

```

1  window.onload = function() {
2      let fileInput = document.getElementById('fileInput');
3      let fileDisplayArea = document.getElementById('fileDisplayArea');
4
5      // On "écoute" si le fichier donné a été modifié.
6      // Si on a donné un nouveau fichier, on essaie de le lire.
7      fileInput.addEventListener('change', function(e) {
8          // Dans le HTML (ligne 22), fileInput est un élément de tag "input" avec un
          // attribut type="file".
9          // On peut récupérer les fichiers données avec le champs ".files" au niveau
          // du javascript.
10         // On peut potentiellement donner plusieurs fichiers,
11         // mais ici on n'en lit qu'un seul, le premier, donc indice 0.
12         let file = fileInput.files[0];
13         // on utilise cette expression régulière pour vérifier qu'on a bien un
          // fichier textString.
14         let textType = new RegExp("text.*");
15
16         if (file.type.match(textType)) { // on vérifie qu'on a bien un fichier
          textString
17             // lecture du fichier. D'abord, on crée un objet qui sait lire un fichier.
18             var reader = new FileReader();
19
20             // on dit au lecteur de fichier de placer le résultat de la lecture
21             // dans la zone d'affichage du textString.
22             reader.onload = function(e) {
23                 fileDisplayArea.innerHTML = reader.result;
24             }
25
26             // on lit concrètement le fichier.
27             // Cette lecture lancera automatiquement la fonction "onload" juste
          // au-dessus.
28             reader.readAsText(file);
29
30             document.getElementById("logger").innerHTML = '<span
          class="infolog">Fichier chargé avec succès</span>';
31         } else { // pas un fichier textString : message d'erreur.
32             fileDisplayArea.innerHTML = "";
33             document.getElementById("logger").innerHTML = '<span
          class="errorlog">Type de fichier non supporté !</span>';
34         }
35     });
36 }
37 function segmentation() {
38
39     const texte = document.getElementById(`fileDisplayArea`).innerHTML;
40     if (texte.length === 0) {alert('Veuillez SVP choisir votre texte !');return}
41
42     const delim = document.getElementById(`delimID`).value;
43     if (delim.length === 0) {alert('Veuillez SVP indiquer vos délimiteurs de mots
          !');return}
44
45     mots = tokenisation(texte, delim);
46     nMots = mots.length;
47
48     // on met les mots dans un objet, pour calculer la fréquence des mots
49     freqMots = [];
50     mots.forEach (m => {if (freqMots[m]) {freqMots[m]++;} else {freqMots[m]=1;}});
51
52     // on met les mots dans un objet longueur de mots, pour y mettre la fréquence et
          // tous les mots ayant cette longueur
53     longMots = [];
54     for (let i=0;i<50;i++){longMots.push([i, 0, '', 0]);} // a priori, il ne devrait
          // pas y avoir des mots de plus de 50 caractères !
55
56     for (m in freqMots) {
57         lg=m.length;
58         if (lg>0) {
59             fr=freqMots[m]
60             longMots[lg][1] += fr;
61             longMots[lg][2] += m + ' ';
62             longMots[lg][3] ++;
63         }

```

```

64     };
65     // Début du remplissage de page-analysis que l'on commence par
    vider-----
66     pageAnalysis = document.getElementById("page-analysis");
67     pageAnalysis.innerHTML = "";
68
69     // ligne total :
70     total = document.createElement("h3");
71     total.textContent = "Le texte comprend " + nMots + " mots.";
72     pageAnalysis.appendChild(total);
73
74     // ligne titre du tableau :
75     titre = document.createElement("h3");
76     titre.textContent = "Liste des mots triés par longueur :";
77     pageAnalysis.appendChild(titre);
78
79     // tableau :
80     tableau = document.createElement("table"); tableau.border = "1";
81     entete = document.createElement("tr");
82     colonnes = ["Longueur", "Fréquence", "Mots"];
83     colonnes.forEach(c => {
84         colonne = document.createElement("th");
85         colonne.textContent = c;
86         entete.appendChild(colonne);
87     });
88     tableau.appendChild(entete);
89
90     // on remplit le tableau à partir de longMots :
91     longMots.forEach ( lg => {
92
93         if (lg[1]>0) {
94             longueur=lg[0];freq=lg[1];m=lg[2]
95             m = m.slice(0,m.length-2) + ' ('+ lg[3] + ')';
96             ligTable = document.createElement("tr");
97
98             cel = document.createElement("td"); cel.align="center"; cel.textContent
            = longueur; ligTable.appendChild(cel);
99             cel = document.createElement("td"); cel.align="center"; cel.textContent
            = freq; ligTable.appendChild(cel);
100            cel = document.createElement("td"); cel.textContent = " " + m;
            ligTable.appendChild(cel);
101
102            tableau.appendChild(ligTable);
103        }
104    });
105    pageAnalysis.appendChild(tableau);
106
107 }
108 function cooccurrences() {
109
110     const texte = document.getElementById(`fileDisplayArea`).innerText;
111     if (texte.length === 0) {alert('Veuillez SVP choisir votre texte !');return}
112
113     const delim = document.getElementById(`delimID`).value;
114     if (delim.length === 0) {alert('Veuillez SVP indiquer vos délimiteurs de mots
        !');return}
115
116     const poleID = document.getElementById(`poleID`).value;
117     if (poleID.length === 0) {alert('Veuillez SVP indiquer votre pôle !');return}
118
119     const lgID = document.getElementById(`lgID`).value;
120     if (lgID === '0') {alert('Veuillez SVP indiquer une longueur positive !');return}
121
122     mots = tokenisation(texte, delim);
123     if (!mots.includes(poleID)) {alert('Attention ! Votre pôle ne figure pas dans le
        texte !');return}
124
125     nMots = mots.length;
126
127     // structure de coocc : clé(mot), [fréquence à gauche, fréquence à droite]
128     coocc = [];
129     for (let i=0;i<nMots;i++){
130         if (mots[i]===poleID){

```

```

131         for (j=1;j<=lgID;j++) {
132             if (i-j>0) {
133                 m=mots[i-j]
134                 if (coocc[m]){coocc[m][0]++;} else {coocc[m]=[1,0]}
135             if (i+j<nMots) {
136                 m=mots[i+j]
137                 if (coocc[m]){coocc[m][1]++;} else {coocc[m]=[0,1]}
138             }
139         }
140
141         //pour trier selon la cofréquence décroissante, on transforme l'objet coocc en
142         //objet de tableaux
143         // structure de coocc_tri : mot, [cofréquence, fréquence à gauche, fréquence à
144         // droite]
145         coocc_tri=[]
146         for (m in coocc) {coocc_tri.push([m, coocc[m][0]+coocc[m][1], coocc[m][0],
147         coocc[m][1]]);}
148         coocc_tri.sort((a, b) => b[1] - a[1]);
149
150         // Début du remplissage de page-analysis que l'on commence par
151         vider-----
152         pageAnalysis = document.getElementById("page-analysis");
153         pageAnalysis.innerHTML = "";
154
155         // ligne total :
156         total = document.createElement("h3");
157         total.textContent = "Dans l'intervalle de " + lgID + " mot(s) autour du mot '" +
158         poleID + "'", il y a au total " + coocc_tri.length + " mots différents :";
159         pageAnalysis.appendChild(total);
160
161         // tableau :
162         tableau = document.createElement("table"); tableau.border = "1";
163         entete = document.createElement("tr");
164         colonnes = ["Cooccurrent(s)", "Co-fréquence", "Fréquence gauche", "%Fréquence
165         gauche", "Fréquence droite", "%Fréquence droite"];
166         colonnes.forEach(c => {
167             colonne = document.createElement("th");
168             colonne.textContent = c;
169             entete.appendChild(colonne);
170         });
171         tableau.appendChild(entete);
172
173         // on remplit le tableau à partir de coocc_tri :
174         coocc_tri.forEach ( c => {
175
176             m=c[0];cofreq=c[1];freq_g=c[2];freq_d=c[3];
177             p_freq_g = Math.round(100*freq_g/cofreq);p_freq_d=100-p_freq_g;
178
179             ligTable = document.createElement("tr");
180
181             cel = document.createElement("td"); cel.textContent = " "+m;
182             ligTable.appendChild(cel);
183             cel = document.createElement("td"); cel.align="center"; cel.textContent =
184             cofreq; ligTable.appendChild(cel);
185             cel = document.createElement("td"); cel.align="center"; cel.textContent =
186             freq_g; ligTable.appendChild(cel);
187             cel = document.createElement("td"); cel.align="center"; cel.textContent =
188             p_freq_g+"%"; ligTable.appendChild(cel);
189             cel = document.createElement("td"); cel.align="center"; cel.textContent =
190             freq_d; ligTable.appendChild(cel);
191             cel = document.createElement("td"); cel.align="center"; cel.textContent =
192             p_freq_d+"%"; ligTable.appendChild(cel);
193
194             tableau.appendChild(ligTable);
195         });
196         pageAnalysis.appendChild(tableau);
197     }
198
199     function tokenisation(texte, delim, mode = 1, casse = false) {
200
201         /* mode = 1 : segmentation à partir des séparateurs (délimiteurs)
202         mode = 2 : segmentation à partir des caractères autorisés dans les mots :
203         lettres, chiffres, caractères diacritiques (dont accentués et
204         spéciaux français) et caractères grecs */

```

```

191
192 // on remplace tous les caractères de fin de ligne ou tabulation (éventuellement
193 // à compléter) par des espaces
194 texte0 = texte.replace(RegExp("\\n|\\r|\\t"), "g"), ' ');
195
196 // on ajoute un espace derrière les apostrophes pour les garder dans le texte =>
197 // "s'", "l'", etc. compteront pour 2 caractères, "qu'" pour 3, etc.
198 texte0 = texte0.replace(/['']/g, "' ");
199
200 // d'après le paramètre en entrée 'casse' (true = sensible à la casse)
201 if (!casse) {texte0 =
202     texte0.toLowerCase()};
203
204 switch (mode) {
205     case 1 : return multipleSplit(texte0, delim);
206     case 2 : return texte0.match(/[''\wĀ-ṛA-ṛ]+/g);
207 }
208
209 function multipleSplit(textString, separators) {
210     /* je n'ai pas réussi à mettre au point un regex, à travers la fonction RegExp,
211     permettant de faire un split sur plusieurs séparateurs, ceci à cause des caractères
212     génériques;
213     j'ai donc développé une fonction sans
214     regex
215     */
216
217     let text0 = textString;
218
219     separators = separators.replace(' ', ','); // on terminera obligatoirement par le
220     séparateur espace
221     for (i=0; i<separators.length; i++) {text0=text0.split(separators[i]).join(' ');};
222
223     /* remplacement de toutes les séquences de plusieurs espaces par un seul espace,
224     suppression des espaces à gauche et à droite, puis split sur le caractère espace
225     */
226     words = text0.replace(/ +/g, ' ').trim().split(' ');
227
228     return words;
229 }

```