

Master Analyse Fonctionnelle et Analyse Appliquée

Les formulaires en HTML : **cours et exercices**

Mme Ibtissam.ARRASSEN
2018-2019

Formulaire

<p>Texte avant le formulaire</p>

<form>

<p>Texte à l'intérieur du formulaire</p>

</form>

<p>Texte après le formulaire</p>

- <form></form>. C'est la balise principale du formulaire, elle permet d'en indiquer le début et la fin.

Formulaire

- comment envoyer le texte rentré par le visiteur ? Par quel moyen ?
- Il existe 2 attributs à la balise `<form>` :
 - **method** : cet attribut indique par quel moyen les données vont être envoyées. Il existe 2 moyens pour envoyer des données sur le web :
 - `method="get"`
 - `method="post"`
 - **action** : c'est l'adresse de la page ou du programme qui va traiter les informations .

GET: Transmission par URL

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

```
<p>
```

```
  <label for="nom"> Votre nom  </label> :
```

```
  <input type="text"    name="nom"    id="nom" />
```

```
</p>
```

```
<p>
```

```
  <label for="prenom"> Votre prenom  </label> :
```

```
  <input type="text"    name="prenom"    id="prenom" />
```

```
</p>
```

```
</form>
```

Votre nom :

Votre prenom :

GET: Transmission par URL

- Par cette méthode, on va envoyer les données insérées dans le formulaire via l'url et on aura donc

`http://www.monsite.com/traitement.php?nom=aaa&prenom=bbb`

- L'envoi des données par URL se fait aussi via les liens:

```
<a href="traitement.php?nom=aaa&prenom=bbb ">  
    Aller à Traitement</a>
```

GET: Transmission par URL

- Récupérer les paramètres en PHP

<p> Bonjour

```
<?php echo $_GET['prenom'] . ' ' . $_GET['nom'];  
?> !
```

</p>

➔ on peut retrouver les paramètres envoyés dans l'URL grâce à un **array** nommé **\$_GET**.

POST

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

```
<p>
```

```
<label> Votre prenom </label>
```

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

```
</p>
```

```
<p>
```

```
<input type="submit" value="Valider" />
```

```
</p>
```

```
</form>
```

Votre prenom

Valider

➔ Le texte que le visiteur aura rentré sera disponible dans **cible.php** sous la forme d'une variable appelée **\$_POST['prenom']**.

Tyes de INPUT

- button : un bouton sans comportement défini.
- checkbox : une case à cocher qui permet de sélectionner/désélectionner une valeur
- color : un contrôle qui permet de définir une couleur.
- date : un contrôle qui permet de saisir une date (composé d'un jour, d'un mois et d'une année).
- email : un champ qui permet de saisir une adresse électronique.

Type de INPUT

- password : un champ texte sur une seule ligne dont la valeur est masquée. Les attributs `maxlength` et `minlength` définissent la taille maximale et minimale de la valeur à saisir dans le champ.
- radio : un bouton radio qui permet de sélectionner une seule valeur parmi un groupe de différentes valeurs.
- range : un contrôle qui permet de saisir un nombre dont la valeur exacte n'est pas importante.
- reset : un bouton qui réinitialise le contenu du formulaire avec les valeurs par défaut.

Type de INPUT

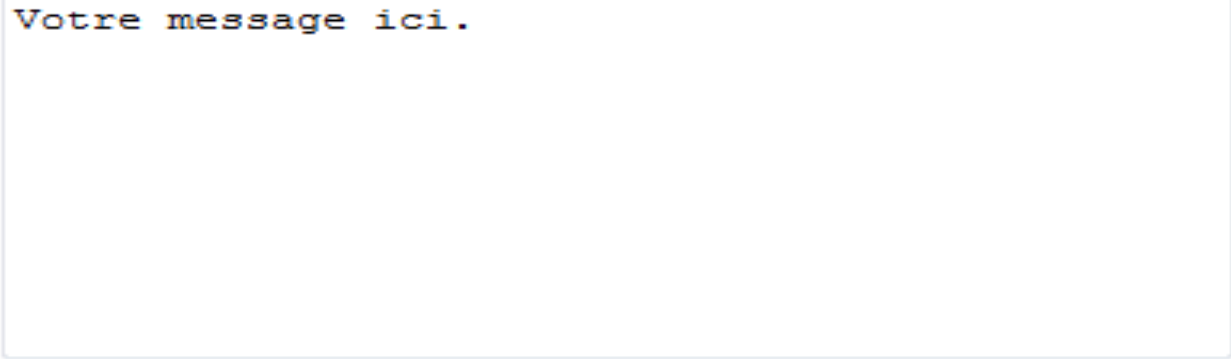
- search : un champ texte sur une ligne pour des termes de recherche. Les sauts de ligne sont automatiquement retirés.
 - submit : un bouton qui envoie le formulaire.
 - tel : un contrôle pour saisir un numéro de téléphone.
 - text : un champ texte sur une seule ligne. Les sauts de ligne sont automatiquement retirés.
 - url : un champ permettant de saisir une URL.
 - file : un contrôle qui permet de sélectionner un fichier.
- L'attribut **accept** définit les types de fichiers qui peuvent être sélectionnés.

Les grandes zones de texte

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

Votre message ici.

```
</textarea>
```



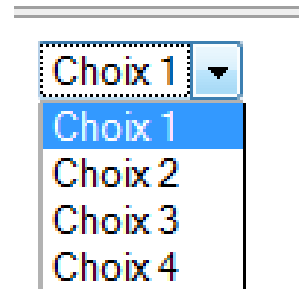
Votre message ici.

➔ Pour récupérer le message, on utilise **`$_POST['message']`**

La liste déroulante

- on utilise la balise `<select>` à laquelle on donne un nom (ici : "choix"). On écrit ensuite les différentes options disponibles... Puis on referme la balise avec `</select>`.

```
<select  name="choix" >  
  <option value= "choix1"> Choix 1</option>  
  <option value= "choix2"> Choix 2</option>  
  <option value= "choix3"> Choix 3</option>  
  <option value= "choix4"> Choix 4</option>  
</select>
```



Ici, une variable **`$_POST['choix']`** sera créée, et elle contiendra le choix qu'a fait l'utilisateur.

S'il a choisi "Choix 3", la variable `$_POST['choix']` sera égale à la valeur correspondante, c'est-à-dire "choix3".

La liste déroulante

- Le nom ne doit pas être trop long
- ne peut contenir ni caractères spéciaux, ni espace, ni caractères avec accent.
- Le premier signe doit être une lettre.
- Ensuite des chiffres sont également possibles, le tiret de soulignement (), le trait d'union (-), les deux points (:) ou le point (.)

Liste déroulante par groupe

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

```
<p>
```

```
<label for="pays">Dans quel pays habitez-vous ?</label><br />
```

```
<select name="pays" id="pays">
```

```
<optgroup label="Europe">
```

```
<option value="france">France</option>
```

```
<option value="espagne">Espagne</option>
```

```
<option value="italie">Italie</option>
```

```
<option value="royaume-uni">Royaume-Uni</option>
```

```
</optgroup>
```

```
<optgroup label="Amérique">
```

```
<option value="canada">Canada</option>
```

```
<option value="etats-unis">Etats-Unis</option>
```

```
</optgroup>
```

```
<optgroup label="Asie">
```

```
<option value="chine">Chine</option>
```

```
<option value="japon">Japon</option>
```

```
</optgroup>
```

```
</select>
```

```
</p>
```

```
</form>
```

Dans quel pays habitez-vous ?

France

Europe

- France
- Espagne
- Italie
- Royaume-Uni

Amérique

- Canada
- Etats-Unis

Asie

- Chine
- Japon

Les cases à cocher

```
<input type="checkbox" name="case1" id="case1" />  
<label for="case1" > Ma case à cocher 1 </label> <br/>
```

☐ Ma case à cocher 1
☐ Ma case à cocher 2

```
<input type="checkbox" name="case2" id="case2" />  
<label for="case2">Ma case à cocher2</label>
```

- Là encore, on donne un nom à la case à cocher via l'attribut name . Ce nom va générer une variable dans la page cible, par exemple **`$_POST['case1']`**.

Si la case est cochée, alors **`$_POST['case1']`** aura pour valeur "on".

Les boutons d'option

Voulez vous continuer?

```
<input type="radio" name="continue" value="oui" checked="checked" /> <label  
for="oui"> Oui </label>
```

```
<input type="radio" name="continue" value="non" />  
<label for="non"> Non </label>
```

- les deux boutons d'option ont le même nom ("**continue**"). C'est très important, car les boutons d'options fonctionnent par groupes : **tous les boutons d'option d'un même groupe doivent avoir le même nom**
- Cela permet au navigateur de savoir quels boutons d'option désactiver quand on active un autre bouton d'option du groupe.
- Il serait impossible en effet de pouvoir sélectionner "Oui" et "Non" à la fois.

Les boutons d'option

- Dans la page cible, une variable **`$_POST['continue']`** sera créée.
- Elle aura la valeur du bouton d'option choisi par le visiteur, issue de l'attribut value.
- Si on veut continuer, alors on aura :
`$_POST['continue']` = 'oui'.

Organiser le formulaire en plusieurs zones

<p>Vos coordonnées</p> <p>Quel est votre nom ?</p> <input type="text"/> <p>Quel est votre prénom ?</p> <input type="text"/> <p>Quel est votre e-mail ?</p> <input type="text"/>
<p>Votre souhait</p> <p>Faites un souhait que vous voudriez voir exaucé :</p> <ul style="list-style-type: none"><input type="radio"/> Etre riche<input type="radio"/> Etre célèbre<input type="radio"/> Etre encore plus intelligent<input type="radio"/> Autre... <p>Si "Autre", veuillez préciser :</p> <input type="text"/>

Un peu plus de design

Vos coordonnées

Quel est votre nom ?

Quel est votre prénom ?

Quel est votre e-mail ?

Votre souhait

Faites un souhait que vous voudriez voir exaucé :

- ☐ Etre riche
- ☐ Etre célèbre
- ☐ Etre **encore** plus intelligent
- ☐ Autre...

Si "Autre", veuillez préciser :

Un peu plus de design

input, textarea

```
{  
  font-family: "Times New Roman", Times, serif;  
  /* On modifie la police du texte tapé l'intérieur des champs */  
}
```

input:focus, textarea:focus {

```
  background-color: #FFFF99;  
  /* Quand le curseur est sur un champ */  
  
}
```

label

```
{  
  color: blue;  
  /* Colorer en bleu tous les labels (bah oui, pourquoi pas en bleu ?) */  
}
```

Un peu plus de design

```
legend /* On met un peu plus en valeur les titres des fieldset */
{
  font-family: Arial, "Arial Black", Georgia, "Times New Roman", Times, serif;
  color: #FF9933;
  font-weight: bold;
}
```

fieldset

```
{
  margin-bottom: 15px; /* Une marge pour séparer les fieldset */
  background-color: #FFFFFFCC;
}
```