|  |  |  |
| --- | --- | --- |
| Numérique et Sciences Informatiques | | |
| 1h30 | **Web : Formulaire et Javascript** |  |
| **Objectif** : formulaire et Javascript | | |
| **Matériel**: éditeur Notepad++ et navigateur web Chrome (car il y a des problèmes avec Firefox) | | |

On trouve souvent des formulaires sur les sites Web, par exemple pour créer un compte :



Pour notre site nous allons créer un formulaire sur la page **contact.html** dans le répertoire **V4**.

Les informations saisies dans ce formulaire pourront être envoyées à un serveur.

La première version de notre formulaire sera très basique, on demandera à l’utilisateur son **nom**, son **adresse mail** et son **message**.

Pour commencer notre formulaire nous allons créer une balise <form> avec un identifiant (**id**) pour le script et une classe (**class**) pour la feuille de style (CSS).

Il faudra placer cette balise directement sous le titre <h1>Formulaire</h1> dans la page **contact.html**.

<form id= "formulaire-contact" class= "formulaire-contact"**>**

...

</form **>**

Code HTML pour créer les éléments de notre formulaire :

<label>Nom: <input type="text" name="nom"> <br></label>



Le **label** permet d'écrire un texte devant la case.

L'**input** permet de créer une case de saisie.

L'attribut **name** permet de récupérer les valeurs saisies lors de l’envoie du formulaire.

**Etape 1** : Ajoutez les champs **Nom**, **Adresse e-mail** et **Message**.

Notre formulaire est prêt mais on ne peut pas encore envoyer les informations saisies au serveur.

Pour envoyer les informations nous allons utiliser l’input **submit** (bouton Envoyer).

**Remarque :** comme nous n'avons pas de serveur, nous allons envoyer le résultat du formulaire à la page **resultat.html**.

Nous allons donc spécifier la page de destination avec l’option **formaction**.

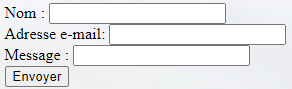
<input type="submit" formaction="./resultat.html" formmethod="get">

L’option **formmethod** permet de spécifier la méthode de transmission des informations saisies dans le formulaire :

* La méthode **POST** permet d’envoyer le résultat vers le serveur sans que le client puisse récupérer ces données dans son navigateur.
* La méthode **GET** permet d’envoyer le résultat vers le serveur en ajoutant le résultat du formulaire à l’URL du serveur, ce qui permet de récupérer les données côté client.

Pour traiter le résultat avec la méthode **POST** il faut utiliser un langage de programmation du côté serveur (le PHP) dans le TP on va utiliser la méthode **GET**.

**Etape 2** : ajoutez le bouton **submit** avec la méthode **GET** puis testez l’envoie du formulaire.



On obtient le contenu de notre formulaire dans l’URL mais rien ne s’affiche sur la page **resultat.html**.

Pour l'affichage vous pourrez constater la présence du code suivant dans la page **resultat.html** :

<div class="resultat" id="resultat">

</div >

Vous allez ajouter un script (Javascript) en bas la page **resultat.html** pour récupérer les informations présentes dans l’URL.

<script >

// Script résultat

const queryString = window.location.search;

console.log(queryString);

const urlParams = new URLSearchParams(queryString);

</script >

Ce script permet de récupérer tous le contenu de notre formulaire (présent dans l’URL) dans la variable **urlParams**.

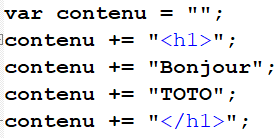
Pour accéder au premier contenu du formulaire nous utiliserons : urlParams.get("nom");

Rappel : "nom" est l’attribut **name** de la première case du formulaire :

<label>Nom: <input type="text" name="nom"> <br></label>

Pour compléter le script sur la page **resultat.html** vous allez créer une chaîne de caractères que nous allons appeler **contenu**.

Exemple :

 → **contenu =** "<h1>Bonjour TOTO</h1>"

Le script devra ajouter **contenu** entre <div class="resultat" id="resultat"> et </di>**,** grâce à l’identifiant (id) **resultat**.

Exemple : si l’on souhaite afficher le **nom** saisi par l'utilisateur dans notre formulaire :

var contenu = "";

contenu += urlParams.get('nom');

var resultat = document.getElementById("resultat");

resultat.innerHTML += contenu;

On peut ajouter des balises HTML pour améliorer l'affichage avec par exemple des paragraphes <p> ou des retours à la ligne <br>.

**Etape 3** : créez un script sur la page **resultat.html** donnant le résultat suivant :

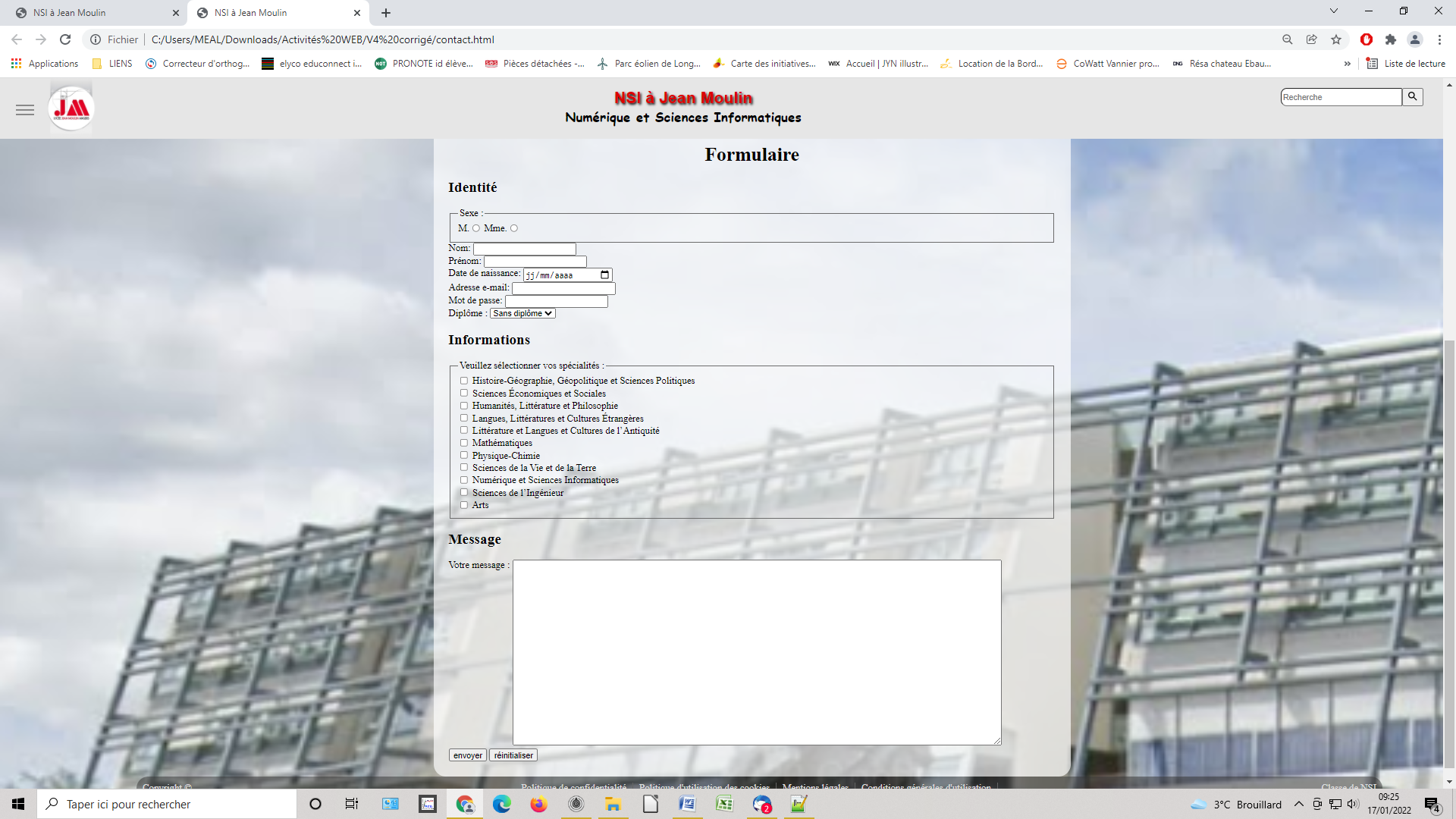


**JavaScript est le langage le plus répandu pour faire de la programmation d'interactions sur des sites web.**

**Il permet d’interagir avec le contenu des pages web pour les rendre dynamiques.**

**Son avantage est qu’il n’a pas besoin de serveur pour être exécuter, il est exécuté directement sur le navigateur web du client.**

**Etape 4** : modifiez votre code pour que notre formulaire ressemble à sa :



Documentation utile : https://developer.mozilla.org/fr/docs/Web/HTML/Element/Input

**Etape 5** : modifiez votre code afin que lorsque vous cliquez sur le bouton **envoyer** le résultat sera présenté comme suit :

