

# PHP: Tableaux Complexes et formulaires



# Sommaire



- Les tableaux multidimensionnels.
- Les formulaires HTML.
- Les tableaux \$\_GET et \$\_POST.
- Fonctions de chaînes de caractères.
- Validation de données.
- Sécurité et intégrité des données.

# Les tableaux multidimensionnels



➤ Un tableau à deux dimensions est un tableau contenu dans un tableau.

```
<?php
    $t2Dim = array( array('Buire', '880'),
                    array('Hirson', '9348'),
                    array('Reims', '181893'),
                    array('Paris', '2240621'));

echo $t2dim[0][0]; // affichage du premier élément du premier tableau
$iNbEnreg = count($t2Dim);
for($i=0; $i<$iNbEnreg; $i++)
{
    foreach($t2Dim[$i] as $indice=>$valeur)
    {
        echo $valeur.' ';
        if($indice%2 == 1) echo "<br/>";
    }
}
?>
```

# Les tableaux multidimensionnels



➤ On peut aussi voir des tableaux associatifs.

```
<?php
    $t2Dim = array( array('ville'=>'Buire', 'population'=>'880'),
                    array('ville'=>'Hirson', 'population'=>'9348'),
                    array('ville'=>'Reims', 'population'=>'181893'),
                    array('ville'=>'Paris', 'population'=>'2240621')

                );
echo $t2dim[1]['ville'];
$iNbEnreg = count($t2Dim);
for($i=0; $i<$iNbEnreg; $i++)
{
    foreach($t2Dim[$i] as $indice=>$valeur)
    {
        echo '<strong>'.$indice.'</strong>: '.$valeur.'<br/>';
    }
}
?>
```

# Exercice pratique:

Afficher dans un tableau HTML un ensemble d'éléments contenus dans un tableau à deux dimensions.



# Les formulaires HTML



- Un formulaire permet de demander des informations à un internaute.
- Ces informations doivent être contrôlées côté navigateur et côté serveur pour plus de sécurité.
- Au clic sur le bouton "submit" les données saisies sont envoyées au serveur en fonction du paramétrage du formulaire.

# Les formulaires HTML



```
<form action="traitement.php" method="post">
```

- `traitement.php` est la page qui recevra le contenu du formulaire et qui le traitera.
- `method="get"` signifie que le contenu du formulaire sera attaché à l'URL de la page de traitement du formulaire (256 caract. max).
- `method="post"` signifie que le contenu du formulaire sera encapsulé dans les entêtes envoyées à la page de traitement. Cette méthode est beaucoup plus sûre et évite d'envoyer des mots de passe en clair.

```
</form>
```

# Les formulaires HTML



```
<input type="" value="" name="">
```

## ➤type:

- text pour les champs texte à remplir
  - password pour les mots de passes
  - radio pour les boutons radio (même attribut name)
  - checkbox pour les cases à cocher
  - hidden pour les champs cachés
  - submit & reset pour les boutons
- value permet d'initialiser une valeur par défaut
- name correspond au nom de la variable qui va être envoyée au script.



# Les formulaires HTML



## ➤ Les listes déroulantes:

```
<select name="test">  
  <option value="12">12</option>  
  <option value="10">10</option>  
</select>
```

## ➤ Les boites de texte:

```
<textarea name="texte" cols="30"  
  rows="5"></textarea>
```

# Les formulaires HTML



`<label for="nomchamps">` Permet de mettre un intitulé pour un champs `<input>`

`<fieldset>` Permet de grouper des éléments de formulaire

`<legend>` Permet de donner un nom à un groupe de champs

# Les tableaux \$\_GET et \$\_POST



- Si la méthode "get" a été sélectionnée, le serveur mettra à disposition du développeur un tableau \$\_GET contenant les couples name/valeur saisies.

```
<?php  
    print_r($_GET);  
?>
```

- Si la méthode "post" a été sélectionnée, le serveur mettra à disposition du développeur un tableau \$\_POST contenant les couples name/valeur saisies.

```
<?php  
    print_r($_POST);  
?>
```

# Exercice pratique

Créer un formulaire avec tous les types de champs et regarder côté PHP comment le tableau est rempli par le serveur.



# Les chaines de caractères



- La fonction `strlen()` renvoie la taille d'une chaine de caractères.
- La fonction `strtoupper()` retourne la chaine de caractères en majuscules.
- La fonction `strtolower()` retourne la chaine de caractères en minuscules.
- La fonction `trim()` enlève les espaces à gauche et à droite d'une chaine de caractères. Il existe des fonctions `ltrim()` et `rtrim()` pour n'enlever les espaces que d'un côté.
- La fonction `ucfirst()` met le premier caractère d'une chaine en majuscule.
- La fonction `ucwords()` met le premier caractère de chaque mot d'une chaine en majuscule.

# Les chaines de caractères



- La fonction `substr($string, $offset, $longueur)` extrait un segment de chaine.
- La fonction `explode()` transforme une chaine en tableau selon le délimiteur choisi.
- La fonction `implode()` transforme un tableau en chaine de caractère selon le délimiteur choisi.

# Les chaînes de caractères



```
<?php
// Position de 'practices' dans 'Best practices PHP5';
echo strpos('Best practices PHP5', 'practices') . "\n";
// Vérifier que la chaîne 'Best practices' contient bien
// 'Best'. Attention : utiliser !== et === plutôt que != et
// ==!
if (strpos('Best practices', 'Best') !== false) {
echo "La chaîne 'Best practices' contient bien 'Best'";
}
// Fonctions similaires : // strrpos(), stripos(), strstr()
// : variantes de strpos() // strrchr() : trouve la dernière
// occurrence d'un caractère // strstr() et strpos() : trouve
// la 1ère occurrence d'une chaîne
?>
```

# Les chaines de caractères



```
<?php
    $phrase = 'Bienvenue à Guillaume et à tous les guillaumes
    !';

    // Remplacement de Guillaume par Gérald, sans respecter la
    casse (i)
    echo str_ireplace('Guillaume', 'Gérald', $phrase);

    // Remplacement de tous les espaces par des _, des o par des
    0, des i par des l et des l par des 7
    echo strtr($phrase, ' oil', '_0l7');
    // Ceci affiche :
    //Bienvenue_à_Gu177aume_et_à_t0us_7es_gu177aumes_!
?>
```



# Les chaines de caractères



```
<?php
// Affiche "Les vacanciers sont bronzés."
$chaine = "Les vacances sont magnifiques.";

// Les valeurs de remplacements
$to = array('vacances', 'magnifique');
$from = array('vacanciers', 'bronzé');

// Remplacement
echo str_replace($to, $from, $chaine);
?>
```

# Les chaines de caractères



```
<?php
$phrase = "Vous devez payer 23489 euros";
// Extraction de 23489
echo substr($phrase, strpos($phrase, ' euros') -5, 5);
// Extraction de 23489 en partant de la fin
echo substr($phrase, -11, 5);
// Remplacement de 23489 par 0
echo substr_replace($phrase, '0', -11, 5);
// 23489

// Vous devez payer 0 euros
?>
```

# Exercices pratiques

reprendre le formulaire précédent et  
afficher les valeurs saisies après nettoyage.



# Validation des données



➤ **Rappel: certaines fonctions permettent de vérifier si la donnée saisie est valide ou non:**

```
<?php
    if(is_numeric($_POST['maVar'])) echo 'maVar est numérique';
    if(is_string($_POST['maVar'])) echo 'maVar est une chaine';
    if(is_bool($_POST['maVar'])) echo 'maVar est booléenne';
    if(is_array($_POST['maVar'])) echo 'maVar est un tableau';
    if(strlen($_POST['mdp'])<8) echo 'Mot de passe trop court';
    if(!isset($_POST['maVar'])) echo 'maVar est non définie';
?>
```

# Exercices pratiques

Contrôler la validité des données d'un formulaire



# Sécurité et intégrité



- La fonction `addslashes()` ajoute des antislash dans une chaîne pour améliorer la sécurité.
- La fonction `stripslashes()` enlève les antislash dans une chaîne.
- La fonction `htmlentities()` convertit les caractères en entités HTML.
- La fonction `htmlspecialchars()` convertit les caractères spéciaux en entités HTML.
- La fonction `strip_tags()` supprime les balises HTML et PHP d'une chaîne.

# Exercices pratiques

