

# Revue des mêlées quotidiennes

(Ici nos mêlées se déroulent tous les 2/3 jours)

Nos mêlées ainsi que toutes nos communications se passent via Discord :

Sprint 3 :

Ici Liinq est Romain, lapersonnedu56 est Corentin et Mr Propre est Thibaut,

Nous nous renommerons pour le prochain rendu, ce sera plus simple.

**Liinq** 02/03/2021

yop, alors du coup j'ai modifié le readme pour que ça match avec le sprint 3, maintenant comme je disais faut qu'on corrige le sprint 2 parce que c'était pas fou et ensuite on attaque le xml (modifié)

**PDFTOTXT**

Projet par Romain Le Tasseur, Corentin Lecomte, Thibaut Québelle dans le cadre de l'UE Projet de développement.

Attention, ce programme est dans un but scolaire et n'est pas destiné à être utilisé par le grand public.

**Fonctionnalités:**

- Prise en charge de fichiers PDF.
- Prend en entrée un dossier contenant un ou des fichiers PDF et crée un sous-dossier pour y déposer les fichiers PDF convertis en TXT du même nom.
- Si un sous-dossier est déjà existant, il sera supprimé et remplacé par le nouveau.

**Comment l'utiliser:**

`python extracting_xml.py [chemin] [nom_dossier]`

Si le PDF cible n'est pas dans le répertoire du programme taper :

`python extracting_xml.py [chemin] [nom_dossier]`

**Syntaxe:**

```
[1] Pour convertir le PDF en un fichier TXT :  
python extracting_xml.py [chemin] [nom_dossier]  
[2] Pour convertir le PDF en un fichier XML sans la partie :  
extraction  
python extracting_xml.py [chemin] [nom_dossier] --no-extraction  
[3] Pour convertir le PDF en un fichier XML avec la partie :  
extraction  
python extracting_xml.py [chemin] [nom_dossier] --with-extraction
```

@Mr Propre pour en revenir a la conversion en xml, y'a pas un # extracting\_xml.py ...

**Liinq** 02/03/2021

jsp, y'a ptet des options disponibles avec PyPDF2 pour le xml

**Mr Propre** 02/03/2021

possible  
j'arrive pas a trouvé

★ **Liinq** a épinglé un message dans ce salon. Voir tous les messages épinglés.  
02/03/2021

**Liinq** 02/03/2021

je check de mon coté

**Mr Propre** 02/03/2021

yep

**Liinq** 02/03/2021

pour l'instant que trouve que pour du json

**lapersonnedu56** 02/03/2021

Alors

Ils ont fait une méthode pour les méta-données

Si y'a pas ce qu'il faut ou que y'a une corruption

Ils vont chercher directement dans le fichier

A grands coups de regex ect

Parce que évidemment ça serait trop simple si fallait juste les méta-données

**Liinq** 02/03/2021

mdr ah ouais

psk j'ai fait un sale

```
if title is None:  
    title = "None"
```

xD

pour éviter l'erreur

faut que je trouve un tuto regex (modifié)

**Mr Propre** 02/03/2021

mdrr

**Liinq** 02/03/2021

ah (modifié)

j'ai trouvé (modifié)

je suis sur une pister avec pdfminer pour le XML

faudra rajouter les options -t machin comme on a vu en système je pense


<https://www.youtube.com/watch?v=UQQsYXa1EHs>


YouTube


Kite


[5 Minute Tutorial] Regular Expressions (Regex) in Python


Regular Expressions


 **lapersonnedu56** 09/03/2021  
T'as le script  
Le problème c'est que si tu regardes le sous dossier txt tu vois que les sorties sont pas bonnes  
Genre y'a des nones dans certains type genre auteur ou titre  
Du coup va falloir qu'on rajoute le pdf2txt pour convertir le pdf en txt brut pour récupérer les infos dessus  
Donc la meilleure sortie c'est genre un moyen de séparer les infos du reste


 **Mr Propre** 09/03/2021  
ça marche pas avec les metadonnées ? (modifié)


 **lapersonnedu56** 09/03/2021  
Ça marche en partie  
Comme je l'ai dit on a des infos manquantes et le contenu est pas bon


 **Mr Propre** 09/03/2021  
ok ouais  
des fois il y a des infos dans les meta données et toi tu veux quand même le récupérer c'est ça ? (modifié)


 **lapersonnedu56** 09/03/2021  
Oui  
Donc on va transformer le pdf en txt  
Et traiter à grands coups de regex et autres  
Quand on obtient pas ce qu'on veut


 **Mr Propre** 09/03/2021  
je sais pas du tout comment marche regex, je me renseigne là dessus


 **lapersonnedu56** 09/03/2021  
C'est chiant ptdr  
En gros c'est pour faire une expression régulière sur ta chaîne de caractères


 **Mr Propre** 09/03/2021  
ouais j'ai lu ça


 **Liinq** 09/03/2021  
en vrai le regex c'est pas mal avec python  
j'avais trouvé un bon tuto regarde plus haut (modifié)


 **Mr Propre** 09/03/2021  
mais déjà tu passes ton pdf en txt avec pdf2txt, c'est ce qu'il y a de mieux ensuite le titre c'est souvent le premier truc de marquer


 **lapersonnedu56** 09/03/2021  
Faut un mix


 **Mr Propre** 09/03/2021  
et comme il y a deux \n entre chaque paragraphe


 **lapersonnedu56** 09/03/2021  
Mouais mais parfois il est pas reconnu par la lib qu'on utilise


 **Mr Propre** 09/03/2021  
et qu'on définit le premier paragraphe comme le titre  
tu prends du début à \n x2 et t'a le titre


 **lapersonnedu56** 09/03/2021  
C'est pas forcément la même recette pour tout mais c'est pas stupide en soi  
Vaut mieux un mix


 **Mr Propre** 09/03/2021  
si le titre est en premier, ça fonctionne


 **lapersonnedu56** 09/03/2021  
Ouais mais imagine si les auteurs sont mis en premiers  
Genre le haut de page


 **Mr Propre** 09/03/2021  
j'avoue que je connais pas du tout regex, je sais pas comment ça marche  
yep c'est pour ça que c'est la solution du pauvre


 **Liinq** 09/03/2021  
fais gaffe en pushant coco j'ai fait différentes branches  
je divise en sprints


 **lapersonnedu56** 09/03/2021  
Ok


 **Liinq** 09/03/2021  
là dcp faut faire une branche sprint 3


 **lapersonnedu56** 09/03/2021  
C'est la main du coup  
Quoi que nan  
D'ailleurs  
Si y'a dans les méta-données les références et les emails  
Faut rajouter


 **Mr Propre** 09/03/2021  
j'ai vérifié coco  
le titre est toujours le premier paragraphe et les auteurs toujours le deuxième

 **lapersonnedu56** 09/03/2021  
Mokay  
On traitera dessus alors  
Avec ton idée

 **Mr Propre** 09/03/2021  
si tu veux éviter les regex c'est ce qu'il y a de plus simple je pense  
surtout que pdf2txt fait ça propre


 **lapersonnedu56** 09/03/2021  
Regex c'est générique tho  
Et y'aura quand même besoin pour l'abstract je pense


 **Liinq** 09/03/2021  
oui pour l'abstract c'est indispensable


 **Mr Propre** 09/03/2021


```
#data - stuff between text tags
firstparagraph = re.search("}{(.*)\n*\n",data,re.DOTALL)
print firstparagraph.group(1)
```


c'est une connerie qui ressemble à ça pour sortir le premier paragraphe d'un text


 **Liinq** 09/03/2021  
pas mal  
avec une librairie spéciale ?


 **Mr Propre** 09/03/2021  
faut importer

 **Liinq** 09/03/2021  
wokay  
ah j'suis con mdr (modifié)  
j'avais pas vu le re (modifié)

 **lapersonnedu56** 09/03/2021  
Re c'est bon  
C'est la regex  
La lib de python pour gérer les regex

 **Mr Propre** 09/03/2021  
oui  
import regex as re

 **Liinq** 09/03/2021  
oui mb j'avais oublié le nom de la lib  
t'es sur Teams déjà Coco ?

 **Mr Propre** 09/03/2021

```
import regex as re
x = [re.search(r"^\n([^\n]*)\n", corpus).group(1).replace("\n", "")]
print(x)
```

- `^\n` find the first paragraph
- `([^\n]*)` tests for a string starting a new line (first line of paragraph).
- `^\n` match the string till next empty line (repeat previous regex till where the paragraph ends)

Ici il y a seulement deux mêlées datant du 2 et 9 mars car le reste a été réalisé par appel vocal.