|  |  |  |
| --- | --- | --- |
| Numérique et Sciences Informatiques | | |
| 2h | **Web : Interaction Homme/Machine** |  |
| **Objectif** : gérer des événements sur un page Web avec JavaScript | | |
| **Matériel**: éditeur Notepad++ et navigateur web Chrome (car il y a des problèmes avec Firefox) | | |

Aujourd’hui les plus grands sites web proposent des interactions avec l’utilisateur pour améliorer leur **expérience utilisateur**.

L’interaction **Homme/Machine** passe par différents moyens comme l’adaptation du contenu au client, la réaction au clic de la souris, l’apparition d’un menu au survol de la souris ...

Nous allons voir comment le langage **JavaScript** va nous permettre d’interagir facilement avec l’utilisateur en programmant des événements qui réagirons aux actions de celui-ci.

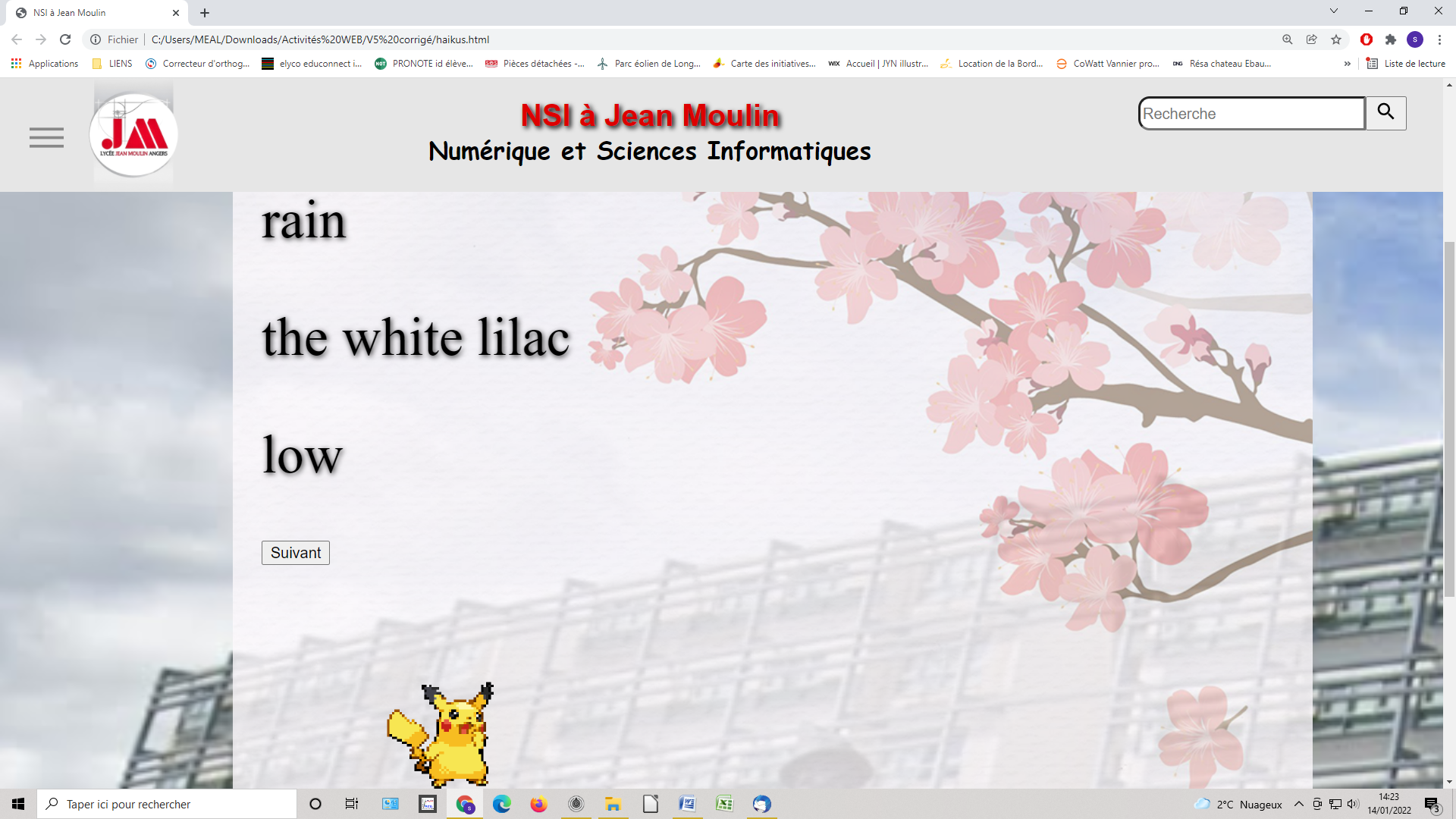
**Générer un Haïku**

Un **haïku** est un poème d'origine japonaise extrêmement bref, célébrant l'évanescence des choses et les sensations qu'elles suscitent.

**Exemple :**

|  |
| --- |
| Dans cette forêt urbaine  Des arbres en béton  Les feuilles ne bougent pas |

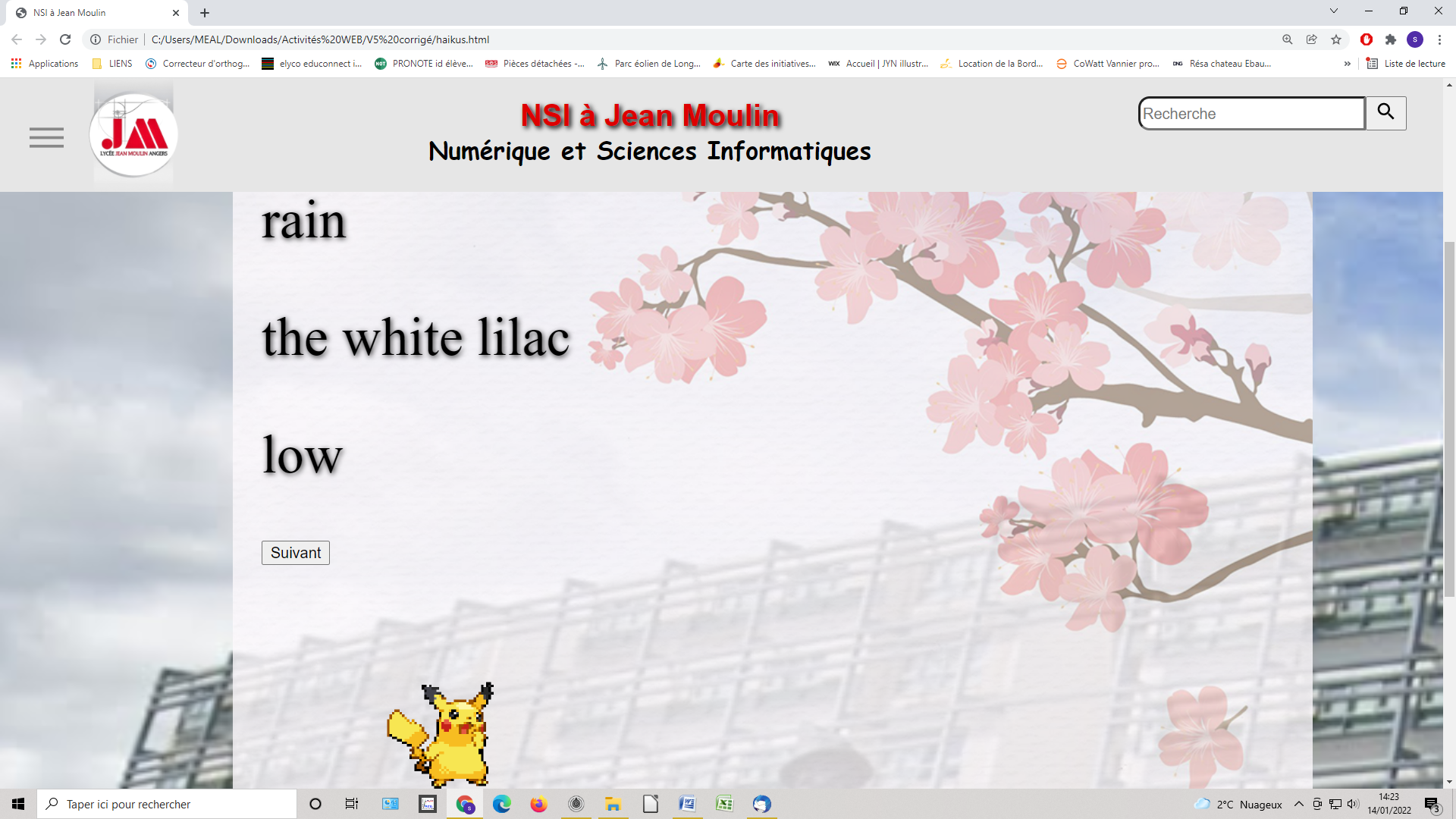
La page **activites.html** dans le répertoire **V5** propose déjà un **haïku** généré automatiquement lors de son chargement.

Nous souhaitons désormais ajouter des interactions avec dans un premier temps un bouton  qui permettra de générer aléatoirement (sans avoir besoin de rafraichir la page) un nouvel **haïku**.

Pour des raisons de clarté on va écrire le code **JavaScript** dans le fichier **haiku.js** et l’inclure dans notre fichier **activites.html** grâce à la ligne suivante :

**<script type="text/javascript" src="haikus.js"> </script>**

Le script **haiku.js** possède une fonction **genererHaiku()** qui se déclenche à l’ouverture de la page web.

**Etape 1 :** ajoutez le code **<button onclick=genererHaiku()>Suivant</button>** dans **activites.html** pour déclencher la fonction **genererHaiku()** lorsque l’on clique sur le bouton 

Testez la génération de nouvel **haiku**.

La balise <**button>** fonctionne bien lorsque l’on a peu d’événements (cliquer sur un bouton est un évènement) à gérer.

Il est possible de faire la même chose en utilisant un **listener**, qui est plus adapté pour gérer beaucoup d’événements.

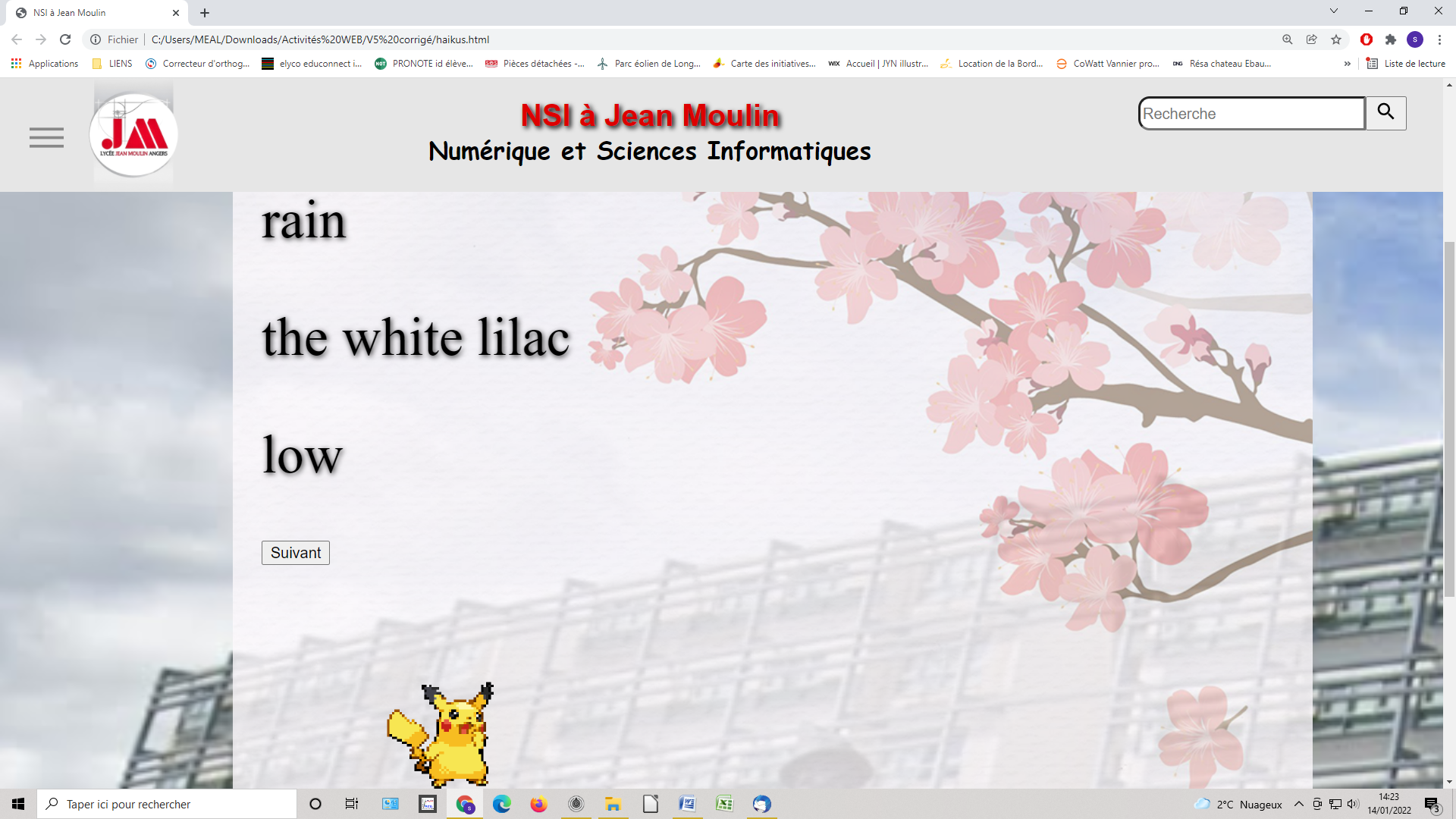
Pour cela on va attribuer un identifiant **id="boutonSuivant"** à notre bouton et lui affecter un événement **click** avec un **listener** :

**<button id="boutonSuivant">Suivant</button>**

On va également ajouter le code suivant dans le script **haiku.js** :

**var bouton = document.getElementById("boutonSuivant");**

**bouton.addEventListener("click",genererHaiku);**

La première ligne du code permet de récupérer, dans la variable **bouton**, les événements sur  grâce à l'**id "boutonSuivant".**

La deuxième ligne permet de définir la fonction (**genererHaiku**) à exécuter lorsque l’événement "**click"** est détecté sur la variable **bouton**.

**Etape 2** : modifiez le code, avec un **listener**, et testez la génération de nouveaux **haikus**.

Maintenant, on souhaite avoir des nouveaux **haïkus** en ne changeant **qu’une seule ligne à la fois**.

Pour cela on déclenchera (événement) uniquement le changement de la ligne sur laquelle on cliquera.

Chacune des lignes à un identifiant **id** : **firstLine**, **secondLine** et **thirdLine**.

**Etape 3** : modifiez le script permettant le changement de chaque ligne de l'**haiku** lorsqu'on clic dessus en créant 3 fonctions : **newFirstLine**, **newSecondLine** et **newThirdLine**.

Chaque ligne sera considérée comme un bouton.

**Etape 4** : programmez l'événement **mouseenter** sur chaque ligne de l'**haïku** afin de changer le style de la ligne survolée par la souris.

**Pour cela suivez les étapes suivantes :**

Code pour écouter la sourie sur la première ligne et déclencher la fonction **enterFirstLineStyle** :

**var bouton1 = document.getElementById("firstLine");**

**bouton1.addEventListener("mouseenter",enterFirstLineStyle);**

Dans le fichier **haiku.css** on peut voir qu’il existe déjà un style pour la classe **.haiku**.

Nous allons créer un nouveau style qu'on appellera **.haikuOver** :

**.haikuOver{**

**font-family:Brush Script Std;**

**font-size: 36pt;**

**text-shadow:5px 5px 5px #000000;**

**color: red;**

**}**

Code de la fonction **enterFirstLineStyle** pour modifier le style de la première ligne de l'**haïku** lorsque la sourie la survolera :

**firstLine.classList.replace("haiku","haikuOver");**

Le code indique que l’on souhaite changer la classe par défaut (**class='haiku'**) de la première ligne par la classe (**class='haikuOver'**).

**Etape 5** : programmez l'événement **mouseleaver** sur chaque ligne de l'**haïku** afin de revenir au style d'origine lorsque la souris ne sera plus sur la ligne (fonction **leaveFirstLineStyle**).