mon_champ'])) {
    echo "Vous avez choisi :";
    for ($i = 0, $c = count($_POST['mon_champ']); $i < $c; $i++) {
        echo "<br/><b>" . $_POST['mon_champ'][$i] . "</b>";
    }
}
?>
<form method="POST">
    <input type="checkbox" name="mon_champ[]" value="Option
    1"/>Option 1<br>
    <input type="checkbox" name="mon_champ[]" value="Option
    2"/>Option 2<br>
    <input type="checkbox" name="mon_champ[]" value="Option
    3"/>Option 3<br>
    <input type="submit" value="OK">
</form>
```





# Les formulaires

## Les éléments du formulaire

---

### ■ Commentaire

- La variable **`$_POST['mon_champ']`** est un tableau qui contient les valeurs que l'utilisateur a cochées
- Pour signifier que les cases cochées seront des éléments d'un tableau, nous avons donc ajouté des crochets



# Les formulaires

## Les éléments du formulaire

---

### ■ Cases à cocher (suite)

- Dans l'exemple suivant, nous allons réafficher dans le formulaire les cases sélectionnées par l'utilisateur
- Encore une fois, nous pourrions utiliser un autre tableau qui contiendrait toutes les cases du formulaire, ce qui nous permettrait d'afficher la case cochée ou non dans une simple boucle foreach mais nous allons rester au plus simple pour que tout ceci soit plus clair !

```

<?php
if (isset($_POST['mon_champ'])) {
    echo "Vous avez choisi :";
    for ($i = 0, $c = count($_POST['mon_champ']); $i < $c; $i++) {
        echo "<br/><b>" . $_POST['mon_champ'][$i] . "</b>";
    }
}
// Renvoie vrai si $option fait partie du résultat
function est_selectionne($option) {
    if (isset($_POST['mon_champ'])) {
        return FALSE;
    }
    for ($i = 0, $c = count($_POST['mon_champ']); $i < $c; $i++) {
        if ($_POST['mon_champ'][$i] == $option) {
            return TRUE;
        }
    }
    return FALSE;
}
?>
<form method="POST">
    <input type="checkbox" name="mon_champ[]" value="Option 1" <?php
        if(est_selectionne("Option 1")) { echo 'checked'; } ?>/>Option 1<br/>
    <input type="checkbox" name="mon_champ[]" value="Option 2" <?php
        if(est_selectionne("Option 2")) { echo 'checked'; } ?>/>Option 2<br/>
    <input type="checkbox" name="mon_champ[]" value="Option 3" <?php
        if(est_selectionne("Option 3")) { echo 'checked'; } ?>/>Option 3<br/>
    <input type="submit" value="OK"/>
</form>

```



# Les formulaires

## Les éléments du formulaire

---

### ■ Les listes déroulantes à sélection simple

- Les listes déroulantes sont couramment utilisées pour que l'utilisateur ne puisse sélectionner qu'une valeur

```
<select name="mon_champ">  
  <option>Option 1</option>  
  <option>Option 2</option>  
  <option>Option 3</option>  
</select>
```

# Les formulaires

## Les éléments du formulaire

- Exemple : form-liste-deroulante1.php

```
<?php
$mon_champ = isset($_POST['mon_champ']) ? $_POST['mon_champ'] : "";
if ($mon_champ) {
?>
    Votre champ contenait :
    <b><?php echo $mon_champ; ?></b>
    <br/><br/>
<?php
}
?>
<form method="POST">
    <select name="mon_champ">
        <option>Option 1</option>
        <option>Option 2</option>
        <option>Option 3</option>
    </select>
    <input type="submit" value="OK"/>
</form>
```

## ■ Exemple : form-liste-deroulante2.php

- Réaffichons maintenant l'option sélectionnée par l'utilisateur dans la liste déroulante :

```
<?php
$mon_champ = isset($_POST['mon_champ']) ? $_POST['mon_champ'] : "";
if ($mon_champ) {
?>
    Votre champ contenait :
    <b><?php echo $mon_champ; ?></b>
    <br/><br/>
<?php
}
?>
<form method="POST">
    <select name="mon_champ">
        <option <?php if($mon_champ == "Option 1") { echo 'selected'; } ?>>Option
        1</option>
        <option <?php if($mon_champ == "Option 2") { echo 'selected'; } ?>>Option
        2</option>
        <option <?php if($mon_champ == "Option 3") { echo 'selected'; } ?>>Option
        3</option>
    </select>
    <input type="submit" value="OK"/>
</form>
```



# Les formulaires

## Les éléments du formulaire

---

### ■ Liste déroulante à sélection multiple

- Les listes déroulantes peuvent également être utilisées pour permettre de sélectionner plusieurs valeurs (voire une ou pas du tout)
- Ce cas est strictement identique aux cases à cocher puisque l'on récupérera au final une variable "tableau" qui hébergera le résultat
- Voyons un premier exemple avec un simple réaffichage des valeurs sélectionnées par l'utilisateur :

## ■ Exemple 1 : form-liste-deroulante3.php

```
<?php
// Nos options définies dans un tableau (plus facile à coder et à maintenir)
$options = array(
    'Option 1',
    'Option 2',
    'Option 3',
    'Option 4'
);
// Affichage de la sélection seulement si le formulaire a été validé
if (isset($_POST['mon_champ'])) {
    echo "Vous avez choisi :";
    for ($i = 0, $c = count($_POST['mon_champ']); $i < $c; $i++) {
        echo '<br/><b>' . $_POST['mon_champ'][$i] . '</b>';
    }
}
?>
<form method="POST">
    <select name="mon_champ[]" size="4" multiple>
        <?php
        foreach ($options as $k) {
            echo '<option>' . $k . '</option>';
        }
        ?>
    </select>
    <input type="submit" value="OK"/>
</form>
```



## ■ Exemple 2 : form-liste-deroulante4.php

- Modifions maintenant l'exemple précédent afin de réafficher ce que l'utilisateur a sélectionné :

```
<?php
// Nos options définies dans un tableau (plus facile à coder et à maintenir)
$options = array(
    'Option 1',
    'Option 2',
    'Option 3',
    'Option 4'
);
// Affichage de la sélection seulement si le formulaire a été validé
if (isset($_POST['mon_champ'])) {
    echo "Vous avez choisi :";
    for ($i = 0, $c = count($_POST['mon_champ']); $i < $c; $i++) {
        echo '<br/><b>' . $_POST['mon_champ'][$i] . '</b>';
    }
}
function est_selectionne($option) {
    for ($i = 0, $c = count($_POST['mon_champ']); $i < $c; $i++) {
        if ($_POST['mon_champ'][$i] == $option) {
            return TRUE;
        }
    }
    return FALSE;
}
?>
```

```
<form method="POST">
  <select name="mon_champ[]" size="4" multiple>
    <?php
      foreach ($options as $k) {
        if (isset($_POST['mon_champ']) && est_selectionne($k)) {
          echo '<option selected>' . $k . '</option>';
        } else {
          echo '<option>' . $k . '</option>';
        }
      }
    ?>
  </select>
  <input type="submit" value="OK"/>
</form>
```



# Les formulaires

## Les éléments du formulaire

---

### ■ Les champs cachés

#### - En quoi ça consiste ?

- C'est un code dans votre formulaire qui n'apparaîtra pas aux yeux du visiteur, mais qui va quand même créer une variable avec une valeur
- Supposons que vous ayez besoin de "retenir" que le pseudo du visiteur est "Mateo21"
- Vous allez taper ce code :
  - ❖ Code : `HTML1<input type="hidden" name="pseudo" value="Mateo21" />`
- A l'écran, vous ne verrez rien
- Mais dans la page cible, une variable `$_POST['pseudo']` sera créée (correspondant à name), et elle aura la valeur "Mateo21" (correspondant à value) !
- C'est apparemment inutile, mais vous verrez que lorsque vous commencerez à créer des formulaires vous en aurez vite besoin

# Les formulaires

## Les éléments du formulaire

### ■ Exemple

- Voici un exemple en HTML :

```
<form action="traitement.php" method="post">
  <p><input type="hidden" name="champ_cache" value="ici ce
    que vous voulez" />Oui</p>
  <p><input type="submit" value="Envoyer" /></p>
</form>
```

- Et voici comment on récupère la valeur de ce champ caché en PHP :

```
<?php
  if(isset($_POST['champ_cache']))
  {
    echo htmlentities($_POST['champ_cache']); //affiche le
    contenu du champ
  }
?>
```