

Parcours Python

13 avril

Python 101

11 mai

TD Webscraping

18 mai

TD : Cartographie

21 mai

TD : Exploration Twitter



Ave Caesar, pythonuri te salutant !

- Traduisez ce texte
- "Svyzx|.hyyp}h'sh'mpu3'plsz'z.đjypiylu{'lu'jovl|y'A')W\x80{ovu3'NP[T'🔥'()5'W|pz'plsz'shwpkiylu{'s.h|{|l|y'kl'jl'ql'kl'tv{'ovu{|l|x7f5"
- Pro tips:
 - pas de 7
 - builtins

Solution

- <https://pixees.fr/projet-cryptographie-seance-2-le-chiffrement-de-cesar/>

```
cypher =  
"Svyzx|.hyyp}h'sh'mpu3'plsz'z.đjypïylu{'lu'jovl|y'A')W\x80{ovu3'NP[T'🔥'()5'W|pz'plsz's  
hwpkïylu{'s.h|{l|y'k1'j1'q1|'k1'tv{'ovu{1|\x7f5 »
```

```
"".join([chr(ord(char)-7) for char in cypher])
```

ou

```
clear = []  
for char in cypher:  
    char_to_int = ord(char)  
    unshifted = char_to_int - 7  
    clean_char = chr(unshifted)  
    clear.append(clean_char)  
"".join(clear)
```

Lorsqu'arriva la fin, iels s'écrièrent en chœur :
"Python, GITM ♥ !". Puis iels lapidèrent l'auteur de ce
jeu de mot honteux.

À vos claviers

- Créez un générateur de mots de passe aléatoire :
 - prenant en paramètres :
 - un nombre total de caractères (obligatoire)
 - un nombre minimum de caractères spéciaux (facultatif)
- Créez une fonction calculant la « force » d'un mot de passe
 - <https://www.ssi.gouv.fr/administration/precautions-elementaires/calculer-la-force-dun-mot-de-passe/>
 - doit déterminer automatiquement la taille de l'alphabet utilisé
- Appelez depuis la ligne de commande le script contenant vos fonctions
 - paramètres :
 - nombre de mots de passe à générer
 - afficher la force ou non
 - affiche la liste des mots de passe et leur force le cas échéant
- Utilisez la bibliothèque string : <https://docs.python.org/fr/3/library/string.html>

Implémentation

- <https://gist.github.com/zulzeen/2f701c5aa39e18261089f3cc2584c2dd>

Matière

- nom
- competences
- duree_en_heures
- moyenne_a_obtenir

Majeure

- nom
- matieres

• Etudiant

- nom, prenom
- adresse_mail (généré automatiquement)
- majeure
- save()
 - sauvegarde dans un fichier les informations sur l'étudiants
- matieres_validees()
 - non implémenté

EtudiantGITM

triade
matieres_validees()
valider_compétence(competence)
competences_a_valider

EtudiantSCIA

groupe
matieres_validees()
noter_matiere(matiere, [notes])
moyenne

```
python = Matiere(« Python », [« programmer »,  
« indenter », « sauver le monde »], 12, 10)  
maths = Matiere(« Mathématiques », 36, 14)  
planete_solidaire = Matiere(« Planete Solidaire »,  
[« gérer des sherpas », « faire des cookies »,  
« sauver le monde », 800])  
gitm = Majeure(« GITM », [python,  
planete_solidaire])  
scia = Majeure(« SCIA », [python, maths])
```

```
marcel = Etudiant(« Marcel », « Cesson », gitm, « Dai Huen Jai »)  
marcel.valider_compétences([« sauver le monde », « programme »,  
« indenter »])  
marcel.competences_a_valider  
=> « gérer des sherpas », « faire des cookies »  
marcel.save()  
=> marcel, cesson, GITM, Python
```

Implémentation

- <https://gist.github.com/zulzeen/ad6f5a6a4738705adcfbefac3e792991>

Webscraping

- Objectif : extraire des informations d'un site web
- <https://www.gog.com/games?price=discounted&sort=popularity&page=1>
- https://www.humblebundle.com/store/search?sort=bestselling&filter=onsale&hmb_source=store_navbar
- Informations recherchées :
 - Nom du jeu
 - Lien vers la page du jeu
 - Prix normal
 - Montant de la réduction
 - Plateformes
- Format : CSV

Avant toute chose

- Redis : base de données NoSQL clé-valeur
- <https://redislabs.com/lp/python-redis/>
- <https://redis.io/commands>
 - GET
 - SET
- `docker run --name redis_cache -p 6379:6379 -d redis`
- `docker exec -ti redis_cache redis-cli`

Mettre en place un cache

- Créer une fonction `get_content` qui prend en paramètre une url
- Cette url est utilisée comme clé dans redis, la valeur associée à la clé est le html de la page correspondant à cette url
- si l'url est présente en tant que clé dans redis, retourner le html correspondant
- si l'url n'est pas présente, télécharger la page et la stocker dans redis (utilisant l'url comme clé), et retourner le html correspondant

Extraire les informations d'une page

- Pour une page donnée, transformer le html en soup pour pouvoir le parser
- Faire une fonction pour extraire tous les jeux d'une page
- Faire une fonction pour extraire les données d'un jeu
- Faire une fonction pour enregistrer les données dans un fichier CSV
 - Certaines données demanderont à être mises en forme ou calculées avant d'être stockées
- Faire une fonction pour récupérer une autre page
 - Permettre de définir un nombre maximal de page

Bibliothèques utiles

- redis - <https://github.com/andymccurdy/redis-py>
- requests - <https://2.python-requests.org/en/master/>
- selenium - <https://selenium-python.readthedocs.io/>
- bs4 - <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
- csv - <https://docs.python.org/3.9/library/csv.html>