

# Les formulaires HTML dans PHP

Le formulaire ne peut se concevoir sans PHP de plus il a un comportement particulier. Lors de la soumission la suite est obligatoirement renvoyée vers une nouvelle page. Cette nouvelle page peut être effectivement nouvelle mais elle peut aussi être la page même du formulaire.

## Table des matières

HTML et PHP.....	1	Naissance de la réponse.....	2
Ecriture du formulaire.....	1	Détection de l'existence d'une variable.....	2
Exécution après soumission.....	1	Fonctionnement.....	2
Traitement de plusieurs points.....	2	La perte des variables.....	2
La soumission.....	2		

## HTML et PHP

Dans un script php nous passons aisément du php au html et vice versa.

Exemple : Les trois formes sont acceptables et acceptées.

```
<form method='post' action='<?php echo $pag; ?>'> // avec $pag='cettepage.php';
```

ou

```
<form method='post' action='cettepage.php'>
```

ou encore

```
echo "<form method='post' action=\"$pag'>";
```

\$pag représente la page qui sera chargée après la soumission. Cela peut être la même page ou une autre page. Il faut se rendre compte quel que soit le cas, la page est relue ce qui entraîne la perte des variables.

## Ecriture du formulaire

L'écriture du formulaire ne pose pas de problème. Par contre il faut bien comprendre ce qu'il se passe après la soumission.

## Exécution après soumission

Le formulaire sera traité à la page indiquée. C'est simple mais pas satisfaisant. Un formulaire peut avoir à répondre à plusieurs questions qui peuvent conduire à des traitements différents. Par exemple, nous avons défini des champs d'une table. L'envoi peut concerner :

- un nouvel enregistrement,
- la mise à jour d'un enregistrement,

- la suppression de l'enregistrement.

Donc avec le même questionnaire on peut avoir trois traitements différents. Comment résoudre le problème ?

## Traitement de plusieurs points

### La soumission

C'est simple la soumission comporte un identifiant. Il faut savoir que cette identifiant n'existe que si l'opérateur à cliquer sur le choix concerné. Nous avons :

```
<input type='submit' name='enregistre' value='Enregister'>  
<input type='submit' name='modif' value='Modifier'>  
<input type='submit' name='supprime' value='Supprimer'>
```

Nous avons donc 3 valeurs possibles '**enregistre**' , '**modif**' , '**supprime**', une par bouton

### Naissance de la réponse

Seule la variable du bouton appuyé existera dans la réponse.

Appuyons sur "**Enregistrer**", la variable de réponse qui est `$_POST['enregistre']` va exister. les valeurs `$_POST['modif']` et `$_POST['supprime']` n'existeront pas.

### Détection de l'existence d'une variable

Or nous avons une instruction qui nous permet de savoir si une variable existe :

```
if(isset($_POST['enregistre'])){  
    le programme passera ici si et uniquement si, le bouton "Enregistrer" a été pressé.  
}
```

Les autres conditions seront ignorées. Dans ce cas, le renvoi sur la même page est possible puisque nous savons isoler les réponses en fonction du bouton appuyé. Cela simplifie, nous aurons un seul script au lieu de plusieurs.

## Fonctionnement

Lors du premier passage sur le script toutes les sections de condition sont ignorées puisqu'il n'y a pas encore eu d'appui sur l'un des boutons. Lors d'un appuis seule la section concernée sera exécutée. Ici nous avons une solution satisfaisante.

## La perte des variables

Que la soumission renvoie sur la même page ou une page différente, le résultat est le même il y a changement de page donc perte des données. Dans le cas où cela est fâcheux se rappeler que la variable de session existe.