

Test de développement informatique

1 Attendu

Le code doit être renvoyé sous la forme d'une archive tar. Une documentation doit indiquer les modalités d'installations (version de python/Node.js) pour lancer les différents programmes.

Le style et les commentaires sont très importants.

2 Exercice 1 – API NLP

2.1 Mise en bouche

Écrire un module avec les fonctions :

- `nbCharacters(str)` `int` qui revoie le nombre de caractères de la chaîne de caractères UTF8;
- `nbWords(str)` `int` qui renvoie le nombre de mots de la chaîne de caractères;
- `occurences(str)` `map(string => int)` qui renvoie une table d'association dont la clé et le mot et la valeur le nombre d'occurrences du mot dans le texte passé en paramètre.

Le module doit être complété par un fichier de tests unitaires (en utilisant le Framework `unittest`)

2.2 Créez une API

Utilisez le module précédent pour réaliser une API Rest HTTP qui renvoie les valeurs des fonctions définies précédemment

2.3 Extraction des mots clés

Enrichir l'API précédente pour inclure une méthode `keywords` qui renvoie les mots clés importants en utilisant l'algorithme Yake (Yet Another Keyword Extractor) voir <https://github.com/LIAAD/yake> En entrée, la méthode doit prendre le texte et le nombre de mots-clés souhaités, en sortie elle doit renvoyer une liste triée par ordre croissant de pertinence de tableaux associatifs. En suivant ce schéma.

```
[
  { keyword: 'mot1', relevance: 0.023},
  { keyword: 'mot2', relevance: 0.4},
  { keyword: 'mot3', relevance: 0.9},
]
```

3 Exercice 2 – client Node.js

Le but est d'écrire un client en Javascript (Node.js) un client pour l'API que vous avez écrit. Le programme doit prendre un texte en ligne de commande, appelle l'API. Pour aller plus loin, écrivez un programme pour visualiser un nuage de mots en Javascript.