

```

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         mots = tokenisation(texte, delim);
66         nMots = mots.length;
67
68         // on met les mots dans un objet, pour calculer la fréquence des mots
69         freqMots = [];
70         mots.forEach (m => {if (freqMots[m]) {freqMots[m]++;} else {freqMots[m]=1;}});
71
72         // on met les mots dans un objet longueur de mots, pour y mettre la fréquence et
73         // tous les mots ayant cette longueur

```

```

62     longMots = [];
63     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 !

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

```

```

128
129 // on met les mots dans un objet, pour calculer la fréquence des mots
130 freqMots = [];
131 mots.forEach (m => {if (freqMots[m]) {freqMots[m]++;} else {freqMots[m]=1;}});
132
133 // on met les mots dans un objet longueur de mots, pour y mettre la fréquence et
134 // tous les mots ayant cette longueur
135 longMots = [];
136 for (let i=0;i<50;i++){longMots.push([i, 0, '', 0]);} // a priori, il ne devrait
137 // pas y avoir des mots de plus de 50 caractères !
138
139 for (m in freqMots) {
140     lg=m.length;
141     if (lg>0) {fr=freqMots[m];longMots[lg][1] += fr;}
142 }
143
144 // Début du remplissage de page-analysis que l'on commence par
145 // vider-----
146 pageAnalysis = document.getElementById("page-analysis");
147 pageAnalysis.innerHTML = "";
148
149 // ligne titre
150 titre = document.createElement("h2");
151 titre.textContent = "Répartition des mots par longueur :";
152 pageAnalysis.appendChild(titre);
153
154 // remplissage des données du graphique camembert
155 let etiq = [];
156 let s1 = [];
157 for (lg of longMots.filter((lg) => lg[1] > 0)) {
158     longueur=lg[0];freq=lg[1];
159     etiq.push(longueur);
160     s1.push(freq);
161 }
162
163 let data = {
164     labels: etiq,
165     series: s1
166 };
167
168 let options = {
169     width: 400,
170     height: 240
171 };
172
173 new Chartist.Pie("#page-analysis", data, options);
174
175 }
176
177 function cooccurrences() {
178
179     const texte = document.getElementById(`fileDisplayArea`).innerText.trim();
180     if (texte.length === 0) {erreur('Veuillez SVP choisir votre texte !');return}
181
182     const delim = document.getElementById(`delimID`).value;
183     if (delim.length === 0) {erreur('Veuillez SVP indiquer vos délimiteurs de mots !'
184 );return}
185
186     const poleID = document.getElementById(`poleID`).value;
187     if (poleID.length === 0) {erreur('Veuillez SVP indiquer votre pôle !');return}
188
189     const lgID = document.getElementById(`lgID`).value;
190     if (lgID === '0') {erreur('Veuillez SVP indiquer une longueur positive !');return}
191
192     mots = tokenisation(texte, delim);
193     if (!mots.includes(poleID)) {erreur('Attention ! Votre pôle ne figure pas dans le
194 texte !');return}
195
196     erreur('');
197
198     if (document.getElementById("boutonAide").textContent === "Masquer l'aide") {aide
199 ()};
200
201     let nMots = mots.length;

```

```

195
196 // structure de coocc0 : clé(mot), [fréquence à gauche, fréquence à droite]
197 let coocc0 = [];
198 let nFois = 0;
199 for (let i=0; i<nMots; i++){
200     if (mots[i]==poleID){
201         nFois++;
202         for (let j=1; j<=lgID; j++) {
203             if (i-j>0) {
204                 m=mots[i-j]
205                 if (coocc0[m]){coocc0[m][0]++} else {coocc0[m]=[1,0]}
206             }
207             if (i+j<nMots) {
208                 m=mots[i+j]
209                 if (coocc0[m]){coocc0[m][1]++} else {coocc0[m]=[0,1]}
210             }
211         }
212     }
213 }
214
215 //pour trier selon la cofréquence décroissante, on transforme l'objet coocc en
216 // objet de tableaux
217 // structure de coocc : mot, [cofréquence, fréquence à gauche, fréquence à
218 // droite] trié par cofréquence décroissante
219 let coocc=[];
220 for (m in coocc0) {coocc.push([m, coocc0[m][0]+coocc0[m][1], coocc0[m][0], coocc0[
221 m][1]]);}
222 coocc.sort((a, b) => b[1] - a[1]);
223
224 // Début du remplissage de page-analysis que l'on commence par
225 vider-----
226 pageAnalysis = document.getElementById("page-analysis");
227 pageAnalysis.innerHTML = "";
228
229 // ligne total :
230 total = document.createElement("h3");
231 total.textContent = "Dans l'intervalle de " + lgID + " mot(s) autour du mot '" +
232 poleID + "' (" + nFois + " fois dans le texte), il y a au total " + coocc.length +
233 " mots différents :";
234 pageAnalysis.appendChild(total);
235
236 // tableau :
237 tableau = document.createElement("table"); tableau.border = "1";
238 entete = document.createElement("tr");
239 colonnes = ["Cooccurrent(s)", "Co-fréquence", "Fréquence gauche", "%Fréquence gauche",
240 "Fréquence droite", "%Fréquence droite"];
241 colonnes.forEach(c => {
242     colonne = document.createElement("th");
243     colonne.textContent = c;
244     entete.appendChild(colonne);
245 });
246 tableau.appendChild(entete);
247
248 // on remplit le tableau à partir de coocc :
249 coocc.forEach ( c => {
250
251     m=c[0];cofreq=c[1];freq_g=c[2];freq_d=c[3];
252     p_freq_g = Math.round(100*freq_g/cofreq); p_freq_d=100-p_freq_g;
253
254     ligTable = document.createElement("tr");
255
256     cel = document.createElement("td"); cel.textContent = " "+m; ligTable.
257     appendChild(cel);
258     cel = document.createElement("td"); cel.align="center"; cel.textContent =
259     cofreq; ligTable.appendChild(cel);
260     cel = document.createElement("td"); cel.align="center"; cel.textContent =
261     freq_g; ligTable.appendChild(cel);
262     cel = document.createElement("td"); cel.align="center"; cel.textContent =
263     p_freq_g+"%"; ligTable.appendChild(cel);
264     cel = document.createElement("td"); cel.align="center"; cel.textContent =
265     freq_d; ligTable.appendChild(cel);
266     cel = document.createElement("td"); cel.align="center"; cel.textContent =
267     p_freq_d+"%"; ligTable.appendChild(cel);
268
269     tableau.appendChild(ligTable);
270 }
271 );
272 pageAnalysis.appendChild(tableau);

```

```

255 }
256
257 function visuCooccurents() {
258
259     const texte = document.getElementById(`fileDisplayArea`).innerText;
260     if (texte.length === 0) {erreur('Veuillez SVP choisir votre texte !');return}
261
262     const delim = document.getElementById(`delimID`).value;
263     if (delim.length === 0) {erreur('Veuillez SVP indiquer vos délimiteurs de mots !'
264 );return}
265
266     const poleID = document.getElementById(`poleID`).value;
267     if (poleID.length === 0) {erreur('Veuillez SVP indiquer votre pôle !');return}
268
269     const lgID = document.getElementById(`lgID`).value;
270     if (lgID === '0') {erreur('Veuillez SVP indiquer une longueur positive !');return}
271
272     mots = tokenisation(texte, delim);
273     if (!mots.includes(poleID)) {erreur('Attention ! Votre pôle ne figure pas dans le
274 texte !');return}
275
276     erreur('');
277
278     if (document.getElementById("boutonAide").textContent === "Masquer l'aide") {aide
279 ()};
280
281     let nMots = mots.length;
282
283     // structure de coocc0 : clé(mot), [fréquence à gauche, fréquence à droite]
284     let coocc0 = [];
285     let nFois = 0;
286
287     for (let i=0; i<nMots;i++){
288         if (mots[i]===poleID){
289             nFois++;
290             for (let j=1;j<=lgID;j++) {
291                 if (i-j>0) {
292                     m=mots[i-j]
293                     if (coocc0[m]){coocc0[m][0]++} else {coocc0[m]=[1,0]}
294                 }
295                 if (i+j<nMots) {
296                     m=mots[i+j]
297                     if (coocc0[m]){coocc0[m][1]++} else {coocc0[m]=[0,1]}
298                 }
299             }
300         }
301     }
302
303     //pour trier selon la cofréquence décroissante, on transforme l'objet coocc en
304     objet de tableaux
305     // structure de coocc : mot, [cofréquence, fréquence à gauche, fréquence à droite]
306     let coocc=[]
307     for (m in coocc0) {coocc.push([m, coocc0[m][0]+coocc0[m][1], coocc0[m][0], coocc0[
308 m][1]]);}
309     coocc.sort((a, b) => b[1] - a[1]);
310
311     if (coocc.length>10) {maxMots = 10;} else {maxMots = coocc.length;};
312
313     // Début du remplissage de page-analysis que l'on commence par
314     vider-----
315     pageAnalysis = document.getElementById("page-analysis");
316     pageAnalysis.innerHTML = "";
317
318     // ligne titre :
319     titre = document.createElement("h2");
320     titre.textContent = "Dans l'intervalle de " + lgID + " mot(s) autour du mot '" +
321 poleID + "' (" + nFois + " fois dans le texte), les " + maxMots + " mots les plus
322 fréquents";
323     if (coocc.length>10) {titre.textContent += " sur un total de " + coocc.length ;};
324     titre.textContent += " sont :";
325     pageAnalysis.appendChild(titre);
326
327     // légende (image intégrée)
328     legende = document.createElement("img");
329     legende.src = "../img/legende_visuCooccurents.jpg";
330     pageAnalysis.appendChild(legende);
331

```

```

320 // remplissage des données du graphique en barres
321 let etiq = [];
322 let s1 = []; let s2 = []; let s3 = [];
323 for(let i=0; i<maxMots; i++) {
324     c=coocc[i];
325     m=c[0];cofreq=c[1];freq_g=c[2];freq_d=c[3]
326
327     etiq.push(m);
328     s1.push(cofreq); s2.push(freq_g); s3.push(freq_d);
329 };
330
331 // Graphique barres
332 let data = {
333     labels: etiq,
334     series: [s1, s2, s3]
335 };
336
337 let options = {
338     width: 550,
339     height: 180
340 };
341
342 new Chartist.Bar("#page-analysis", data, options);
343
344 }
345 function tokenisation(texte, delim, mode = 1, casse = false) {
346
347     /* mode = 1 : segmentation à partir des séparateurs (délimiteurs)
348     mode = 2 : segmentation à partir des caractères autorisés dans les mots :
349     ***** 3 à 4 fois plus rapide *****
350     lettres, chiffres, caractères diacritiques (dont accentués et
351     spéciaux français) et caractères grecs */
352
353     // on remplace tous les caractères de fin de ligne ou tabulation (éventuellement
354     // à compléter) par des espaces
355     texte0 = texte.replace(RegExp("(\\n|\\r|\\t)","g"),' ');
356
357     // on ajoute un espace derrière les apostrophes pour les garder dans le texte =>
358     // "s'", "l'", etc. compteront pour 2 caractères, "qu'" pour 3, etc.
359     texte0 = texte0.replace(/[']/g,"' ");
360
361     // d'après le paramètre en entrée 'casse' (true = sensible à la casse)
362     if (!casse) {texte0 = texte0.toLowerCase()};
363
364     switch (mode) {
365         case 1 : return multipleSplit(texte0, delim);
366         case 2 : return texte0.match(/['\wÁ-žA-ſ]+/g);
367     }
368 }
369 function multipleSplit(textString, separators, method = 1) {
370     /*method : 1 = avec regex
371     2 = sans regex ***** 20% plus rapide
372     ***** */
373
374     switch (method) {
375         case 1 :
376             const specChar= "[] ()-|.|?+^$";
377             let sep_r='[';
378             for (let s of separators) {
379                 if (specChar.includes(s)) {sep_r += '\\';}
380                 sep_r += s;
381             }
382             sep_r += ']+';
383             words = textString.split(RegExp(sep_r));
384             if (words[0]==='') {words.shift()};
385             if (words[words.length-1]==='') {words.pop()};
386         case 2:
387             separators = separators.replace(' ',''); // on terminera obligatoirement
388             // par le séparateur espace
389             for (s of separators) {textString=textString.split(s).join(' ')};
390             /* remplacement de toutes les séquences de plusieurs espaces par un seul
391             espace,

```

```

385         suppression des espaces à gauche et à droite, puis split sur le caractère
386         espace */
387         words = textString.replace(/ +/g, ' ').trim().split(' ');
388     }
389     return words;
390 }
391 function aide() {
392
393     boutonAide = document.getElementById("boutonAide");
394     Global = document.getElementById("Global");
395     Aide = document.getElementById("Aide");
396
397     if (boutonAide.textContent=="Afficher l'aide") {
398         Global.style.display = "none";
399         Aide.style.display = "block";
400
401         boutonAide.textContent="Masquer l'aide";
402
403     } else {
404         Global.style.display = "block";
405         Aide.style.display = "none";
406
407         boutonAide.textContent="Afficher l'aide";
408
409     }
410 }
411 function erreur(msg, alerte = false) {
412
413     Erreur = document.getElementById("Erreur");
414     Erreur.innerHTML = '';
415
416     if (msg != '') {
417         if (!alerte) {Erreur.innerHTML = '<span class="errorlog">' + msg + '</span>';}
418         else {alert(msg);}
419     };
420 }

```