

# Les formulaires

---

## 1. Introduction

Un formulaire HTML est une page WEB qui offre au client (ici l'utilisateur) de renvoyer des données (informations saisies par ses soins) au serveur aux fins de traitements et/ou enregistrement. A charge pour le programmeur côté serveur d'évaluer, traiter et mettre en forme ses données qui lui parviennent au format http.

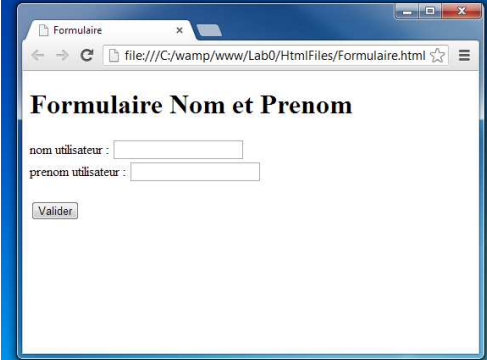
Image

Lors de la soumission du formulaire, les données saisies par l'utilisateur seront envoyées par le réseau (souvent le web) vers un serveur qui exécutera un script destiné à les traiter. Ce script qui s'exécute côté serveur est un programme écrit en PHP, ASP, Perl, Java, Python ...

## 2. Constitution d'un formulaire

Un formulaire est constitué d'un ensemble de composants (champs) HTML inclus dans une balise `<form></form>`

Exemple de formulaire (voir ci-contre):



The screenshot shows a web browser window with the title 'Formulaire'. The address bar displays 'file:///C:/wamp/www/Lab0/HtmlFiles/Formulaire.html'. The main content area has the heading 'Formulaire Nom et Prenom'. Below the heading, there are two text input fields. The first is labeled 'nom utilisateur :' and the second is labeled 'prenom utilisateur :'. At the bottom of the form, there is a button labeled 'Valider'.

Code de cette page HTML

```
<!DOCTYPE html>
<html>
<head>
  <title> Formulaire </title>
</head>
<body>
<h1> Formulaire Nom et Prenom</h1>
<form action="http://www.monsite.com/traitement1.php" method="get">

  nom utilisateur : <input name="username" type="text"> <br/>
  prenom utilisateur : <input name="userfirstname" type="text"> <br/><br/>
  <input type="submit" value="Valider">
</form>

</body>
</html>
```

### 3. Les composants HTML d'un formulaire

#### 3.1. Les balises INPUT

##### 3.1.1. INPUT TEXT

utilisateur :

```
<input name="username" type="text"/>
```

- Nous pouvons spécifier la taille du champs (attribut) ou la longueur maximale de la saisie en caractères (attribut maxlength) :

```
<input name="username" type="text" size = "50" maxlength="10"/>
```

- Un input text peut être protégé en écriture

```
<input name="username" type="text" maxlength="10" readonly="true"/>
```

- Il peut être destiné à la saisie d'un mot de passe (affiche des pastilles en lieu et place des caractères saisis) :

```
<input name="username" type="password" size = "50" maxlength="10"/>
```

### 3.1.2. La balise TEXT AREA (Champ texte multiligne)

```
<textarea cols="50" rows="10"></textarea>
```

- Les attributs cols et rows spécifient la taille du champs en colonnes / lignes.

### 3.1.3. La balise SELECT (liste de sélection)

```
<select name = "choixPays">
  <option value="1">Canada</option>
  <option value="2">USA</option>
  <option value="3">Mexique</option>
  <option value="4">Bresil</option>
</select>
```



Pour présélectionner une des options :

```
<option value="1" selected = "true">Canada</option>
```

### 3.1.4. La balise SELECT multiple (liste à choix multiple)

```
<select name = "voiture[]" multiple="true" size="4">
  <option value="1" selected = "true">Lamborghini</option>
  <option value="2">Ferrari</option>
  <option value="3">Porsche</option>
  <option value="4">Lexus</option>
  <option value="4">Mercedes</option>
  <option value="4">Bentley</option>
</select>
```



Selectionnez une marque

Lamborghini : ☒ Ferrari : ☒ Porsche : ☐ Lexus : ☐

Valider

### 3.1.5. INPUT CHECKBOX

Choix multiple

```
<h3> Selectionnez une marque </h3>
Lamborghini : <input type="checkbox" name = "voiture[]" value="1"/>
Ferrari : <input type="checkbox" name = "voiture[]" value="2"/>
Porsche : <input type="checkbox" name = "voiture[]" value="3"/>
Lexus : <input type="checkbox" name = "voiture[]" value="4"/>
```

### Selectionnez une marque

Lamborghini : ☐ Ferrari : ☒ Porsche : ☐ Lexus : ☐

Valider

### 3.1.6. INPUT RADIO (boutons radio)

Comme les CheckBox mais a un choix unique

```
<h3> Selectionnez une marque </h3>
Lamborghini : <input type="radio" name = "voiture[]" value="1"/>
Ferrari : <input type="radio" name = "voiture[]" value="2"/>
Porsche : <input type="radio" name = "voiture[]" value="3"/>
Lexus : <input type="radio" name = "voiture[]" value="4"/>
```

Pour des boutons radio non mutuellement exclusifs (sélection multiple) :

```
<h3> Selectionnez une marque </h3>
Lamborghini : <input type="radio" name = "voiture0" value="1"/>
Ferrari : <input type="radio" name = "voiture1" value="2"/>
Porsche : <input type="radio" name = "voiture2" value="3"/>
Lexus : <input type="radio" name = "voiture3" value="4"/>
```

### 3.1.7. INPUT BUTTON

Effacer le formulaire

Valider

Pour soumettre ou effacer (remettre a son état par défaut) le formulaire

```
<input type="reset" value="Effacer le formulaire"/>
<input type="submit" value="Valider"/>
```

Pour déclencher une autre action que submit ou reset

```
<input type="button" value="Cliquez ici "/>
```

### 3.1.8. Bouton sous forme d'image

```
<input type="image" src="nom_image.png"/>
```

### INPUT HIDDEN

Champs invisible destiné a stocker une information ou une donnée a envoyer au serveur.

```
<input type="hidden" name="cd_index" val=="1"/>
```

## 4. Ce qui se passe lors de la soumission d'un formulaire

La balise <form></form> est munie de deux attributs : action et method.

L'attribut method (parfois appelée 'verbe' dans la littérature) est celui qui spécifie la façon dont seront expédiées les données du formulaire.

Le protocole http reconnaît deux méthodes : GET et POST.

### 4.1. Méthode GET

Les données du formulaire seront envoyées en clair par l'URL (paramètres d'URL). Elles seront concaténées et dans l'entête à la suite de l'adresse comme dans :

```
http://www.monsite.com/traitement1.php?username=LAPOINTE&userfirstname=Albert&voiture%5B%5D=1&voiture1=2
```

- Ces paramètres d'URL sont visibles dans la barre d'adresse du navigateur.

#### 4.1.1. Récupération des paramètres d'URL en PHP (\$\_GET )

Du côté serveur, on doit indiquer la méthode de récupération ainsi que la clé (correspondant l'attribut name dans le formulaire html)

```
<?php
echo "Nom de l'utilisateur".$_GET['username']. " - " . $_GET['userfirstname'] ;
?>
```

**Important :** La méthode GET n'est utilisée que pour les données publiques. Ne jamais utiliser cette méthode pour envoyer des données sensibles telles que des mots-de-passe ou des données bancaires.

En fait la super-globale \$\_GET est un tableau associatif préparé par le préprocesseur PHP lors de l'appel d'une page PHP. On peut donc afficher son contenu par :

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

### 4.1.2. Encodage des paramètres d'URL en PHP

Pour transmettre des paramètres d'URL à partir d'une page PHP sans passer par un formulaire, nous devons encoder l'URL grâce à **urlencode("key","value")** de la façon suivante :

```
<?php
$monURL = "http://www.monsite.com/traitement1.php?";
$monURL .= urlencode("username"."="."LAPOINTE");
$monURL .= "&".urlencode("userfirstname"."="."Albert");
echo("<a href='".$monURL.'">Cliquez ici</a>");
?>
```

- Ce code produit un lien [Cliquez ici](#) qui va renvoyer vers traitement1.php avec les paramètres d'URL vus plus haut.

## 4.2. Méthode POST

Les données critiques (données privées ou mot-de-passe) sont souvent transmises de sorte à ce qu'elle ne soit pas exposées au regard de l'utilisateur dans le but de les protéger. De plus, l'url étant du texte, sa taille (longueur) est limitée.

L'alternative offerte par le http est la méthode POST. Les paramètres du POST sont placés dans le corps de la requête et non pas dans la partie visible contrairement à GET.

### 4.2.1. Récupération des paramètres post en PHP (\$\_POST)

Dans un formulaire, on peut invoquer un envoi de données par la méthode POST grâce à l'attribut method = 'POST'.

En PHP, on accèdera à ces données par **\$\_POST['nomParam']**

```
<?php
echo "Utilisateur".$_POST['username']." - ".$_POST['userfirstname'] ;
?>
```

### 4.3. La super-globale \$\_REQUEST

La requête étant formée d'un entête et d'un corps, elle contient aussi bien les paramètres du GET que ceux du POST.

Si on a envie qu'un script soit valide pour les deux méthodes (sans devoir le réécrire au changement de méthode), on peut invoquer la super-globale \$\_REQUEST

```
<?php
    echo "Utilisateur".$_REQUEST['username'];
?>
```

Code valable indépendamment de la méthode utilisée (GET ou POST).