

<FORM> FORMULAIRES HTML </FORM>

Un certain nombre d'éléments html servent à créer des formulaires de saisie (Balise <FORM>).
On y trouve des zones de saisie, des listes déroulantes, des cases à cocher, des boutons, ...
Les informations saisies dans un formulaire doivent être traitées par un script PHP ou ASP.

Exemple de structure d'un formulaire :

```
<form name="Exemple1" method="GET" action="page2.php" >
  Votre nom
  <input type="text" name="login" /> </br>
  Mot de passe:
  <input type="password" name="mdp" /> </br>
  <input type="submit" value="OK" />
  <input type="reset" value="Effacer"/>
</form>
```

Votre nom

Mot de passe:

<form>

C'est la balise la plus importante : elle définit le comportement du formulaire.

L'attribut action est le nom de la page appelée par le formulaire.

L'attribut method prend la valeur GET ou POST et détermine la façon dont les variables sont transmises à la page action :

- GET : sous forme visible (ex : `page2.php ?login=Dupont&mdp=12345`)
- POST : sous forme non visible (en-tête http).

Tout ce qui a été saisi dans le formulaire sera transmis, à **condition** que l'élément à envoyer ait un **attribut name**.

Dans **l'exemple**, la page « page2.php » est appelée quand on clique sur le bouton *submit*.

Le script PHP « page2.php » va recevoir les valeurs saisies dans les deux <input> du nom et du mot de passe, sous forme de variables portant le nom `$_GET['login']` et `$_GET['mdp']`.

L'attribut name représente donc le nom de la variable qui contiendra la valeur saisie.

Les variables POST ou GET sont utilisées dans la page action par des scripts php, asp, ... (côté serveur).

Les éléments à l'intérieur des balises <form> constituent le formulaire : input, select, ...

Les **balises <input>** sont de différents types : text, radio, password, button, submit, reset, ...

Le type reset correspond à un bouton qui efface ce qui est en cours de saisie.

Le type submit correspond à la validation du formulaire : l'action décrite dans la balise <form> est exécutée.

Si l'attribut action n'est pas précisé, la page est rechargée, avec la même possibilité de récupérer les variables du formulaire (par exemple dans un script intégré à la page). Cette solution permet de définir deux comportements de la page : premier chargement, lorsque le formulaire est vide, et les chargements suite à la saisie d'informations dans le formulaire.

JavaScript ne peut pas accéder à ces valeurs par leur nom de variable GET ou POST mais il peut y accéder directement par le DOM (avec `querySelector`, ou si le formulaire et ses éléments ont tous un attribut name : par `document.forms["monForm"]["nom"].value`)

Cela permet, par exemple d'effectuer un contrôle de la saisie avant d'envoyer le formulaire au serveur. Grâce à l'attribut **onsubmit** on peut exécuter un JavaScript avant d'appeler l'action.

Site de référence recommandé :

<http://www.w3schools.com>
<http://www.startyourdev.com/>