

Exercice 1

L'un des principaux problèmes du dictionnaire `occurrences` est qu'il intègre de la ponctuation.

1. Combien y a-t-il de questions dans le roman ?
2. La méthode `isalpha()` renvoie `True` si un texte ne contient que des lettres, et `False` sinon. Tester sur les entrées suivantes :

a) "bonjour". <code>isalpha()</code>	a) "aubépines.". <code>isalpha()</code>
b) "Aujourd'hui". <code>isalpha()</code>	b) "!". <code>isalpha()</code>
3. Créer un dictionnaire `occurrences_2`, qui recense les « vraies » occurrences des mots, c'est-à-dire sans prendre en compte les ponctuations parasites.

Exercice 2

Dans le texte d'origine, deux types de mots peuvent commencer par une majuscule : les noms propres, et les noms en début de phrase. Déterminer la liste de tous les noms propres apparaissant dans l'œuvre de Proust.

La méthode `isupper()` renvoie `True` si un texte ne contient que des majuscules (même syntaxe que pour `isalpha()`).

Exercice 3

Pour mieux quantifier la loi de Zipf, on cherche à écrire la fonction tracée sous la forme $f : x \mapsto \frac{a}{x} + b$. La bibliothèque `numpy` contient la fonction `polyfit`, qui permet de faire de la régression polynomiale, c'est-à-dire qu'elle permet de trouver un polynôme qui s'approche le mieux de la courbe fournie. La syntaxe est `polyfit(A, B, 1)`, avec `A` les abscisses, `B` les ordonnées et `1` spécifie le degré du polynôme (ici ...degré 1). Elle renvoie un tableau des coefficients du polynôme en question : ici, si `t = polyfit(A, B, 1)`, `t[0]` sera l'ordonnée à l'origine et `t[1]` la pente.

Tracer, par-dessus la courbe de la loi de Zipf, une hyperbole qui s'en rapproche.

Exercice 4 : chiffre de Vigenère

Il est recommandé d'avoir fait le chiffre de César du TP02 pour faire cet exercice.

Le chiffre de Vigenère est une technique de cryptographie pour encoder du texte. Dans le chiffre de César, on choisit un décalage uniforme; dans le chiffre de Vigenère, le décalage varie selon un mot choisi à l'avance, appelé « clé ». Par exemple, prenons comme clé le mot `avaient` : on commence par un décalage selon la première lettre `a`, qui correspond à un décalage de 0; puis selon la lettre `v`, qui correspond à un décalage de 17; etc. Quand on arrive au bout de clé, on recommence; la clé est utilisée de façon périodique.

1. Écrire une fonction `vigenere_encoder(texte, i)` qui renvoie le chiffrement de `texte` selon la clé `oeuvre_integrale[i]`. Par exemple, `oeuvre_integrale[5437]` est censé valoir `avaient`, et l'encodage de « *je suis descendu ce matin chez mon medecin hermenogene* » (*Mémoires d'Hadrien*, de Marguerite Yourcenar) est censé renvoyer "jz ayvl yeagrgdp ki faoiv paeu usa mzdmvgv cezqbzeie".
2. Toujours avec la clé `avaient`, déchiffrer le texte "ig tihk vvimg szmjpr a twyf lzs xehin yyr c ebevm pnm ohnie mqxe d epaeoez px xhmzne".