

# Projet 10 - Formulaire

## Le Besoin

Créer le front-end d'un formulaire en JavaScript ainsi que ces messages d'erreurs.

## Pré-requis

- la condition `if`
- Récupérer la longueur d'une string

```
const prenom = "Massinissa";  
prenom.length // Nombre de caractère
```

- Récupérer la valeur d'une balise `<input>`

```
const balise = document.querySelector("input");  
balise.value; // Renvoie le contenu de l'input.  
balise.value = "Contenu"; // Modifie le contenu de l'input.
```

- L'événement `submit`

```
const formulaire = document.querySelector("form");  
formulaire.addEventListener("submit", function(event){  
    event.preventDefault(); // Empêche le rechargement de la page  
  
    /**  
     * Traite le contenu du formulaire ...  
     */  
});
```

L'événement `submit` se produit **UNIQUEMENT** sur la balise `form` lorsque l'utilisateur clique sur un bouton de type `submit` ou tape sur la touche entrée.

Par défaut l'événement `submit` demande au navigateur d'envoyer le contenu du formulaire au serveur dans une requête HTTP ce qui provoque un chargement de la page.

Si vous voulez vérifiez les champs du formulaire en JavaScript avant d'envoyer un message au serveur il faut désactiver le comportement par default de l'événement grâce à la fonction `preventDefault()` .

## Récupérer les valeurs du formulaire à partir de l'event

Dans le code suivant j'utilise l'objet event et la classe FormData pour former un objet facile à utiliser contenant comme attributs les `value` des balises `<input>` .

Le nom des attributs est défini par l'attribut HTML `name` des balises `<input>` .

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <form action="">
    <input type="email" name="mail" id="">
    <input type="text" name="nom" id="">
    <button type="submit">submit</button>
  </form>
</body>
<script>

  const form = document.querySelector("form");

  form.addEventListener("submit", event=>{
    event.preventDefault();
    const formData = new FormData(form);
    console.log(formData.mail);
    console.log(formData.nom);
    console.log(formData);
  });
</script>
</html>
```

# Améliorer la structure

Notez que je pourrais me servir de la variable global `form` plutôt que de `event.target` mais je préfère rendre mon code le moins dépendant possible de variables globales pour pouvoir facilement encapsuler cette fonction anonyme dans une fonction bien définie et améliorer la structure et la lisibilité du code.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <form action="">
    <input type="email" name="mail" id="">
    <input type="text" name="nom" id="">
    <button type="submit">submit</button>
  </form>
</body>
<script>

  const form = document.querySelector("form");

  form.addEventListener("submit", onLogin);

  function onLogin(event){
    event.preventDefault();
    const formData = new FormData(form);
    console.log(formData.mail);
    console.log(formData.nom);
    console.log(formData);

    // Ici je pourrais par exemple faire une requête HTTP pour demander un jeton d'acces

  }
</script>
</html>
```

# Cahier des charges

Tâches	Description	Contraintes
Intégration du formulaire	Intégrer en HTML/CSS un formulaire de contact	Il doit contenir : nom, prénom, email et un message avec une bouton ENVOYER de type <code>submit</code>
Message d'erreur du champs nom	Afficher un message d'erreur sous le champs nom si les conditions ne sont pas remplies lors de l'événement <code>submit</code>	nom est supérieur à 2 et inférieur à 20 caractères.
Message d'erreur du champs prénom	Afficher un message d'erreur sous le champs prénom si les conditions ne sont pas remplies lors de l'événement <code>submit</code>	prénom est supérieur à 2 et inférieur à 20 caractères.
Message d'erreur du champs email	Afficher un message d'erreur sous le champs email si les conditions ne sont pas remplies lors de l'événement <code>submit</code>	L'email doit être une adresse email valide (voir la fonction <code>isValidEmail</code> ci-dessus).
Message d'erreur du champs message	Afficher un message d'erreur sous le champs message si les conditions ne sont pas remplies lors de l'événement <code>submit</code>	Le message doit être inférieur à 100 et supérieur à 10 caractères.

## La fonction `isValidEmail`

Voici la fonction qui permet de tester si une email est valide.

La fonction utilise le concept de regex, qui est un langage de parsing de texte qui permet de tester ou extraire des données d'un texte.

Rendez-vous sur le site web [regex101](https://regex101.com/) si vous avez besoin d'apprendre les regex.

```
/**
 * Renvoie vrai si la string email passé en paramètre correspond à une adresse email va
 */
function isValidEmail(email){
    const emailFormat = /^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*\\.\\w{2,3}+$/; // Création d'u
    if (emailFormat.test(email))
    {
        return true;
    }else{
        return false
    }
}
```