

Wikigame

Un TP de webscraping à écrire en Python 3 (obligé) avec BeautifulSoup (recommandé).

Ce TP est à réaliser seul.

Niveau : Développeurs Python confirmés (Bac + 3 / Bac + 4)

3 séances de cours seront consacrées au TP
Vous devrez le finir chez vous si besoin.

Le jeu

L'idée c'est de tirer 2 pages Wikipédia au sort, et d'aller de la 1er à la 2eme, uniquement en utilisant les liens hypertexte présent sur la page, en utilisant le moins de coups possible.

L'idée du jeu n'est pas de moi : il existe même une appli

<https://play.google.com/store/apps/details?id=com.armadill.thewikigame>

ou un championnat avec des flammes dans le générique !

<https://hooktube.com/watch?v=FdJN2iK0BSO>

Exemple :

Je part de la page [Python](#) et je doit aller à la page [République démocratique du Congo](#)

- Je choisi le lien Pays Bas¹
- Puis Afrique du Sud²
- Puis Afrique
- Et République démocratique du Congo

Gagné en 4 coups

Le programme que vous allez écrire sera un « habillage » pour ce jeu

1 Guido Van Rossum, l'inventeur du Python est natif des Pays Bas

2 Les Pays Bas sont les colons historiques de l'Afrique du Sud

Version 1

Dans un premier temps, vous devez écrire un jeu qui fonctionne, mais en « CLI brute »
C'est à dire, en mode texte sans fioritures d'affichage.

Votre affichage doit ressembler à ça :

```
***** WikiGame **** tour 1
Départ : Python (langage)
Cible : République démocratique du Congo
Actuellement : Pays-Bas
01 - Pays-Bas (homonymie)
02 - Hollande
03 - néerlandais
04 - Europe de l'Ouest
05 - Belgique
06 - Allemagne
07 - France
08 - île de Saint-Martin
09 - Caraïbes
10 - Monarchie constitutionnelle
11 - 2017
12 - Amsterdam
13 - pouvoirs exécutif
14 - législatif
15 - judiciaire
16 - La Haye
17 - territoires autonomes
18 - Aruba
19 - Curaçao
20 - Saint-Martin
99 - Voir la suite
Votre choix :
```

Explications :

Les 2 bornes (Page de départ et d'arrivée) sont générés grâce à l'URL « random » de Wikipedia :
https://fr.wikipedia.org/wiki/Sp%C3%A9cial:Page_au_hasard

Les choix 01 à 20 sont les 20 premiers liens hypertexte de la page (on affiche les liens par page de 20
Car certaines pages en ont plusieurs centaines !)

Attention, ne prendre que les liens de la partie centrale, pas ceux du menu de gauche ni du haut.
Ne pas prendre non plus les liens interne du sommaire

En cas de pagination on ajoute les liens :

98 : Page précédente

99 : Page suivante

Ces 2 liens ne sont afficher que si besoin (pas de page précédente sur la 1^{er} page, etc ..)
Ils ne changent pas le compteur de tour joués

A chaque tour, vous commencerez l'affichage par effacer l'écran (cls / Windows ; clear / Linux & Mac)³

³ Attention à détecter l'OS (je testerai votre code sous Linux)

Version 2

Une fois que votre jeu fonctionne, il faut l'embellir !

Vous pouvez soit rester en CLI (Comand Line Interface, ou « mode texte ») et vous améliorez le rendu avec des librairies comme [PyInquirer](#) ou autre

Soit vous passez en GUI (Graphic User Interface) et utilisez des librairies comme [TkInter](#) ou l'une de ses nombreuses alternative

Soit encore, vous utilisez une librairie comme [Fel](#) qui génère un serveur web et vous permet d'utiliser le HTML / CSS pour l'affichage

Soit ...

En cas de doute, je suis à votre écoute pour valider votre projet

Version 3

L'idée est d'enrichir le jeu avec des fonctionnalités.

Quelques idées en vrac (mais vous pouvez / devez ajouter les vôtres)

Attention : Vous ne devez pas toutes les développer : Ce ne sont que des suggestions.

Mode preview

Le programme affiche, au survol du lien, le résumé Wikipédia de la page

Mode Historique

En cas de victoire, le programme affiche l'historique des pages utilisées

Mode timer

Le programme ne compte pas les tours mais le temps.

Et s'arrête si le joueur dépasse n minutes (à paramétrer)

Mode thématique

Les URLs ne sont pas tirés complètement au sort, mais dans un corpus réduit (Jeux vidéos, Cinéma, ...)

Pour cela, utilisez les portails thématiques Wikipédia.

Ex : [Portail du jeu vidéo](#)

- Écrire un second programme qui vas scraper le portail pour en extraire une liste d'URLs qu'il va stocker dans un fichier texte.

- Ajouter un switch -d <fichier> au programme principal.

Si le switch -d est présent, ne pas tirer 2 pages au sort, mais les tirer au sort dans le fichier d'URLs passé en paramètre

Ex :

```
#python dicoMaker.py
Ok, fichier jeuvideo.txt généré
#python wikigame.py -d jeuvideo.txt
***** WikiGame **** tour 0
Départ : Body Harvest
Cible : Fortnite
Actuellement : Body Harvest
01 - jeu vidéo
02 - DMA Design
03 - Rockstar North
...
```

Mode duel

Un switch « -n » permet, non pas le lancer le jeu, mais de générer « une graine »

ex :

```
#python wikigame.py -n
lancer "python wikigame.py -s Andasta_benoiti#Henri_Durville"
```

Et si on lance le jeu avec le switch -s je jeu ne tire pas des pages au sort, mais lit les 2 URLs Wikipédia contenu dans le second paramètre (séparées par un #) et les utilise comme début & fin.

Fonctionnement :

Un joueur génère une graine (-n)

Il partage l'URL avec les autres (Discord, ...)

Tout les joueurs lancent le programme avec la graine (-s) en même temps et le 1^{er} arrivé à gagné.

Amélioration : La graine est encodé en base64 pour ne pas permettre aux joueurs de « voir » les URLs qu'elle contient

```
#python wikigame.py -n  
lancer "python wikigame.py -s QW5kYXN0YV9iZW5vaXRpI0hlbnJpX0R1cnZpbGxl"
```

Livrable

Un dépôt Git (GitLab (bien), GitHub (bouh) ...) avec un README.md qui explique comment installer le jeu (dépendances PIP, ...)

Pour ceux qui veulent garder leur dépôt privé : Ok, mais pensez à m'inviter.

Mon identifiant GitLab : <https://gitlab.com/arrobe>

Mon identifiant GitHub : <https://github.com/ArrobeHugues>

Vous n'avez pas à livrer 3 versions du jeu : Seule la plus aboutie (Version 3) est attendue

Barème

La version 1 est noté sur 7 points

Sera pris en compte :

- Le bon fonctionnement du jeu
- La qualité d'écriture du code (optimisé, commenté, ...)
- La résistance du jeu aux bugs, erreurs de saisie, ...

La version 2 est sur 7 points

Sera pris en compte :

- Rendu graphique
- Fluidité / facilité d'utilisation
- Ergonomie du jeu

La version 3 est sur 7 points⁴

Sera pris en compte :

- Richesse fonctionnelle (Fonctionnalités ajoutées)
- Qualité de la réalisation (nombre d'options ajoutées)
- Originalité des options ajoutées