

Un enseignement au numérique pour notre académie : La recherche d'un mot dans un texte

Le numérique, un moyen de re-convoquer l'une de nos valeurs républicaines

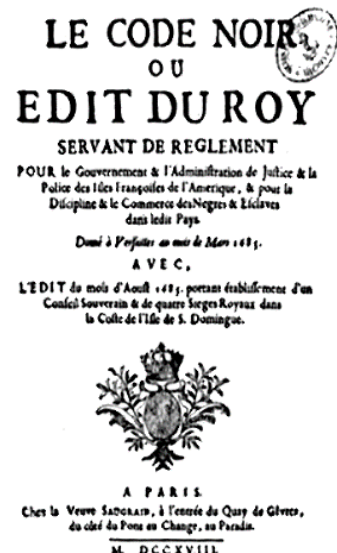
Jean-Alain Roddier
IA-IPR de mathématiques

La recherche d'un mot dans un texte est une capacité essentielle dont dispose le numérique, capacité que nous utilisons au quotidien. Si nous lançons la requête **esclavage** sur un moteur de recherche, le simple fait de taper les premières lettres de ce mot, nous donne au fur et à mesure des propositions de mots qui contiennent ces lettres. Une fois le mot **esclavage** entièrement marqué dans la fenêtre des requêtes, le moteur de recherche nous renvoie des pages web dans lequel figure en effet le mot **esclavage**. L'algorithme de recherche d'un mot dans un texte est lancé de façon tacite à chaque fois que nous cherchons le nom d'une personne dans un répertoire, le nom d'un fichier dans un dossier, etc. ; il s'agit donc bien de l'omniprésence d'un algorithme dans notre quotidien, algorithme sur lequel nous nous proposons de lever le voile.

Nous choisissons à séant le mot **esclavage** dans l'écriture de cet article pour montrer ici comment la formation au numérique peut constituer un moyen de re-convoquer l'une de nos valeurs républicaines.

Un adjectif associé au mot *esclavage* était jusqu'à présent l'adjectif *moderne*, on parlait ainsi d'*esclavage moderne* pour signifier que telle ou telle entreprise nous semblait enfreindre le code du travail. Aujourd'hui, un fait marquant de l'année 2017 a été de nous montrer un *esclavage contemporain* qui applique le code noir, code tristement célèbre qui réglementait au dix-huit siècle la vie des esclaves dans les Antilles françaises.

Page de garde d'un exemplaire du Code noir
Source : Wikipédia Creative Commons
Auteur de la photographie : Ambre Troizat

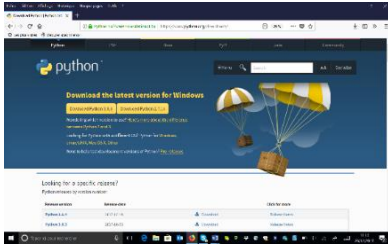


Revenons au numérique, pour nous intéresser à la recherche d'un mot dans un texte et pour savoir comment notre ordinateur fait pour analyser des textes et savoir si Oui ou Non tel ou tel mot y figure.

Nous savons que le numérique procède souvent de façon laborieuse et que c'est finalement sa rapidité de traitement qui le rend efficace. Cela va encore se vérifier en ce qui concerne la recherche d'un mot dans un texte.

Afin de rendre nos explications concrètes, nous choisissons de développer un algorithme de recherche du mot « esclavage » dans un texte très court (une phrase) puis dans un texte plus long. Ce développement va se faire avec le logiciel Python que nous vous proposons – dans un premier temps - de télécharger.

Téléchargement du logiciel Python



Le lien suivant : <https://www.python.org/downloads/> donne accès à la page de téléchargement du logiciel Python. Nous choisissons de télécharger la dernière version 3.6.4.

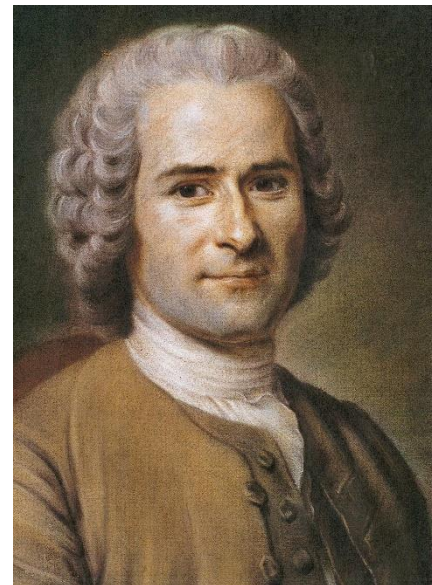
L'algorithme de recherche d'un mot

Soit à chercher si le mot « esclavage » est présent ou non dans le texte suivant :

« C'est la force et la liberté qui font les excellents hommes. La faiblesse et l'esclavage n'ont fait jamais que des méchants. »

Jean-Jacques Rousseau, les rêveries du promeneur solitaire.

Jean-Jacques Rousseau (1712-1778)
Tableau de Maurice-Quentin de la Tour
Source : site Wikipédia



La notion de chaîne de caractères

Pour le logiciel Python, le mot « esclavage » est une chaîne de caractères autrement dit des caractères mis bout à bout qui constituent ni plus ni moins une chaîne.

```
>>> mot="esclavage"  
>>> type(mot)  
<class 'str'>
```

Si l'on définit ainsi le mot « esclavage » et que l'on demande à Python quel est la nature de mot, celui-ci nous renvoie str qui est l'abréviation du string (chaîne de caractères).

De la chaîne de caractères à la liste de caractères

Le problème d'une chaîne de caractères réside dans le fait qu'une chaîne ne peut être construite maillon par maillon, on doit la construire - tout le temps - dans son entier. En revanche, une liste est beaucoup plus pratique car elle peut être construite - par exemple - en partant d'une liste vide et en ajoutant en fin de liste des éléments de façon successive.

Le logiciel Python a tout prévu car il permet de transformer une chaîne de caractères en liste. Transformons ainsi notre mot en liste. Pour ce faire, il suffit d'utiliser la fonction `list` disponible sous Python et de l'appliquer à la variable `mot`. On définit ainsi la liste `L`.

```
>>> L = list(mot)
>>> L[2]
'c'
```

Attention : les indices des listes commencent en effet au numéro zéro. L'élément de la liste `L` de rang numéro 2 est la lettre « c ».

Nous pourrions faire la même chose pour n'importe quelle phrase que nous souhaiterions analyser. Pour ce faire, **nous pouvons copier-coller dans la fenêtre du logiciel Python la phrase à analyser**. Nous pouvons ensuite transformer la phrase à analyser en liste et demander ensuite l'élément d'indice 50 qui est la lettre `t`.

```
>>> phrase="C'est la force et la liberté qui font les excellents
>>> LL=list(phrase)
>>> LL[50]
't'
```

Concernant les copier-coller effectués dans la fenêtre Python, lorsque l'on a à analyser un texte il est donc inutile de le retaper. Il est cependant possible que le logiciel bute sur les retours à la ligne créés lorsque l'on part d'un document en PDF. Il sera ainsi utile de veiller soit à effacer tous les retours chariots apparus à notre insu lors du copier-coller, soit à effectuer la recherche paragraphe par paragraphe.

L'algorithme proprement dit

Nous allons définir une fonction nommée `recherche`, fonction de deux variables :

- L'une de ces variables est le mot à chercher ;
- L'autre variable est le texte dans lequel nous devons chercher le mot.

La fonction `recherche` va créer avec les lettres du texte à analyser une succession de listes de même longueur que la liste obtenue avec le mot à chercher. Ainsi dans l'exemple où il s'agit de chercher le mot « esclavage » (mot de 9 lettres) dans la phrase de Jean-Jacques Rousseau, la fonction `recherche` crée les listes de 9 éléments suivantes :

```

['C', ' ', 'e', 's', 't', ' ', 'l', 'a', ' ']
[' ', 'e', 's', 't', ' ', 'l', 'a', ' ', 'f']
['e', 's', 't', ' ', 'l', 'a', ' ', 'f', 'o']
['s', 't', ' ', 'l', 'a', ' ', 'f', 'o', 'r']
['t', ' ', 'l', 'a', ' ', 'f', 'o', 'r', 'c']
[' ', 'l', 'a', ' ', 'f', 'o', 'r', 'c', 'e']
['l', 'a', ' ', 'f', 'o', 'r', 'c', 'e', ' ']
['a', ' ', 'f', 'o', 'r', 'c', 'e', ' ', 'e']
[' ', 'f', 'o', 'r', 'c', 'e', ' ', 'e', 't']
['f', 'o', 'r', 'c', 'e', ' ', 'e', 't', ' ']
['o', 'r', 'c', 'e', ' ', 'e', 't', ' ', 'l']
['r', 'c', 'e', ' ', 'e', 't', ' ', 'l', 'a']
['c', 'e', ' ', 'e', 't', ' ', 'l', 'a', ' ']

...

['f', 'a', 'i', 't', ' ', 'q', 'u', 'e', ' ']
['a', 'i', 't', ' ', 'q', 'u', 'e', ' ', 'd']
['i', 't', ' ', 'q', 'u', 'e', ' ', 'd', 'e']
['t', ' ', 'q', 'u', 'e', ' ', 'd', 'e', 's']
[' ', 'q', 'u', 'e', ' ', 'd', 'e', 's', ' ']
['q', 'u', 'e', ' ', 'd', 'e', 's', ' ', 'm']
['u', 'e', ' ', 'd', 'e', 's', ' ', 'm', 'é']
['e', ' ', 'd', 'e', 's', ' ', 'm', 'é', 'c']
[' ', 'd', 'e', 's', ' ', 'm', 'é', 'c', 'h']
['d', 'e', 's', ' ', 'm', 'é', 'c', 'h', 'a']
['e', 's', ' ', 'm', 'é', 'c', 'h', 'a', 'n']
['s', ' ', 'm', 'é', 'c', 'h', 'a', 'n', 't']
[' ', 'm', 'é', 'c', 'h', 'a', 'n', 't', 's']
['m', 'é', 'c', 'h', 'a', 'n', 't', 's', ' ']

```

Chacune de ces listes va être méticuleusement comparée à celle du mot à chercher pour notre exemple il s'agit ainsi de chercher si parmi toutes les listes ci-dessus figure ou non la liste ['e','s','c','l','a','v','a','g','e']. Notre algorithme renverra true uniquement si l'une de ces listes lui est identique. Par sécurité, il conviendra une fois la recherche du mot « esclavage » effectuée de lancer une recherche sur le mot « Esclavage » avec cette fois-ci la lettre « e » majuscule.

L'algorithme développé en langage Python

Nous concevons parfaitement la difficulté pour une personne non initiée de tout ce que nous venons d'expliquer aussi nous allons reprendre tout cela dans le détail en décortiquant point par point le programme que nous allons utiliser.

Voici le programme :

```

>>> def recherche(mot, texte):
    L=list(mot)
    LL=list(texte)
    test=[]
    booleen='false'
    for i in range(len(LL)-len(L)+1):
        for j in range(len(L)):
            test.append(LL[i+j])
        if L==test:
            booleen='true'
        test=[]
    return(booleen)

```

Le tableau suivant reprend le programme morceau par morceau avec pour chaque morceau des explications détaillées.

Morceau du programme	Explications
<pre> >>> def recherche(mot, texte):</pre>	Cette ligne nous permet de définir la fonction recherche d'arguments mot et texte qui sont pour l'instant des chaînes de caractères.
<pre> L=list(mot) LL=list(texte)</pre>	On transforme nos variables mot et liste en des listes nommées L pour celle du mot à chercher et LL pour celle du texte dans lequel nous allons rechercher le mot.
<pre> test =[]</pre>	La variable test est une liste qui au départ est vide. Cette variable va contenir successivement les listes de lettres que nous allons comparer à la liste du mot à chercher.
<pre> booleen='false'</pre>	La variable booleen est affectée au départ de la valeur 'false', elle prendra la valeur 'true' uniquement si l'une au moins de ces listes est identique à celle du mot cherché.
<pre> for i in range(len(LL)-len(L)+1): for j in range(len(L)): test.append(LL[i+j]) if L==test: booleen='true' test=[]</pre>	Cette boucle est le cœur du programme. La variable i va nous permettre de repérer les caractères du texte : <u>au départ i prend la valeur 0</u> et si l'on se réfère à notre exemple, la liste test va être construite progressivement – grâce à l'indice j - pour arriver à donner : ['C', '', 'e', 's', 't', ' ', 'l', 'a', ' '] On effectue ensuite un test pour savoir si cette liste est identique ou pas à notre liste L. La liste test est réinitialisée à une liste vide. <u>Puis i prend la valeur 1</u>, la liste test va être construite progressivement pour arriver à donner : ['', 'e', 's', 't', ' ', 'l', 'a', ' ', 'f'] On effectue ensuite le test et la liste test est réinitialisée à une liste vide. ...
<pre> return(booleen)</pre>	À la fin du programme, le programme renvoie la valeur de la variable booleen qui « sera passée » à true uniquement si le mot cherché a été rencontré dans notre texte.

Le lancement de notre fonction s'effectue en tapant le nom de la fonction puis en marquant entre parenthèses le mot à chercher et le texte dans lequel le mot est à chercher.

```
>>> mot="esclavage"  
>>> texte="C'est la force et la liberté qui font les excellents hommes. La faiblesse et l'esclavage n'ont jamais fait que des méchants."  
>>> recherche(mot, texte)  
'true'
```

La technique mise à l'épreuve

Afin d'éprouver l'opérationnalité de notre fonction sur un texte plus long, nous vous proposons de l'appliquer à la recherche du mot « esclavage » dans l'article de Riss intitulé : « Tintin au Congo en Libye » (voir ci-dessous où vous trouverez l'intégralité du texte).

Cet article est paru dans le numéro 1323 en date du 29 novembre 2017 du journal CHARLIE HEBDO.



Photographie de Riss.
Source : Wikipédia Creative Commons
Auteur de la photographie : Romain Bréget

Riss nous a autorisé à utiliser de cette façon toute particulière son texte magistral. Pour le remercier, nous reprendrons simplement la première phrase de Rousseau :

« C'est la force et la liberté qui font les excellents hommes. ».

TINTIN AU CONGO EN LIBYE

par Riss

Parmi les milliers d'images d'actualité dont nous sommes abreuvés chaque semaine, il est souvent difficile de faire le tri entre le superflu, l'obscène et l'indispensable. Celles qui furent filmées par une journaliste de CNN montrant la vente d'esclaves en Libye appartiennent à la catégorie rare des images qui apportent de l'information et du sens. On y voit des silhouettes d'hommes originaires d'Afrique noire qui, tentant de migrer vers l'Europe, sont vendus comme esclaves par leurs geôliers libyens. Attendant leur sort dans l'obscurité, ces hommes écoutent la voix des acheteurs, qui renchérissent sur leur vie : « 400, 500, 700 ... » Adjugé, vendu ! Un être humain vaudrait donc autour de 350 euros. Un peu moins que le tableau de Léonard de Vinci vendu aux enchères 450 millions de dollars la semaine dernière. C'est logique, car ce qui est rare est cher, et le tableau de Léonard de Vinci est unique, alors que les Africains sont si nombreux. Il y en a tellement que, lorsque quelques 33 000¹ d'entre eux disparaissent au fond de la Méditerranée, on ne s'en aperçoit même pas. Le marché aux esclaves respecte les lois élémentaires du libéralisme économique.

Sur l'esclavage, on a beau avoir lu des livres et vu des films très bien faits avec des acteurs très convaincants, c'est seulement par notre imagination qu'on devait se représenter la vente d'êtres humains. Comme la pièce manquante d'un puzzle, ce film comble cette lacune. Avec cette vidéo, on croit entendre la bande-son d'une enchère quelque part en Louisiane ou aux Antilles au XIX^e siècle. Quelques chiffres, quelques

phrases, et votre destin ne vous appartient plus, votre existence devient la propriété d'un autre.

Vendus comme esclaves par des marchands libyens, ces hommes sont noirs. Dans la hiérarchie abjecte du racisme, l'homme noir est toujours placé en dessous de tout. C'est le principe même du racisme : trouver quelqu'un de plus bas que soi. Dans le sketch d'une humoriste célèbre, une fille avoue à sa mère qu'elle va se marier avec un homme noir. La mère inquiète demande à sa fille ; « *Il est noir-noir-NOIR ? Ou noir un peu, un peu blanc ?* ». La salle rit, mais rien n'indique si c'est au second degré.

« *Des hommes forts pour travailler dans les champs* », insiste, dans la vidéo, le vendeur pour motiver les enchérisseurs. La seule vertu reconnue à l'homme noir par le raciste, c'est sa force musculaire parce qu'elle le relie au monde animal. Jamais son intelligence, car elle le relierait à l'humanité. Originaire de Côte-d'Ivoire, le chanteur John William était en 1944 ajusteur-outilleur, quand il fut déporté au camp de Neuengamme. « *Un jour, un colonel SS est arrivé au camp. Et il est la Metalwerk où je travaillais. [...] c'était un endroit où l'on faisait vraiment de la mécanique de haute précision. [...] Je revois encore le colonel venir vers mon établi et me regarder avec perplexité. [...] On lui avait dit que les Africains étaient des hommes de Cro-Magnon, et il ne comprenait pas comment moi, en tant qu'Africain, j'arrivais à lire des plans industriels écrits en allemand et à travailler l'acier au centième de millimètre. Ça l'avait complètement suffoqué [...], [...] à chaque question qu'il posait, je poussais le zèle jusqu'à lui répondre en allemand sans attendre que le traducteur intervienne. Quand il est parti, il fallait le voir hocher la tête².* ». Les libyens qui se livrent à ce trafic d'esclaves auraient fait de bons SS à Neuengamme.

On apprend que le Ministre de la Culture veut remplacer le mot « nègre » utilisé pour désigner celui qui écrit un livre à la place de l'auteur par l'expression « prête-plume ». Pourquoi pas, si ça peut aider.

Et le mot « esclave », qui heurte notre conscience, par quoi le remplacer ? Si à la place on utilisait le mot « bouquin » ? « Viens voir, chérie, j'ai acheté deux " bouquins " : un pour faire le jardinage et un autre pour faire le ménage. Pourquoi « bouquin » ? Parce que 350 euros, le prix d'un esclave noir en Libye, c'est aussi le prix de cinq exemplaires de la Pléiade.

© Riss/Charlie hebdo

Tous droits réservés

1. 33 293 réfugiés morts entre le 1^{er} janvier 1993 et le 29 mai 2017, selon le Tagesspiegel du vendredi 10 novembre.

2. Noirs dans les camps nazis, de Serge Bilé (Le serpent à plumes, 2003).