

Pour ce devoir j'ai choisi le poème de Germain Nouveau, *La Déesse*

Question n°1 :

> let poem = `LA DEESSE

J'adore la Mythologie,
Sa science en fleurs, sa magie,
Ses Dieux... souvent si singuliers,
Et ses Femmes surnaturelles
Qui mêlent leurs noms aux querelles
Des peuples et des écoliers.

Cachés parfois dans les nuages,
Leurs noms luisent... sur nos voyages.
J'ai vu leurs temples phéniciens ;
Et je songe, quand bat la diane,
Involontairement à Diane
Battant les bois avec ses chiens.

Tenez, Madame, je l'adore
Pour une autre raison encore :
C'est qu'elle offre à tous les amants,
Pour leur Belle entre les plus belles,
Des compliments par ribambelles
Dans d'éternels rapprochements.

Car toutes, ce sont des Déeses,
Leur inspirant mille prouesses
Dans le présent et l'avenir,
Comme dans le passé... farouche ;
Je me ferai casser la... bouche
Plutôt que n'en pas... convenir !

J'ai utilisé la variable "let poem"
pour déclarer mon poème et son
titre (mis en majuscule)

Sur la capture d'écran ci-contre, seuls
les premiers paragraphes sont visible
car l

Question n°2 :

Dans cette deuxième partie on cherche d'abord les 5 mots les plus fréquents du poème.

```
> let mots = poem.toLowerCase().match(/\b\w+\b/g);
< undefined
> let frequency = {};
< undefined
> mots.forEach(mots => {
    frequency[mots] = (frequency[mots] || 0) + 1;});
< undefined
> let tris = Object.keys(frequency).sort((a, b) => frequency[b] -
    frequency[a]);
< undefined
> let top5 = 5;
  for (let i = 0; i < top5; i++) {
    console.log(`Mots: ${tris[i]}, Fréquence: ${frequency[tris[i]]}`);}
Mots: la, Fréquence: 27 VM2916:3
Mots: vous, Fréquence: 27 VM2916:3
Mots: de, Fréquence: 25 VM2916:3
Mots: je, Fréquence: 24 VM2916:3
Mots: est, Fréquence: 23 VM2916:3
< undefined
```

- a. Le poème est structuré sous forme de tableau, défini comme "mots".
- b. Nous initialisons "fréquence" en tant qu'ensemble vide pour stocker les données à venir.
- c. Nous utilisons la méthode "forEach" pour parcourir tous les éléments du tableau "mots".
- d. Les mots du poème sont ensuite classés en fonction de leur fréquence d'apparition, du plus fréquent au moins fréquent.
- c. Une boucle "for" est utilisée pour afficher les mots les plus fréquents, qui sont stockés dans un tableau. Ensuite, nous affichons les 5 premiers mots de ce tableau.

J'obtiens la liste de résultat suivante :

- 1) « la »
- 2) « vous »
- 3) « de »
- 4) « je »
- 5) « est »

On cherche ensuite à calculer la richesse lexicale du poème avec le nombre de mots unique ainsi que le nombre de mots total.

Pour déterminer le nombre de mots (total), nous avons deux options.

Premièrement, nous pouvons simplement utiliser "mots" pour afficher tous les éléments présents dans le tableau. Sinon, nous pouvons utiliser la ligne de code suivante : "let total = mots.length;", qui nous donnera le nombre total de mots.

```
> let total = mots.length;
< undefined
> mots
< (941) ['la', 'deesse', 'j', 'adore', 'la', 'mythologie', 'sa', 'science', 'en', 'fleurs', 'sa', 'magie', 'ses', 'dieux', 'souvent', 'si', 'singuliers', 'et', 'ses', 'femmes', 'surnaturelles', 'qui', 'm', 'lent', 'leurs', 'noms', 'aux', 'querelles', 'des', 'peuples', 'et', 'des', 'coliers', 'cach', 's', 'parfois', 'dans', 'les', 'nuages', 'leurs', 'noms', 'luisent', 'sur', 'nos', 'voyages', 'j', 'ai', 'vu', 'leur', 's', 'temples', 'ph', 'niciens', 'et', 'je', 'songe', 'quand', 'bat', 'la', 'diane', 'involontairement', 'diane', 'battant', 'les', 'bois', 'avec', 'ses', 'chiens', 'tenez', 'madame', 'je', 'l', 'adore', 'pour', 'une', 'autre', 'raison', 'encore', 'c', 'est', 'qu', 'elle', 'offre', 'tous', 'les', 'amants', 'pour', 'leur', 'belle', 'entre', 'les', 'plus', 'belles', 'des', 'compliments', 'par', 'ribambelles', 'dans', 'd', 'ternels', 'rapprochements', ...] i
  ▶ [0 ... 99]
  ▶ [100 ... 199]
  ▶ [200 ... 299]
  ▶ [300 ... 399]
  ▶ [400 ... 499]
  ▶ [500 ... 599]
  ▶ [600 ... 699]
  ▶ [700 ... 799]
  ▶ [800 ... 899]
  ▶ [900 ... 940]
  length: 941
  ▶ [[Prototype]]: Array(0)
> total
< 941
```

Le poème est donc composé de 941 mots (total)

Pour dénombrer les mots uniques, c'est-à-dire ceux qui n'apparaissent qu'une seule fois dans le texte, nous utilisons la boucle "for...in" pour parcourir notre tableau. En utilisant la condition "if ... === 1", nous vérifions si la fréquence est égale à 1. Si c'est le cas, cela signifie que le mot n'apparaît qu'une seule fois dans notre poème.

```
> for (let mots in frequency) {
    if (frequency[mots] === 1) {
        console.log(mots);
    }
}
```

Ci-contre, les premières occurrences de la liste de mots unique obtenu

deesse
mythologie
science
fleurs
magie
souvent
singuliers
femmes
surnaturelles
lent
querelles
peuples
coliers
cach
parfois
nuages
luisent
nos
voyages

Question n°3 :

Pour cette troisième partie, nous cherchons à connaître le nombre de phrase dans le poème

```
> let phrase = poem.match(/^[.!?]+[.!?]+/g);  
⏏ undefined  
> let nombrephrase = phrase ? phrase.length : 0;  
⏏ undefined  
> nombrephrase  
⏏ 96
```

Pour trouver les phrases dans notre poème, nous utilisons une expression régulière pour rechercher une séquence de caractères délimitée par les marqueurs suivants : «., !, ou ? ». Ensuite, nous utilisons une expression ternaire pour compter le nombre de phrases dans notre poème.

Le poème est composé de 11 phrases

Question n°4 :

Dans cette quatrième et dernière partie, nous voulons savoir de combien de strophe est composé le poème.

```
> let strophe = poem.trim().split('\n\n');  
⏏ undefined  
⏏  
> let total_strophe = strophe.length;  
⏏ undefined  
⏏  
> total_strophe  
⏏ 26
```

Ce code nous aide à identifier les strophes de notre poème. Nous commençons par supprimer tous les espaces en début et en fin de chaque chaîne. Ensuite, nous définissons que les strophes sont délimitées par les caractères « \n » et « \r » (obtenus après avoir mis notre poème sous forme de tableau). Enfin, nous comptons le nombre de strophes présentes.

Le poème est constitué de 26 strophes.

Pour finir, on va faire une typologie des strophes du poème en fonction :

- A) Du nombre de lignes des strophes
- B) Des vers en fonctions du nombre de syllabes

```
> let typestrophes = {};  
⏏ undefined  
⏏  
> let typesvers = {};  
⏏ undefined  
⏏  
> strophe.forEach((strophe, index) => {  
  let nombrelignes = (strophe.match(/\n/g) || []).length + 1;  
  typestrophes[`${Strophe ${index + 1}`}] = nombrelignes;  
  
  let syllabesparvers = strophe.split('\n').map(ligne => ligne.split(' ').reduce((acc, mot) => acc +  
    (mot.match(/[aeiouy]/ig) || []).length, 0));  
});  
⏏ undefined
```

Nous définissons d'abord des espaces vides appelés "typestrophes" et "typesvers". Ensuite, nous examinons chaque ligne de chaque strophe du poème pour compter le nombre de syllabes.

`(strophe.match(/\n/g) || []).length + 1;` → Utilise une expression régulière pour compter le nombre de ligne en recherchant les sauts de ligne. Puis on stocke tout cela dans notre ensemble vide « typestrophes ».

Ensuite, nous subdivisons le texte en lignes puis en mots afin de compter le nombre de syllabes. Cette information est ensuite stockée dans « syllabesparvers »

Enfin, nous parcourons tout ce qui se trouve dans le tableau "syllabesparvers" et transférons son contenu dans l'ensemble vide "typesvers" que nous avons créé précédemment.

Cependant lorsque j'appelle les variables « nombreligne » et « syllabesparvers », je reçois un message d'erreur me disant que ces variables n'ont pas été défini.

Je n'ai malheureusement pas pu corriger ce problème