Créer des formulaires propres et sécurisés

Créer des formulaires HTML

1. **Structure d’un formulaire HTML**
    a. <form>
    b. <input>
    c. <label>
    d. <textarea>
    e. <select> et <option>
    f. <button>
    g. <fieldset> et <legend>
2. **Sécuriser ses formulaires en HTML**

Structure d’un formulaire HTML

Un formulaire en HTML se compose de plusieurs éléments :

- **<form> :** La balise qui englobe le formulaire.
- **<input> :** Un champ de saisie (texte, email, mot de passe, etc.).
- **<label> :** Une étiquette pour décrire un champ.
- **<textarea> :** Une zone de texte multilignes.
- **<select> et <option> :** Une liste déroulante.
- **<button> :** Un bouton pour soumettre ou réinitialiser le formulaire.
- **<fieldset> et <legend> :** Pour regrouper des champs sous un même bloc.

La balise <form>

👉 La balise **<form>** englobe tous les éléments du formulaire.

```
<form action="/traitement.php" method="POST" enctype="multipart/form-data">
  <!-- Les champs du formulaire viendront ici -->
</form>
```

- L'attribut **method** définit la méthode d'envoi :
  - POST : Envoi sécurisé (inscription, connexion...).
  - GET : Affichage des données dans l'URL (recherche, filtres...).
- L'attribut **action** spécifie où seront envoyées les données (une page de traitement).

En HTML pur, le formulaire ne fonctionne pas seul.

Il a besoin d'un langage autre langage (ex: **JavaScript, PHP**) pour traiter les données.

👉 **Cas particulier :** si un formulaire contient un champs Fichier, alors il faut ajouter enctype="multipart/form-data"

La balise <input>

👉 La balise **<input>** permet de saisir des données. C’est une balise orpheline (pas de fermeture). Son comportement dépend de l’attribut type.

L’attribut **type=""** est essentiel, car il définit le type de champ.

```
<input type="text" placeholder="Nom" name=”nom” required>
```

- **required** : Oblige l’utilisateur à remplir le champ.
- **placeholder** : Texte indicatif dans le champ.
- **name** : Identifie le champ dans l’envoi des données.

La balise <input> : les attributs type=””

Type	Utilité	Exemple
text	Champ texte basique	<input type="text">
email	Adresse email	<input type="email">
password	Mot de passe caché	<input type="password">
tel	Numéro de téléphone	<input type="tel">
number	Nombre limité ou non	<input type="number" min="1" max="10">
date	Sélection de date	<input type="date">
file	Sélection de fichier	<input type="file">. Nécessite enctype="multipart/form-data" sur <form>
checkbox	Case à cocher (plusieurs choix possibles)	<input type="checkbox">
radio	Boutons radio (choix unique)	<input type="radio">
color	Sélecteur de couleur	<input type="color">
hidden	Champ invisible pour l’utilisateur	<input type="hidden" value="42">

Les boutons <button> et <input type="submit">

👉 Les boutons permettent **d’envoyer ou de réinitialiser** un formulaire.

Il y a plusieurs options pour écrire un bouton en HTML :

```
<input type="submit" value="Envoyer">
```

```
<input type="reset" value="Réinitialiser">
```

```
<button type="submit">Envoyer</button>
```

Différence entre <input type="submit"> et <button>

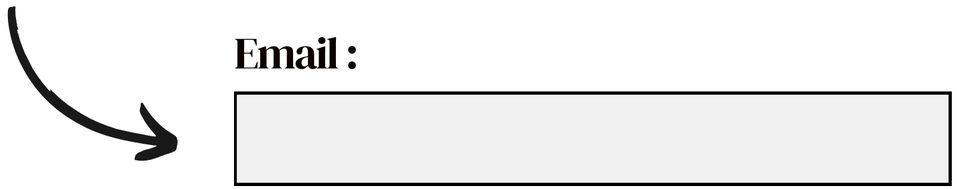
- **<input>** : Simple bouton d’envoi.
- **<button>** : Plus flexible (ajout d’icônes, texte personnalisé).

La balise <label> : associer une étiquette à un champ

👉 Les **labels** améliorent l'accessibilité et permettent de cliquer sur le texte pour activer un champ.

```
<label for="email">Email :</label>
<input type="email" id="email">
```

L'attribut **for** du <label> doit être identique à id du champ pour les lier.



La balise <textarea> : saisie de texte long

👉 Un champ pour les commentaires, messages, etc.

```
<label for="message">Message :</label>

<textarea name=”message” id="message" rows=”5" cols=”30"></textarea>
```

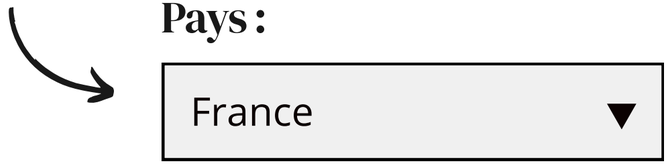
Contrairement à <input type=”text”>, **<textarea>** a une balise fermante (</textarea>) et peut contenir du texte par défaut.

Message :

Les listes déroulantes (<select> et <option>)

Elles permettent aux utilisateurs de choisir une option parmi plusieurs.

```
<label for="pays">Pays :</label>
<select id="pays" name="pays">
  <option value="fr" selected>France</option>
  <option value="us">États-Unis</option>
  <option value="jp">Japon</option>
</select>
```



👉 L’attribut **selected** permet de définir l’option sélectionnée par défaut.

Les cases à cocher

👉 **Checkbox** <input type="checkbox"> : Plusieurs choix possibles.

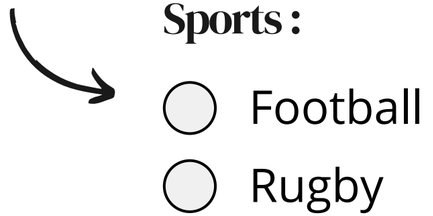
```
<label for=”sport”>Sports :
<input type="checkbox" name=”sport” value=”foot” id=”sport”> Football
</label>
```

Les boutons radio

👉 **Radio** <input type="radio"> : Un seul choix possible.

```
<label for=”sport”>Sports :
<input type="radio" name=”sport” value=”foot” id=”sport”> Football
</label>
```

Tous les boutons doivent avoir le même name pour fonctionner ensemble.



Organiser un Formulaire avec <fieldset> et <legend>

👉 La balise **<fieldset>** permet de regrouper des champs liés, et **<legend>** ajoute un titre au groupe.

```
<fieldset>
  <legend>Informations personnelles</legend>
  <label for="nom">Nom :</label>
  <input type="text" id="nom" name="nom">
  <label for="email">Email :</label>
  <input type="email" id="email" name="email">
</fieldset>
```

Sécuriser un formulaire HTML : valider les données d'entrée

👉 Les formulaires sont une porte d'entrée pour les données des utilisateurs mais aussi pour les attaques.

Il existe deux types d'utilisateurs qui peuvent poser problème :

- **Utilisateurs maladroits** : Ils envoient des données incorrectes par erreur.
- **Utilisateurs malveillants** : Ils tentent d'exploiter des failles pour voler ou modifier des données.

HTML propose des **attributs de validation** qui empêchent l'envoi de données invalides.

👉 **Exemples :**

```
<input type="text" name="nom" required>
<input type="email" name="email" required>
<input type="number" name="age" min="1" max="100">
<input type="password" name="password" minlength="8" required>
<input type="file" name="photo" accept="image/*">
```

Attributs de validation d'un formulaire HTML

Attribut	Définition
size	Permet de spécifier le nombre de caractères dans un champ
minlength	Permet de spécifier le nombre minimum de caractères dans un champ
maxlength	Permet de spécifier le nombre maximum de caractères dans un champ
min	Permet de spécifier une valeur minimale pour un champ de type number ou date
max	Permet de spécifier une valeur maximale pour un champ de type number ou date
step	Permet de définir un multiple de validité pour un champ acceptant des donnés de type nombre ou date. En indiquant step="4" , les nombres valides seront -8, -4, 0, 4, 8, etc.
autocomplete	Permet d'activer l'auto complétion pour un champ : si un utilisateur a déjà rempli un formulaire, des valeurs lui seront proposées automatiquement lorsqu'il va commencer à remplir le champ
required	Permet de forcer le remplissage d'un champ. Le formulaire ne pourra pas être envoyé si le champ est vide
pattern	Permet de préciser une expression régulière. La valeur du champ devra respecter la contrainte de la regex pour être valide

Sécuriser un formulaire HTML : utiliser SSL

Un formulaire HTML sans SSL envoie les données en clair sur le réseau, ce qui signifie que :

- Un hacker peut intercepter et lire les données envoyées (ex : mots de passe, emails, numéros de carte bancaire).
- L'utilisateur ne peut pas être sûr qu'il communique avec le bon serveur.

Protocole	Sécurité	Utilisation
HTTP	❌ Non sécurisé, les données sont envoyées en clair	❌ ⚠ Formulaires sensibles à éviter !
HTTPS	✅ Chiffrement SSL/TLS, protège les données	✅ 🔒 Recommandé pour tous les sites, surtout ceux avec des formulaires

Sécuriser un formulaire HTML : utiliser SSL

👉 **Que se passe-t-il si un formulaire est envoyé en HTTP ?**

- Les données peuvent être volées si l'utilisateur est sur un réseau Wi-Fi public ou piraté.
- Les hackers peuvent intercepter les identifiants de connexion et se connecter à la place de l'utilisateur.
- Un attaquant peut modifier la requête envoyée au serveur avant qu'elle n'arrive (attaque MITM).

👉 **Exemple d'attaque man-in-the-middle (MITM) :**

1. L'utilisateur remplit son formulaire sur http://mon-site.com.
2. Un hacker intercepte la requête et modifie les données.
3. Le hacker récupère l'email et le mot de passe et peut accéder au compte.

👉 **Avec HTTPS, ces attaques sont impossibles, car les données sont chiffrées.**


RTS
AVEC LE SOUTIEN DE
Groupe Mutuel
Assurances
Versicherungen
Assicurazioni
Assuré. Là. Maintenant.

Sécuriser un formulaire HTML : utiliser un CAPTCHA

👉 **Objectifs du Captcha :**

- Empêcher les robots d'envoyer des formulaires en masse.
- Protéger contre le spam et les attaques automatisées.
- Limiter les inscriptions frauduleuses et le piratage de comptes.

👉 **Exemples de Captcha :**

- Texte à recopier (ex : "Tapez le mot affiché").
- Opération mathématique simple (ex : "5 + 3 = ?").
- Sélection d'images (ex : "Cochez toutes les images avec un vélo").
- reCAPTCHA de Google (cocher une case "Je ne suis pas un robot").
- Honeypot, consiste à placer un champs invisible. S'il est complété c'est qu'il s'agit d'un robot.

Sécuriser un formulaire HTML : les failles XSS

👉 Le **XSS** (Cross-Site Scripting) est une attaque qui consiste **à injecter du code JavaScript** malveillant dans une page web. Ce code peut être exécuté dans le navigateur des autres utilisateurs à leur insu.

Le problème vient des formulaires HTML qui **acceptent du texte sans le filtrer**.

👉 **Comment bloquer les attaques XSS ?**

- Toujours échapper les entrées utilisateurs des formulaires
- Utiliser des règles de validation côté serveur
- Tester son site avec des outils come XSS Strike

Mes recettes | Title | s0md3v/XSStrike: Most adv | PHP: htmlspecialchars - Ma

localhost:3000/recettes/search?q=poulet

Mes recettes

Résultat pour votre recherche "poulet"

Poulet rôti aux herbes

Délicieux poulet rôti avec un mélange d'herbes aromatiques.

Voir la recette

Curry de poulet indien

Plat épicé de poulet avec une sauce curry parfumée aux épices indiennes.

Voir la recette

Poulet au beurre indien

Plat indien de poulet cuit dans une sauce riche et crémeuse.

Voir la recette

Ouvrir le QCM

Ressources

Les formulaires
https://developer.mozilla.org/fr/docs/Web/HTML/Element/form
https://developer.mozilla.org/fr/docs/Web/HTML/Element/input
https://developer.mozilla.org/fr/docs/Web/HTML/Reference

Le problème des données utilisateurs
https://www.pierre-giraud.com/html-css-apprendre-coder-cours/attribut-formulaire/

(vidéo) Attaque MITM
https://www.afecdax.ovh/ressources/videos/securite/attaque-mitm.mp4

(vidéo) Attaque XSS
https://www.afecdax.ovh/ressources/videos/securite/attaque-xss.mp4