

# TP: Validation de Formulaire en JavaScript (Front-End)

## Objectif :

Cr  er un formulaire d'inscription en HTML qui demande    l'utilisateur de saisir :

- **Nom d'utilisateur** (minimum 5 caract  res)
- **Adresse e-mail** (doit   tre une adresse e-mail valide)
- **Mot de passe** (minimum 8 caract  res, doit contenir au moins une lettre majuscule, une lettre minuscule et un chiffre)
- **Confirmation du mot de passe** (doit correspondre au mot de passe saisi)

Vous devez utiliser **JavaScript** pour valider les entr  es de l'utilisateur avant la soumission du formulaire. Si des erreurs sont d  tect  es, le formulaire ne doit pas   tre soumis, et des messages d'erreur appropri  s doivent   tre affich  s    c  t   des champs correspondants.

## Instructions :

1. **HTML :**
  - Cr  ez un fichier `index.html`.
  - Int  grez le formulaire avec les champs mentionn  s.
  - Assurez-vous d'ajouter des `id` ou des `class` aux   l  ments pour pouvoir les cibler en CSS et JavaScript.
2. **CSS :**
  - Cr  ez un fichier `styles.css`.
  - Stylisez le formulaire pour qu'il soit clair et convivial.
  - Les messages d'erreur doivent   tre visibles et attirants (par exemple, en rouge).
3. **JavaScript :**
  - Cr  ez un fichier `script.js`.
  - Ajoutez des   couteurs d'  v  nements pour la soumission du formulaire.
  - Validez chaque champ selon les crit  res sp  cifi  s.
  - Si un champ n'est pas valide, emp  chez la soumission du formulaire et affichez un message d'erreur sp  cifique sous le champ.
  - Si tous les champs sont valides, affichez un message de succ  s (par exemple, "Inscription r  ussie !").

## Consignes suppl  mentaires :

- **Validation du Nom d'utilisateur :** doit contenir au moins 5 caract  res.
- **Validation de l'Adresse e-mail :** utilisez une expression r  guli  re pour v  rifier le format.
- **Validation du Mot de passe :** doit avoir au moins 8 caract  res, une majuscule, une minuscule et un chiffre.
- **Confirmation du Mot de passe :** doit   tre identique au mot de passe saisi.

## Exemple de Structure HTML :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Formulaire d'inscription</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <form id="registrationForm">
    <div>
      <label for="username">Nom d'utilisateur :</label>
      <input type="text" id="username" name="username">
      <small class="error-message"></small>
    </div>
    <div>
      <label for="email">Adresse e-mail :</label>
      <input type="text" id="email" name="email">
      <small class="error-message"></small>
    </div>
    <div>
      <label for="password">Mot de passe :</label>
      <input type="password" id="password" name="password">
      <small class="error-message"></small>
    </div>
    <div>
      <label for="confirmPassword">Confirmer le mot de passe :</label>
      <input type="password" id="confirmPassword" name="confirmPassword">
      <small class="error-message"></small>
    </div>
    <button type="submit">S'inscrire</button>
  </form>

  <script src="script.js"></script>
</body>
</html>
```

## Conseils :

- **Expressions R  guli  res :**
  - Pour l'e-mail : `/^\S+@(\S+\.){1,4}\S+$/`
  - Pour le mot de passe : utilisez une expression qui v  rifie les crit  res requis.
  - [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/RegExp/test](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/RegExp/test)
- **Pr  venir la Soumission du Formulaire :**
  - Utilisez `event.preventDefault()` dans votre gestionnaire d'  v  nement pour emp  cher la soumission si le formulaire n'est pas valide.
- **Affichage des Erreurs :**
  - S  lectionnez l'  l  ment `<small class="error-message">` correspondant pour afficher le message d'erreur.

##   tapes pour la Validation en JavaScript :

1. S  lectionnez le formulaire et les champs d'entr  e en utilisant `document.getElementById` ou `document.querySelector`.
2. Ajoutez un   couteur d'  v  nement pour l'  v  nement `submit` du formulaire.
3. Dans la fonction de validation :
  - Initialisez une variable pour suivre l'  tat de validation (par exemple, `let isValid = true;`).
  - Pour chaque champ :
    - V  rifiez la condition.
    - Si la condition n'est pas remplie :
      - Mettez `isValid`    `false`.
      - Affichez un message d'erreur dans le `<small>` correspondant.
    - Sinon :
      - Effacez le message d'erreur s'il y en a un.
4. Si `isValid` est toujours `true` apr  s toutes les v  rifications, vous pouvez afficher un message de succ  s.

## Extension Possible :

- **Validation en Temps R  el :**
  - Ajoutez des   couteurs d'  v  nements `input` ou `blur` sur chaque champ pour valider en temps r  el.

##    quoi sert l'  v  nement `input` ?

L'  v  nement `input` est d  clench   **   chaque fois que tu tapes une lettre** ou que tu changes quelque chose dans un champ de formulaire.

### Exemple simple:

Quand tu tapes ton nom dans une bo  te de texte,

`input` peut v  rifier :

- Si tu as mis assez de lettres.
- Si tu n'as pas utilis   de caract  res interdits (par exemple, des chiffres dans un champ de texte).

C'est utile pour **valider en temps r  el** ce que l'utilisateur   crit, sans attendre qu'il clique sur "Envoyer".

##    quoi sert l'  v  nement `blur` ?

L'  v  nement `blur` est d  clench   quand tu **quittes un champ** de formulaire. Par exemple :

- Tu tapes ton email.
- Puis tu cliques dans une autre bo  te pour entrer ton mot de passe.

   ce moment-l  , `blur` peut v  rifier :

- Si ton email est valide (par exemple, il contient bien `@` et un domaine comme `.com`).

Cela permet de valider **quand l'utilisateur a fini avec un champ**, pas pendant qu'il est en train de taper.

## Pourquoi les utiliser ensemble ?

- `input` te permet de donner des retours **imm  diats** pendant que l'utilisateur tape.
  - Par exemple: "OK, ton mot de passe est assez long."
- `blur` te permet de valider seulement quand l'utilisateur **passe    autre chose**, ce qui   vite de v  rifier trop souvent.

Utilis  s ensemble, ils offrent une bonne exp  rience utilisateur : des **indications en temps r  el** tout en   vitant d'  tre trop intrusif.

**Bon Courage !**