

Exercice 04

	Exercice : 04	
	Exercice 04 : Parser du html	
	Dossier de rendu : <i>ex04/</i>	
	Fichiers à rendre : <code>roads_to_philosophy.py</code> , <code>requirement.txt</code>	
	Fonctions Autorisées : modules Requests et BeautifulSoup	
	Remarques : n/a	

La légende dit que : si vous partez d'un article quelconque de Wikipédia, que vous cliquez sur le premier lien dans l'introduction de cet article qui ne soit ni en italique ni entre parenthèse, puis que vous répétez ce processus en boucle, vous aboutirez invariablement sur l'article Philosophie.

Et bien sachez que ce n'est pas une légende ! (Faites 'Ooooooh!' avec un air abasourdi). En témoigne cet [article](#) de ... Wikipédia.

Cependant, comme vous ne croyez que ce que vous voyez de vos propres yeux, vous **devez** créer un programme qui met ce phénomène à l'épreuve en listant puis en comptant tous les articles entre votre requête et l'article Wikipédia : les **roads to philosophy**.

Ce programme doit se nommer `roads_to_philosophy.py` et avoir le comportement suivant :

- Le programme doit prendre en paramètre une string qui est un mot ou un ensemble de mots correspondant à une seule recherche Wikipédia.
- Le programme doit requêter une url du Wikipedia **anglais** identique à celle d'une recherche standard sur un navigateur. Autrement dit : il n'est **pas permis** d'utiliser l'API du site.
- Il doit parser la page html grâce à la librairie BeautifulSoup afin de :
 - Trouver l'éventuelle redirection, et la prendre en compte dans les **roads to philosophy**. Attention, il n'est pas question ici de redirection d'url.
 - Trouver le titre principale de cette page et l'ajouter aux **roads to philosophy**.
 - Trouver (s'il existe) **le premier lien du paragraphe d'introduction** conduisant à un autre article de Wikipédia. Plutôt que d'ignorer ce qui est en italique et entre

parenthèse, le programme doit **ignorer** soigneusement les liens qui ne pointent pas vers un nouvel article, tels que ceux qui menant à la section aide de Wikipédia.

- Le programme doit recommencer à partir de l'**étape 2** en partant du lien obtenu à l'étape précédente jusqu'à tomber sur un des cas suivant :
 - Le lien mène à la page philosophy, auquel cas il doit imprimer sur la sortie standard les articles visités ainsi que le compte total de ces articles sous le format <nombre> roads from <requête> to philosophy
 - La page ne contenait **aucun liens valides**, le programme doit afficher : It leads to a dead end !.
 - Le lien conduit à une page déjà visitée, ce qui signifie que le programme s'apprête à boucler indéfiniment. Dans ce cas le message affiché doit être : It leads to an infinite loop !
- A ce stade, après avoir affiché sur la sortie standard les messages nécessaires, le programme doit quitter proprement.



Si à n'importe quel moment dans l'exécution du programme, une erreur survient, telle qu'une erreur de connexion, de server, de paramètre, de requête ou autre : le programme doit quitter proprement avec un message d'erreur pertinent.

Comme pour l'exercice précédent, vous devrez fournir avec votre programme un fichier requirement.txt afin de faciliter l'installation des librairies.

La sortie de votre programme doit ressembler à celle-ci :

```
$> python3 roads_to_philosophy.py "43 (number)"
43 (number)
Natural number
Mathematics
Ancient Greek
Greek language
Modern Greek
Colloquialism
Word
Linguistics
Science
Knowledge
Awareness
Conscious
Consciousness
Quality (philosophy)
Philosophy
17 roads from 43 (number) to philosophy !
$> python3 roads_to_philosophy.py Accuvio
It's a dead end !
$>
```



La communauté Wikipédia met régulièrement ses articles à jour. Il est possible qu'entre la date de rédaction de ce sujet et celle à laquelle vous effectuez cette formation, l'exemple accuvio ne soit plus une impasse.