

Les formulaires en HTML

1. Introduction

Les formulaires HTML permettent d'instaurer un dialogue entre un utilisateur et un site web ou une application. Ils permettent à l'utilisateur d'envoyer des données au site web. La plupart du temps, ces données sont envoyées à des serveurs web mais la page peut aussi les intercepter et les utiliser elle-même.

L'analyse et le traitement du formulaire devant être effectués côté serveur, la page du formulaire sera écrite en langage PHP. En cliquant sur le bouton envoyer les contenus des zones du texte du formulaire seront transmis à une page cible. PHP sous forme de paramètres directement intégrés dans la requête HTTP envoyée par le client au serveur.

2. Crédation d'un formulaire avec la balise < form >

a. La balise < form >

Les formulaires sont un moyen de récupérer des données à partir du client pour les passer à l'application serveur. Dans le fichier html, on utilise la balise `< form >` en y ajoutant l'attribut `method` qui permet d'indiquer quel est le type de requête HTTP.

Entrez votre nom:

Entrez votre email:

le code:

```
<form method="get">

    <label for="nom"> Entrez votre nom: </label>
    <input type="text" name="nom" minlength="4" maxlength="20" required> <br >

    <label for="email"> Entrez votre email: </label>
    <input type="email" name="email" required> <br>

    <input type="submit" value="Envoyer">
    <input type="reset" value="Effacer">
</form >
```

- **method** : permet de préciser comment les paramètres saisis dans le formulaire vont être envoyés vers le serveur. Cet attribut peut prendre 2 valeurs :
 - **GET**: Les paramètres sont transmis dans l'URL, et la taille est limitée. Cette requête est utile, par exemple pour enregistrer une requête effectuée sur un site marchand sous forme de marque-page, dans le navigateur
 - **POST**: Les paramètres sont transmis dans le corps de la requête HTTP, la taille est illimitée, les paramètres ne sont pas visibles dans l'URL, mais sont accessibles à l'aide de l'inspecteur de propriétés du navigateur web. Il est possible de transmettre tout type de données concernant l'utilisateur comme des textes courts, des photos ou des vidéos, etc..
- **Dans un formulaire, on trouve en général un bouton pour soumettre le formulaire.** Ce bouton a comme type "submit", On peut aussi rajouter un bouton permettant de vider tous les champs du formulaire avec le type "reset"

A faire:

Saisir les informations demandées et cliquer sur enregistrer. Que remarquez-vous dans l'URL? Que doit-on changer dans le code si on veut éviter cela?

Recopier le code dans notepad++ ou visual studio et le modifier afin que les informations remplies ne soient pas visibles dans l'URL

Voir le [lien suivant](#) pour mieux comprendre

b. Quelques types de composants graphiques d'un formulaire

Pour rendre une page web interactive, on peut insérer des composants d'interface graphique avec lesquels l'utilisateur peut interagir, en agissant dessus avec le doigt (écran tactile) ou la souris. Ces composants sont compatibles avec les navigateurs les plus utilisés.

Il existe un grand nombre de composants graphiques, les plus courants sont: **zone de texte, bouton poussoir, bouton radio, zone de mot de passe, case à cocher.**

en HTML, chaque composant peut être inséré avec la balise suivante:

```
<input type = "text" value = "Hello World!">
```

type et value sont des attributs "text " et "Hello World!" sont les valeurs de ces attributs. L'attribut **type** indique le type du composant, l'attribut **value** définit la valeur par défaut qui sera affichée lors du chargement de la page web. On peut ajouter un attribut **id** pour identifier de manière unique chaque composant, Chaque composant devient alors manipulable avec un script. On peut aussi modifier l'aspect d'un composant avec du code css.

Sauf cas contraires :

- On utilise la balise <input> pour définir un composant
- Le type de composant est indiqué grâce à l'attribut *type*
- Un identifiant est associé au composant pour pouvoir l'exploiter depuis javascript ou css (Voir après avec la séquence sur javascript)
- Un nom est associé au composant pour pouvoir récupérer les données saisies côté serveur
- Un attribut *placeholder* permet d'afficher en grisé un texte informatif sur l'information à saisir

1. Zone de saisie de texte

Il est utilisé pour créer un champ de saisie permettant à l'utilisateur d'entrer du texte sur une seule ligne.

```
<input type="text" size="nbCaractères" id="identifiant" name="nom" placeholder="texteDAide"/>
```

texte à saisir

le code:

```
<input type="text" size="15" placeholder = "texte à saisir">
```

Le placeholder dans un champ input HTML sert à afficher un texte indicatif à l'intérieur du champ avant que l'utilisateur ne saisisse quoi que ce soit.

2. Zone de saisie de mot de passe

Il est utilisé pour créer un champ de saisie où les caractères entrés sont masqués (souvent remplacés par des points ou des astérisques). Cela permet de protéger visuellement les informations sensibles, comme les mots de passe, lors de leur saisie.

```
<input type="password" size="nbCaractères" id="identifiant" name="nom" title="texteDAide">
```

le code:

```
<input type="password" size="15" title = "mot de passe">
```

L'attribut title sur chaque champ affiche une infobulle quand l'utilisateur survole le champ avec sa souris.

3. Zone de saisie de nombre

Permet de créer un champ de formulaire permettant à l'utilisateur de saisir uniquement des valeurs numériques. Ce type de champ inclut des fonctionnalités intégrées pour valider les entrées et rejeter les valeurs non numériques.

```
<input type="number" id="identifiant" name="nom" placeholder="texteDAide">
```

nb de pattes

le code:

```
<input type="number" id="id1" placeholder="nb de pattes">
```

on peut aussi ajouter des attributs min et max pour limiter les choix.

4. Zone de saisie de date

C'est un attribut utilisé dans les formulaires pour permettre à l'utilisateur de sélectionner une date via un sélecteur de date (date picker). Ce champ est particulièrement utile pour garantir que les dates saisies respectent un format standardisé (année-mois-jour : YYYY-MM-DD).

```
<input type="date" id="identifiant" name="nom">
```

mm/dd/yyyy 

le code:

```
<input type="date">
```

5. Case à cocher

C'est un type d'élément de formulaire utilisé pour afficher une case à cocher. Cet élément permet à l'utilisateur de sélectionner une ou plusieurs options parmi un ensemble de choix.

```
<input type="checkbox" id="identifiant" name="nom">
```

participera à la réunion

le code:

```
<input type="checkbox" id="id2">
```

6. Bouton radio

Il est utilisé pour créer des boutons radio dans un formulaire. Ces boutons permettent à l'utilisateur de choisir une seule option parmi un groupe d'options prédéfinies. Ils sont souvent utilisés lorsque vous voulez limiter la sélection à une seule valeur dans un ensemble.

```
<input type="radio" id="identifiant" name="nom">
```

Attention : veillez à mettre le même nom aux boutons radio fonctionnant ensemble.

homme

femme

le code:

```
homme <input type="radio" name="sexe" value="homme">
femme <input type="radio" name="sexe" value="femme">
```

Explications :

- input type="radio" : Définit un bouton radio.
- name="sexe" : Tous les boutons radio ayant le même nom appartiennent au même groupe, ce qui permet de sélectionner une seule option à la fois.
- value="femme" : La valeur envoyée au serveur si ce bouton est sélectionné.

Lorsque vous ouvrez ce code dans un navigateur, vous pourrez sélectionner un seul choix parmi les options proposées.

7. Bouton poussoir

C'est utilisé pour créer des boutons interactifs sur une page web. Les boutons de type button sont souvent utilisés avec des scripts JavaScript pour exécuter des actions personnalisées lorsqu'ils sont cliqués.

```
<input type="button" value= "valeur "id="identifiant" name="nom">
```

cliquer ici

le code:

```
<input type="button" name="bouton" value="cliquer ici">
```

8. Liste d'options

C'est un élément utilisé pour définir des choix dans un menu déroulant ou une liste de sélection.

```
<select" id="identifiant">
    <option value="valeurRenvoyée1">Texte affiché1</option>
    <option value="valeurRenvoyée2" selected="true">Texte affiché2</option>
    <option value="valeurRenvoyée3">Texte affiché3</option>
</select">
```

choix 1 ▾ Envoyer

le code:

```
<select>
<option value="choix 1" > choix 1</option>
<option value="choix 2" > choix 2 </option>
```

```
<option value="choix 3" > choix 3</option>
</select>
<button type="submit">Envoyer</button>
```

Explications :

- select : Crée une liste déroulante.
- option : Définit chaque option dans la liste.
- L'attribut value indique la valeur envoyée au serveur lors de la soumission du formulaire.
- button : Permet de soumettre le formulaire.

9. **type range:**

C'est un type d'élément de formulaire utilisé pour créer un contrôle de curseur (slider). Il permet à l'utilisateur de sélectionner une valeur numérique comprise entre deux bornes définies. Ce type est souvent utilisé lorsque la valeur exacte n'est pas critique, mais qu'une estimation visuelle est suffisante.

Niveau en html: 

le code:

```
Niveau en html: <input type="range" name="niveau" min="0" max= "10" >
```

3. Exercices

- **Exercice 1:** Ecrire un code pour afficher un formulaire d'inscription comme sur l'image ci-dessous:

Nom d'utilisateur :

Mot de passe :

Âge :

Correction

- **Exercice 2:** Ecrire un code pour lancer le sondage suivant:

Quel est votre langage préféré ?

- HTML
- CSS
- JavaScript

Quels outils utilisez-vous ?

- VS Code
- Notepad++

Quel est votre niveau?



Correction

