

```

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 nbsp = String.fromCharCode(160);
40
41     const texte = document.getElementById(`fileDisplayArea`).innerHTML;
42     const delim = document.getElementById(`delimID`).value;
43
44     if (texte.length > 0) {
45
46         // on remplace tous les caractères de fin de ligne ou tabulation (éventuellement
          // à compléter) par des espaces
47         texte0 = texte.replace(RegExp("\\n|\\r|\\t"), "g"), ' ');
48
49         // on ajoute un espace derrière les apostrophes pour les garder dans le texte =>
          // "s'", "l'", etc. compteront pour 2 caractères, "qu'" pour 3, etc.
50         texte0 = texte0.replace(/[']/g, "' ");
51
52         // en cas de besoin, on peut mettre en commentaires la ligne suivante
53         texte0 = texte0.toLowerCase();
54
55         mots = multipleSplit(texte0, delim); // voir explications fonction ci-dessous
56         nMots = mots.length;
57
58         // on met les mots dans un objet, pour calculer la fréquence des mots en cas de
          // besoin
59         freqMots = [];
60         mots.forEach (m => {if (freqMots[m]) {freqMots[m]++;} else {freqMots[m]=1;}});
61
62         // on met les mots dans un objet longueur de mots, pour y mettre la fréquence et
          // tous les mots ayant cette longueur

```

```

63     longMots = [];
64     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 !
65
66     for (m in freqMots) {
67         lg=m.length;
68         if (lg>0) {
69             fr=freqMots[m]
70             longMots[lg][1] += fr;
71             longMots[lg][2] += m + ' ';
72             longMots[lg][3] ++;
73         }
74     };
75
76     analyse = `Le texte comprend ${nMots} mots.`;
77
78     analyse += `\n\nListe des mots triés par longueur :`;
79
80     analyse += '\n+ ' + '-'.repeat(59) + '+';
81     analyse += '\n|Longueur|Fréquence|' + nbsp.repeat(18) + 'Mots' + nbsp.repeat(18)
    + '|';
82     analyse += '\n| ' + '-'.repeat(59) + '|';
83
84     /* pour construire un tableau bien colonné, j'ai développé une fonction
85     ajoutant des espaces insécables (Ascii 160) à gauche (nombres) ou à droite
    (texte),
86     car les espaces simples ( Ascii 32) sont "tassés" dans html;
87     j'ai également chois la police Courier New dans le fichiers css.
88     NB : je n'ai pas réussi à gérer un tableau HTML dans 'page-analysis' en
    javascript */
89
90     longMots.forEach ( lg => {
91         if (lg[1]>0) {
92             analyse += '\n|'+ ajoutCarac(lg[0], 7, 'gauche', nbsp) + nbsp + '|' +
    ajoutCarac(lg[1], 8, 'gauche', nbsp) + nbsp + '|' + nbsp;
93             tab_m=lg[2].split(' ');
94             m=''
95             for (i=0;i<tab_m.length;i++) {
96                 if (m.length+tab_m[i].length > 32) {
97                     analyse += ajoutCarac(m, 38, 'droite', nbsp) + nbsp + '|';
98                     m = '';
99                     analyse += '\n|'+ nbsp.repeat(8) + '|' + nbsp.repeat(9) + '|' +
    nbsp;
100                 };
101                 m+=tab_m[i]+' '+nbsp
102             }
103             m = m.slice(0,m.length-4) + ' (' + lg[3] + ')';
104             analyse+=ajoutCarac(m , 38, 'droite', nbsp) + nbsp + '|';
105         };
106     });
107     analyse += '\n+ ' + '-'.repeat(59) + '+';
108
109     let pageAnalysis = document.getElementById(`page-analysis`);
110     pageAnalysis.innerText = analyse
111
112 }}
113 function multipleSplit(textString, separators) {
114     /* je n'ai pas réussi à mettre au point un regex, à travers la fonction RegExp,
115     permettant de faire un split sur plusieurs séparateurs, ceci à cause des caractères
    génériques;
116     j'ai donc développé une fonction sans
    regex */
117
118     let text0 = textString;
119
120     separators = separators.replace(' ',''); // on terminera obligatoirement par le
    séparateur espace
121     for (i=0;i<separators.length;i++) {text0=text0.split(separators[i]).join(' ');};
122
123     /* remplacement de toutes les séquences de plusieurs espaces par un seul espace,
124     suppression des espaces à gauche et à droite, puis split sur le caractère espace
    */
125     words = text0.replace(/ +/g, ' ').trim().split(' ');

```

```
126
127     return words;
128
129 }
130 function ajoutCarac(texte, longueur, cote, carac) {
131
132     if (cote[0].toLowerCase() === 'g') {
133         texte0=carac.repeat(longueur)+texte;
134         texte0=texte0.slice(texte0.length-longueur);}
135     else {
136         texte0=texte+carac.repeat(longueur);
137         texte0=texte0.slice(0, longueur);}
138
139     return texte0;
140 }
```