

```

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
9          if (document.getElementById("boutonAide").textContent === "Masquer l'aide") {
10             aide();
11
12             // Dans le HTML (ligne 22), fileInput est un élément de tag "input" avec un
13             // attribut type="file".
14             // On peut récupérer les fichiers données avec le champs ".files" au niveau
15             // du javascript.
16             // On peut potentiellement donner plusieurs fichiers,
17             // mais ici on n'en lit qu'un seul, le premier, donc indice 0.
18             let file = fileInput.files[0];
19             // on utilise cette expression régulière pour vérifier qu'on a bien un
20             // fichier textString.
21             let textType = new RegExp("text.*");
22
23             if (file.type.match(textType)) { // on vérifie qu'on a bien un fichier
24                 textString
25                 // lecture du fichier. D'abord, on crée un objet qui sait lire un fichier.
26                 var reader = new FileReader();
27
28                 // on dit au lecteur de fichier de placer le résultat de la lecture
29                 // dans la zone d'affichage du textString.
30                 reader.onload = function(e) {
31                     fileDisplayArea.innerHTML = reader.result;
32                 }
33
34                 // on lit concrètement le fichier.
35                 // Cette lecture lancera automatiquement la fonction "onload" juste
36                 // au-dessus.
37                 reader.readAsText(file);
38
39                 document.getElementById("logger").innerHTML = '<span
40                 class="infolog">Fichier chargé avec succès</span>';
41                 erreur('');
42
43             } else { // pas un fichier textString : message d'erreur.
44                 fileDisplayArea.innerHTML = "";
45                 document.getElementById("logger").innerHTML = '<span
46                 class="errorlog">Type de fichier non supporté !</span>';
47             }
48         });
49     }
50     function segmentation() {
51
52         const texte = document.getElementById(`fileDisplayArea`).innerHTML;
53         if (texte.trim().length === 0) {erreur('Veuillez SVP choisir un texte non vide !'
54         );return}
55
56         const delim = document.getElementById(`delimID`).value;
57         if (delim.length === 0) {erreur('Veuillez SVP indiquer vos délimiteurs de mots !'
58         );return}
59
60         erreur('');
61
62         if (document.getElementById("boutonAide").textContent === "Masquer l'aide") {aide
63         ()};
64
65         let mots = tokenisation(texte, delim);
66
67         let longMots = segmentationCalculs(mots);
68
69         // Début du remplissage de page-analysis que l'on commence par
70         // vider-----
71         pageAnalysis = document.getElementById("page-analysis");
72         pageAnalysis.innerHTML = "";

```

```

62 // ligne total :
63 total = document.createElement("h3");
64 total.textContent = "Le texte comprend " + nMots + " mots.";
65 pageAnalysis.appendChild(total);
66
67 // ligne titre du tableau :
68 titre = document.createElement("h3");
69 titre.textContent = "Liste des mots triés par longueur :";
70 pageAnalysis.appendChild(titre);
71
72 // tableau :
73 tableau = document.createElement("table"); tableau.border = "1";
74 entete = document.createElement("tr");
75 colonnes = ["Longueur", "Fréquence", "Mots"];
76 colonnes.forEach(c => {
77     colonne = document.createElement("th");
78     colonne.textContent = c;
79     entete.appendChild(colonne);
80 });
81 tableau.appendChild(entete);
82
83 // on remplit le tableau à partir de longMots :
84 for (lg of longMots.filter((lg) => lg[1] > 0)) {
85     longueur=lg[0];freq=lg[1];m=lg[2]
86     m = m.slice(0,m.length-2) + ' ('+ lg[3] + ')';
87     ligTable = document.createElement("tr");
88
89     cel = document.createElement("td"); cel.align="center"; cel.textContent =
90     longueur; ligTable.appendChild(cel);
91     cel = document.createElement("td"); cel.align="center"; cel.textContent = freq
92     ; ligTable.appendChild(cel);
93     cel = document.createElement("td"); cel.textContent = " " + m; ligTable.
94     appendChild(cel);
95
96     tableau.appendChild(ligTable);
97 };
98 pageAnalysis.appendChild(tableau);
99
100 }
101 function visuSegmentation() {
102
103     const texte = document.getElementById(`fileDisplayArea`).innerText;
104     if (texte.length === 0) {erreur('Veuillez SVP choisir votre texte !');return}
105
106     const delim = document.getElementById(`delimID`).value;
107     if (delim.length === 0) {erreur('Veuillez SVP indiquer vos délimiteurs de mots !'
108     );return}
109
110     erreur('');
111
112     let mots = tokenisation(texte, delim);
113
114     let longMots = segmentationCalculs(mots);
115
116     // Début du remplissage de page-analysis que l'on commence par
117     vider-----
118     pageAnalysis = document.getElementById("page-analysis");
119     pageAnalysis.innerHTML = "";
120
121     // ligne titre
122     titre = document.createElement("h2");
123     titre.textContent = "Répartition des mots par longueur :";
124     pageAnalysis.appendChild(titre);
125
126     // remplissage des données du graphique camembert
127     let etiq = [];
128     let s1 = [];
129     for (lg of longMots.filter((lg) => lg[1] > 0)) {
130         longueur=lg[0];freq=lg[1];
131         etiq.push(longueur);
132         s1.push(freq);
133     };
134 }

```

```

130     let data = {
131         labels: etiq,
132         series: s1
133     };
134
135     let options = {
136         width: 400,
137         height: 240
138     };
139
140     new Chartist.Pie("#page-analysis", data, options);
141
142 }
143 function cooccurrences() {
144
145     const texte = document.getElementById(`fileDisplayArea`).innerText.trim();
146     if (texte.length === 0) {erreur('Veuillez SVP choisir votre texte !');return}
147
148     const delim = document.getElementById(`delimID`).value;
149     if (delim.length === 0) {erreur('Veuillez SVP indiquer vos délimiteurs de mots !');return}
150
151     const poleID = document.getElementById(`poleID`).value;
152     if (poleID.length === 0) {erreur('Veuillez SVP indiquer votre pôle !');return}
153
154     const lgID = document.getElementById(`lgID`).value;
155     if (lgID === '0') {erreur('Veuillez SVP indiquer une longueur positive !');return}
156
157     let mots = tokenisation(texte, delim);
158     if (!mots.includes(poleID)) {erreur('Attention ! Votre pôle ne figure pas dans le
159     texte !');return}
160
161     erreur('');
162
163     if (document.getElementById("boutonAide").textContent === "Masquer l'aide") {aide
164     ()};
165
166     let coocc0 = cooccurrencesCalculs(mots, poleID, lgID);
167     let nFois = coocc0[0]; let coocc = coocc0[1];
168
169     // Début du remplissage de page-analysis que l'on commence par
170     vider-----
171     pageAnalysis = document.getElementById("page-analysis");
172     pageAnalysis.innerHTML = "";
173
174     // ligne total :
175     total = document.createElement("h3");
176     total.textContent = "Dans l'intervalle de " + lgID + " mot(s) autour du mot '" +
177     poleID + "' (" + nFois + " fois dans le texte), il y a au total " + coocc.length +
178     " mots différents :";
179     pageAnalysis.appendChild(total);
180
181     // tableau :
182     tableau = document.createElement("table"); tableau.border = "1";
183     entete = document.createElement("tr");
184     colonnes = ["Cooccurrent(s)", "Co-fréquence", "Fréquence gauche", "%Fréquence gauche",
185     "Fréquence droite", "%Fréquence droite"];
186     colonnes.forEach(c => {
187         colonne = document.createElement("th");
188         colonne.textContent = c;
189         entete.appendChild(colonne);
190     });
191     tableau.appendChild(entete);
192
193     // on remplit le tableau à partir de coocc :
194     coocc.forEach ( c => {
195
196         m=c[0];cofreq=c[1];freq_g=c[2];freq_d=c[3];
197         p_freq_g = Math.round(100*freq_g/cofreq);p_freq_d=100-p_freq_g;
198
199         ligTable = document.createElement("tr");
200
201         cel = document.createElement("td"); cel.textContent = " "+m; ligTable.

```

```

196     appendChild(cel);
197     cel = document.createElement("td"); cel.align="center"; cel.textContent =
cofreq; ligTable.appendChild(cel);
198     cel = document.createElement("td"); cel.align="center"; cel.textContent =
freq_g; ligTable.appendChild(cel);
199     cel = document.createElement("td"); cel.align="center"; cel.textContent =
p_freq_g+"%"; ligTable.appendChild(cel);
200     cel = document.createElement("td"); cel.align="center"; cel.textContent =
freq_d; ligTable.appendChild(cel);
201     cel = document.createElement("td"); cel.align="center"; cel.textContent =
p_freq_d+"%"; ligTable.appendChild(cel);
202     tableau.appendChild(ligTable);
203 });
204 pageAnalysis.appendChild(tableau);
205
206 }
207 function visuCooccurents() {
208
209     const texte = document.getElementById(`fileDisplayArea`).innerText;
210     if (texte.length === 0) {erreur('Veuillez SVP choisir votre texte !');return}
211
212     const delim = document.getElementById(`delimID`).value;
213     if (delim.length === 0) {erreur('Veuillez SVP indiquer vos délimiteurs de mots !'
);return}
214
215     const poleID = document.getElementById(`poleID`).value;
216     if (poleID.length === 0) {erreur('Veuillez SVP indiquer votre pôle !');return}
217
218     const lgID = document.getElementById(`lgID`).value;
219     if (lgID === '0') {erreur('Veuillez SVP indiquer une longueur positive !');return}
220
221     let mots = tokenisation(texte, delim);
222     if (!mots.includes(poleID)) {erreur('Attention ! Votre pôle ne figure pas dans le
texte !');return}
223
224     erreur('');
225
226     if (document.getElementById("boutonAide").textContent === "Masquer l'aide") {aide
()};
227
228     let coocc0 = cooccurrencesCalculs(mots, poleID, lgID);
229     let nFois = coocc0[0]; let coocc = coocc0[1];
230
231     if (coocc.length>10) {maxMots = 10;} else {maxMots = coocc.length;};
232
233     // Début du remplissage de page-analysis que l'on commence par
vider-----
234     pageAnalysis = document.getElementById("page-analysis");
235     pageAnalysis.innerHTML = "";
236
237     // ligne titre :
238     titre = document.createElement("h2");
239     titre.textContent = "Dans l'intervalle de " + lgID + " mot(s) autour du mot '" +
poleID + "' (" + nFois + " fois dans le texte), les " + maxMots + " mots les plus
fréquents";
240     if (coocc.length>10) {titre.textContent += " sur un total de " + coocc.length ;};
241     titre.textContent += " sont :";
242     pageAnalysis.appendChild(titre);
243
244     // légende (image intégrée)
245     legende = document.createElement("img");
246     legende.src = "../img/legende_visuCooccurents.jpg";
247     pageAnalysis.appendChild(legende);
248
249     // remplissage des données du graphique en barres
250     let etiq = [];
251     let s1 = []; let s2 = []; let s3 = [];
252     for(let i=0; i<maxMots; i++) {
253         c=coocc[i];
254         m=c[0];cofreq=c[1];freq_g=c[2];freq_d=c[3]
255
256         etiq.push(m);

```

```

257     s1.push(cofreq); s2.push(freq_g); s3.push(freq_d);
258 };
259
260 // Graphique barres
261 let data = {
262     labels: etiq,
263     series: [s1, s2, s3]
264 };
265
266 let options = {
267     width: 550,
268     height: 180
269 };
270
271 new Chartist.Bar("#page-analysis", data, options);
272
273 }
274 function segmentationCalculs(mots) {
275
276     nMots = mots.length;
277
278     // on met les mots dans un objet, pour calculer la fréquence des mots
279     freqMots = [];
280     mots.forEach (m => {if (freqMots[m]) {freqMots[m]++;} else {freqMots[m]=1;}});
281
282     // on met les mots dans un objet longueur de mots, pour y mettre la fréaunce et
283     // tous les mots ayant cette longueur
284     longMots = [];
285     for (let i=0;i<50;i++){longMots.push([i, 0, '', 0]);} // a priori, il ne devrait
286     // pas y avoir des mots de plus de 50 caractères !
287
288     for (m in freqMots) {
289         lg=m.length;
290         if (lg>0) {
291             fr=freqMots[m]
292             longMots[lg][1] += fr;
293             longMots[lg][2] += m + ', ';
294             longMots[lg][3] ++;
295         }
296     };
297
298     return longMots
299 }
300 function cooccurrencesCalculs(mots, pole, lg) {
301
302     let nMots = mots.length;
303
304     // structure de coocc0 : clé(mot), [fréquence à gauche, fréquence à droite]
305     let coocc0 = [];
306     let nFois = 0;
307     for (let i=0; i<nMots; i++){
308         if (mots[i]===pole){
309             nFois++;
310             for (let j=1;j<=lg;j++) {
311                 if (i-j>0) {
312                     m=mots[i-j]
313                     if (coocc0[m]){coocc0[m][0]++;} else {coocc0[m]=[1,0]}
314                 }
315                 if (i+j<nMots) {
316                     m=mots[i+j]
317                     if (coocc0[m]){coocc0[m][1]++;} else {coocc0[m]=[0,1]}
318                 }
319             }
320         }
321     };
322
323     //pour trier selon la cofréquence décroissante, on transforme l'objet coocc en
324     // objet de tableaux
325     // structure de coocc : mot, [cofréquence, fréquence à gauche, fréquence à
326     // droite] trié par cofréquence décroissante
327     let coocc=[];
328     for (m in coocc0) {coocc.push([m, coocc0[m][0]+coocc0[m][1], coocc0[m][0], coocc0[
329     m][1]]);}
330     coocc.sort((a, b) => b[1] - a[1]);
331
332     return [nFois, coocc]

```

```

325 }
326 function tokenisation(texte, delim, mode = 1, casse = false) {
327
328     /* mode = 1 : segmentation à partir des séparateurs (délimiteurs)
329     mode = 2 : segmentation à partir des caractères autorisés dans les mots :
330     ***** 3 à 4 fois plus rapide *****
331     lettres, chiffres, caractères diacritiques (dont accentués et
332     spéciaux français) et caractères grecs */
333
334     // on remplace tous les caractères de fin de ligne ou tabulation (éventuellement
335     à compléter) par des espaces
336     texte0 = texte.replace(RegExp("(\\n|\\r|\\t)","g"),' ');
337
338     // on ajoute un espace derrière les apostrophes pour les garder dans le texte =>
339     "s'", "l'", etc. compteront pour 2 caractères, "qu'" pour 3, etc.
340     texte0 = texte0.replace(/['']/g,"' ");
341
342     // d'après le paramètre en entrée 'casse' (true = sensible à la casse)
343     if (!casse) {texte0 = texte0.toLowerCase()};
344
345     switch (mode) {
346     case 1 : return multipleSplit(texte0, delim);
347     case 2 : return texte0.match(/['\wÁ-žA-ſ]+/g);
348     }
349 }
350 function multipleSplit(textString, separators, method = 1) {
351     /*method : 1 = avec regex
352     2 = sans regex ***** 20% plus rapide
353     ***** */
354
355     switch (method) {
356     case 1 :
357         const specChar= "[] ()-|.|?.*+^$";
358         let sep_r='[';
359         for (let s of separators) {
360             if (specChar.includes(s)) {sep_r += '\\';}
361             sep_r += s;
362         }
363         sep_r += ']+';
364         words = textString.split(RegExp(sep_r));
365         if (words[0]===')') {words.shift()};
366         if (words[words.length-1]===')') {words.pop()};
367     case 2:
368         separators = separators.replace(' ',''); // on terminera obligatoirement
369         par le séparateur espace
370         for (s of separators) {textString=textString.split(s).join(' ')};
371         /* remplacement de toutes les séquences de plusieurs espaces par un seul
372         espace,
373         suppression des espaces à gauche et à droite, puis split sur le caractère
374         espace */
375         words = textString.replace(/ +/g, ' ').trim().split(' ');
376     }
377     return words;
378 }
379 function aide() {
380
381     boutonAide = document.getElementById("boutonAide");
382     Global = document.getElementById("Global");
383     Aide = document.getElementById("Aide");
384
385     if (boutonAide.textContent==="Afficher l'aide") {
386         Global.style.display = "none";
387         Aide.style.display = "block";
388
389         boutonAide.textContent="Masquer l'aide";
390
391     } else {
392         Global.style.display = "block";
393         Aide.style.display = "none";
394
395         boutonAide.textContent="Afficher l'aide";
396     }
397 }

```

```
389
390     }
391 }
392 function erreur(msg, alerte = false) {
393
394     Erreur = document.getElementById("Erreur");
395     Erreur.innerHTML = '';
396
397     if (msg != '') {
398         if (!alerte) {Erreur.innerHTML = '<span class="errorlog">' + msg + '</span>';}
399         else {alert(msg);}
400     };
401 }
402
```