



Recherche successive



1

Recherche INITIALE

Première étape de mon projet

Je cherche un parseur en python

- "Parseur python"
 - Liste d'outils et de librairies en python
 - <https://tomassetti.me/parsing-in-python/>
- "Parser library python"
 - Documentation python
 - <https://docs.python.org/3.7/library/parser.html>
 - Thread reddit
 - https://www.reddit.com/r/Python/comments/ltwl4g/good_parser_library_in_python/
 - Répertoire github
 - <https://github.com/lark-parser/lark>
- "Python GUI"
 - Top 10 Python GUI
 - <https://towardsdatascience.com/top-10-python-gui-frameworks-for-developers-adca32fbe6fc>



Je cherche un lexique des mots français

- "Online french dictionary"
 - Mauvaise recherche, n'a rien donné de concluant
- "All french words database online"
 - Blog sur une base de donnée
 - <https://ielanguages.com/blog/frequent-french-words-lexique/>
 - Subreddit sur les bases de données de mots français
 - https://www.reddit.com/r/datasets/comments/9z447q/dictionnar_y_of_all_french_words/
- "Correcteur en python"
 - Thread d'aide sur un correcteur en python
 - <https://webdevdesigner.com/q/spell-checker-for-python-129284/>
 - Article sur des correcteurs automatiques en python
 - <https://www.delftstack.com/fr/howto/python/python-spell-checker/>



Mots identifiés dans la recherche 1

ANTLR

Librairie de parsing pour java/python

parser

[Module](#) python

Lark-parser

Librairie de parsing entièrement en python, disponible sur [github](#)

Parsimonious

Mot identifié dans le répertoire de Lark, probablement une autre librairie

Nearly

Mot identifié dans le répertoire de Lark, a un répertoire [github](#)

Parsley

Trouvé dans le subreddit, à approfondir. [Docs](#)

Top 10 GUI

Site avec une liste de GUI, à approfondir

Lexique

Reviens sur le subreddit sur le lexique de mots, et sur l'article du blog <http://www.lexique.org/>

Peter Norvig

Lien sur le site d'aide, renvoie sur un tutoriel <http://norvig.com/spell-correct.html>



2

Seconde vague

À partir de la recherche initiale

Recherche à partir des mots

- Explorer le Top 1 GUI
 - Site de Kivy
 - <https://kivy.org/#home>
 - Site de PyForms
 - <https://pyforms.readthedocs.io/projects/Pyforms-GUI/en/v4/>
 - Site de PyQt
 - <https://riverbankcomputing.com/software/pyqt/intro>
 - Site de Tkinter
 - <https://wiki.python.org/moin/TkInter>
 - Déjà dans mon vocabulaire
- Explorer et comparer ces technos pour en choisir une seule



Recherche à partir des mots

→ ANTLR

- De base en Java, mais disponible en python
- Disponible en plugins sur pas mal d'IDE
- Grosse doc ici :
<https://github.com/antlr/antlr4/blob/4.6/doc/index.md>

→ Parser

- N'existe plus depuis Python 2.5, le module ast l'a remplacé

→ Lark

- Lien vers la doc :
<https://lark-parser.readthedocs.io/en/latest/>
- Semble simple d'utilisation
- Tutoriel pour créer un parseur de JSON :
https://github.com/lark-parser/lark/blob/master/docs/json_tutorial.md



Recherche à partir des mots

- Parsimonious
 - <https://github.com/erikrose/parsimonious>
- Nearly
 - Librairie de parsing en javascript
 - Lark peut utiliser ses grammaires
 - Doc : <https://github.com/kach/nearley>
- Parsley
 - Un autre parser, mais il peut créer des calculateurs
 - Doc : <https://parsley.readthedocs.io/en/latest/tutorial.html>
- Lexique
 - Base de donnée de tous les mots de la langue française
 - Lien vers OpenLexicon :
<https://github.com/chrplr/openlexicon>



Recherche à partir des mots

- Blog de Peter Norvig
 - Correction pas à 100% juste, se base sur des probabilités
 - Liens vers différentes implémentations dans d'autres langages



Mots identifiés dans la recherche 2

Grammar

Permet de créer le lexique pour la correction dans le parser

Tkinter

Très connu

PyQt

Facilement utilisable,
A une interface graphique directement utilisable

ANTLR

Très connu et utilisé,
difficile de trouver la difficulté d'utilisation

Lark

Semble facile à utiliser,
bien documenté

Parser

Maintenant asr, module de base de python, plus dur d'utilisation qu'une librairie

PyForms

Semble facile à prendre en main, bonne doc

Kivy

Ressemble à un mélange de HTML et CSS par moment.
Simple à mettre en place, a sa propre extension de fichier

Parlsey

Semble dur à coder, mais la possibilité d'un calculateur intégré est intéressante



Mots identifiés dans la recherche 2

OpenLexicon

Projet regroupant plusieurs bases de données lexicales

Langages

Différentes applications du correcteur en langages, très intéressant

Pandas

Librairie de traitement de fichier de base de données



3

Troisième vague

À partir de la recherche initiale

Approfondissement par question

→ Lark

- Code pour l'installation :
 - `$ pip install lark --upgrade`
- Surlignage de la syntaxe des fichiers .lark dans plusieurs IDE : Sublime Text, vscode, IntelliJ, PyCharm, Vim, Atom
- Exemple de code pour un "Hello World" :

```
→ from lark import Lark  
  
l = Lark("""start: WORD " , " WORD "!"  
  
    %import common.WORD // imports from terminal library  
    %ignore " " // Disregard spaces in text  
    """)  
  
print( l.parse("Hello, World!") )
```



Approfondissement par question

→ PyQt

- Sur le site internet, code pour installer PyQt5 :
 - `$ pip install PyQt5`
- Qt Designer est l'interface graphique pour construire des applications, exemple d'utilisation du code généré :

- ```
import sys
from PyQt5.QtWidgets import QApplication, QDialog
from ui_imagedialog import Ui_ImageDialog
```

```
app = QApplication(sys.argv)
window = QDialog()
ui = Ui_ImageDialog()
ui.setupUi(window)
```

```
window.show()
sys.exit(app.exec_())
```



# Approfondissement par question

## → Pandas

- Système pour accéder à des données stockées dans des tableaux ou des bases de données
- Installation de pandas :
  - `$ pip install pandas`
- Tutoriel pour commencer à utiliser pandas :
  - [https://pandas.pydata.org/docs/user\\_guide/10min.html](https://pandas.pydata.org/docs/user_guide/10min.html)





# Approfondissement par question

## → OpenLexicon

- Projet regroupant plusieurs lexiques dans plusieurs langues
- Exemples de scripts pour utiliser les bases de données :
  - <https://github.com/chrplr/openlexicon/blob/master/scripts/README.md>
- Ligne pour importer la base et la stocker localement en python :
  - ```
import pandas as pd  
lex = pd.read_csv('http://www.lexique.org/databases/Lexique383/Lexique383.tsv', sep='\t')  
lex.head()
```



Recherche successive



FIN