# TEST TECHNIQUE: Partie 2/2

Coucou cher candidat,

Instructions:

- Pour les questions théoriques, écrire juste dans ce document, comme tu le sens.

- Installe comme tu veux les librairies necessaires pour la partie programmation.

(via un environement anaconda, soit directement, mais environement recommandé).

- Hesite pas à regarder sur le web pour des questionnements techniques (quelle fonction de quelle librairie utiliser, etc)

- Faire chaque bout de code dans un fichier indépendant, puis soit copier/coller

- Aucune tache ne devrait prendre trop de temps, le tout devrait être assez court.

------------------------- Question plutôt théoriques sur la technologie -------------------------

- Quelle techno doit on utiliser pour executer des taches/algos indépendant, en même temps, sur Python?

--> Est ce de la parallelisation?

--> Est ce de la conccurence?

----> si tu ne sais pas la différence, pas grave! :)

- Quelle techno doit on utiliser pour executer des taches sur plusieurs coeur du CPU?

Multiprocessing

--> avons nous besoin d'une librairie spéciale?

Oui nous pouvons utiliser la library « multiprocessing » en python

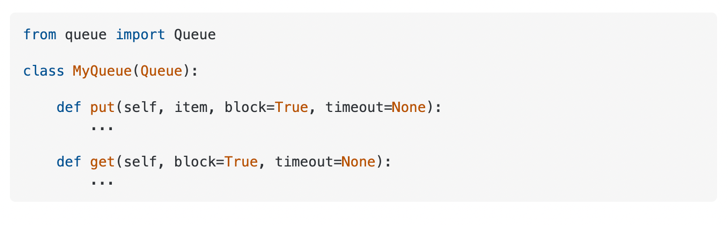
Notre logiciel est très "IO-intense" ((input/output), beaucoup de requetes HTTP, de download/uploads, etc).

Lequel est le plus important dans ce cas (multithreader ou multiprocesser)? (selon votre feeling)

Je dirai Multithreading car Multiprocessing est plus utilisé pour CPU bound

Quel autre axe d'amélioration voyez vous pour optimiser les performances? Si il y en a un.

Implémenter son propre système de Queue pour avoir sa propre logic, exemple de template de class :



Quelle sorte de mécanique est à mettre en place pour gérer les aléas I/O (pannes réseaux, rejections, etc)?

Gestion des erreurs (Try/Except). Gérer le maximum type d’erreur differente pour savoir quel problème il y a eu.

Comment ne jamais avoir un programme qui plante / ou tombe dans un état irrecupérable, selon toi? (selon tes mots)

Un programme qui plante comme je l’ai dit dans la question précédente, la gestion d’un maximum d’erreur ainsi que un maximum de testing pour chaque fonctionnalité implementé. Pour l’état dans un premier temps les backups et le versionning (gestion basique pour tout projet).

Quel est l'avantage de Python? Quels sont les possibles inconvénients? Est ce que c'est un problème pour nous?

Python est un language multi threads ce qui est ce que nous recherchons. Pour les autres avantages : dynamic binding donc améliore la rapidité des taches, langage fait pour le Machine Learning donc totalement dans notre projet. Ainsi que utilisable pour différent type de programmation  comme data science et software (beaucoup de Library pour le data processing) comme le cœur de Exorde ainsi que Web app (Flask, Django) pour création de Admin Dashboard ou autre Web app possible. Ce qui nous permet de rester dans le même stack que la partie software. Ainsi que match bien avec shell scripting pour tout type de script en python ou bash.

Pour les inconvénients :

Ce n’est pas un langage frontend mais il y a beaucoup d’autre alternative a utiliser pour le frontend

Compare a d’autre langage comme Java ou C++, python est plus lent en exécution mais ça peut être améliorer par le multi threading et amélioration de code donc si le code est bien structuré et géré ça n’est pas un problème.

(programme I/O intense, multi threads, database)

:)

------------------------- Exercice 1 (plutot rapide) -------------------------

- Faire un script python qui compte le nombre d'élements unique sur fichier (sans la virgule ou autre séparateur)

https://raw.githubusercontent.com/exorde-labs/TestnetProtocol/main/targets/keywords.txt

- Trier par ordre alphabétique chaque élement

-- Combien d'élement ayant <= de 4 caractères?

-- Verifier si le nombre de keywords unique n'est pas nul

------------------------- Exercice 2 (rapide) -------------------------

Récuprer le contenu de : https://bafybeidqw6oyclcnmqjbj2472vzofyhafdtot5woenazsgpmm3b6irnaoi.ipfs.w3s.link/

C'est un fichier json qui contient une liste d'"items"

-- itérer et récuperer chaque champ "Content"

Voici l'exemple d'un item sans contenu:

*{*

*"CheckedIn": "",*

*"Content": "None",*

*"ContentMedias": {},*

*"ContentMentions": [],*

*"CreationDateTime": "2022-08-20 06:59:53+00:00",*

*"DomainName": "old.reddit.com",*

*"ErcAddress": "",*

*"FormattingDateTime": null,*

*"Language": null,*

*"SpotHash": "bafybeiei3iptgggr247r3ymlfgxytjwqfld3gljoee5lc35kp6yaljmxfi",*

*"SpottedIn": "RU",*

*"SpottingDateTime": "2022-08-20 15:05:22.905126",*

*"Url": "https://old.reddit.com/r/ethtrader/comments/wt0pcm/uniglo\_glo\_chainlink\_link\_and\_uniswap\_uni\_remain/"*

*},*

-- Combien d'items en tout?

-- Combien il y a t'il de liens http (ou https), au total, dans tous les "Content" uniquement?

-- Calculer le top 5 des mots (séparé par des espaces ou virgule) ayant >= de 8 et <= à 20 caractères.

-> si souci de RAM, me contacter bien sûr

------------------------- Exercice 3 -------------------------

0) installer les librairies (avec pip install) snscrape, datetime, pandas, pytz

--> le keyword est déjà renseigné, la ligne de requetage Twitter également

---> des parametres sont à préciser

a) recuperer 10 tweets exactement sur le keyword "bitcoin".

--> récuperer les champ "author, lang, id, hashtags et content".

----> construire un dataframe pandas avec les champs ci dessus, dans l'ordre que tu veux

b) afficher en bout de programme le nombre de hastags unique (dans l'ensemble des 10 tweets)

**FIN**

**merci <3**