

Générateur de Nuage de Mots

Cette application web permet d'analyser la fréquence des mots au sein d'un texte (saisi manuellement sur une zone de texte ou via un fichier .txt importé par l'utilisateur) et de générer une visualisation dynamique sous forme de nuage de mots circulaire.

1. Présentation du Projet

L'objectif est de fournir une interface intuitive où l'utilisateur soumet un texte. Le système traite les données pour éliminer les "mots vides" (donc les mots communs), calcule l'occurrence de chaque mot significatif, puis les dispose de manière esthétique dans un espace circulaire, en faisant varier la taille des mots selon leur occurrence.

2. Technologies / langages utilisés

Les langages utilisés sont :

- HTML 5 et CSS 3 pour la structure sémantique du site web ainsi que son design.
- JavaScript pour la gestion d'évènements, la lecture des fichiers en entrée, les appels API ainsi que les algorithmes de positionnement.
- PHP pour le nettoyage du texte (suppression des ponctuations et des mots-vides, tokenisation et comptage d'occurrences).
- JSON pour la communication entre le client et le serveur.

3. Flux de traitement et fonctionnement du programme

Le fonctionnement de l'application repose sur quatre étapes :

1. Entrée des données (par le client)

L'utilisateur entre des données soit dans la zone de texte directement ou soit par l'importation d'un fichier brut .txt (l'importation du fichier au la priorité par rapport à la zone de texte). Le script .js intercepte le texte lit le fichier et envoie les données brute au serveur grâce à une requête POST.

2. Analyse et nettoyage (côté serveur)

Le fichier .php effectue plusieurs tâches côté serveur :

- Il convertit les majuscules en minuscules et supprime les ponctuations
- Il effectue la comparaison avec une liste de mots-vides afin de ne garder que les mots porteurs de sens
- Il compte ensuite les occurrences et fait un tri décroissant
- Il effectue ensuite des créations de balises avec des données comme la taille des mots et sa couleur générée aléatoirement
- Le texte ainsi filtré est ensuite sauvegardé dans un fichier /sauvegardes/

3. Réponse et initialisation (côté serveur)

Le serveur renvoie un objet JSON qui contient les fréquences et le code HTML des mots. Le .js injecte ces mots dans #cloud.

4. Algorithme de placement (côté client)

La fonction placerMotsAvecCollision() utilise une stratégie de spirale d'Archimède qui suit trois règles simples :

- Chaque mot est placé au centre
- On vérifie si le mot entre en collision avec un autre mot ou avec les limites du canvas
- Et, si c'est le cas, on déplace le mot en spirale jusqu'à ce qu'il trouve un espace libre