TD8 : programmation Python et base de données

UCBL - Département Informatique de Lyon 1 – BDW - automne 2024

Objectif du TD : programmer en Python

Pour améliorer le site d'activités hebdomadaires, écrivons des fonctions Python et interagissons avec la base de données PostgreSQL.

Rappel du schéma relationnel :

ACTIVITÉS (idA, titre, type, #idU)
SEMAINES (année, numéro, ouverteAuVote)
UTILISATRICES (idU, nom, prénom, email)
VOTER (#idA, date, #idU, #année, #numéro, aAimé)

%m	Month as a zero-padded decimal number.	01, 02, ..., 12
%y	Year without century as a zero-padded decimal number.	00, 01, ..., 99
%Y	Year with century as a decimal number.	0001, 0002, ..., 2013, 2014, ..., 9998, 9999
%H	Hour (24-hour clock) as a zero-padded decimal number.	00, 01, ..., 23
%I	Hour (12-hour clock) as a zero-padded decimal number.	01, 02, ..., 12
%M	Minute as a zero-padded decimal number.	00, 01, ..., 59
%j	Day of the year as a zero-padded decimal number.	001, 002, ..., 366
%W	Week number of the year (Monday as the first day of the week) as a zero-padded decimal number. All days in a new year preceding the first Monday are considered to be in week 0.	00, 01, ..., 53

Extrait de documentation sur les motifs que l'on peut utiliser dans un format de *strftime*

Exercice 1 Révisions de la syntaxe Python

1. Le code suivant permet d'afficher des statistiques sur les votes, mais il contient des erreurs. Détectez-les et corrigez-les. Pourquoi est-ce une mauvaise pratique d'écrire une fonction de cette manière ?

```
1 fonction afficher_stats(nb_users, nb_total_votes) {  
2     avg = nb_total_votes - nb_users;  
3     print('<p>Total de ' + nb_total_votes + ' votes pour + nb_users + ' utilisateurs</p>')  
4     print("Nombre moyen de votes par utilisateur : {avg}</p>")  
5 }
```

2. Écrire une fonction Python *convert_nb_votes_milliers(nb_votes)* qui retourne le nombre de votes (e.g., 1234) sous forme d'une chaîne de caractères. Si le nombre de votes est supérieur à 1000, le résultat est divisé par 1000, arrondi à un chiffre après la virgule et suivi de la lettre k (e.g., 1.2k).
3. Écrire une fonction Python *sommer_nb_votes(tab)* qui retourne la somme des nombres de votes. La fonction prend un paramètre en entrée, une liste contenant les nombres à sommer. On vérifiera que chaque nombre est bien un nombre positif.
4. Écrire une fonction Python *get_week_year(d)* qui prend en paramètre une date au format ISO (AAAA-MM-JJ) et retourne le numéro de semaine et l'année (séparés par un tiret) pour cette date. La fonction `date.fromisoformat(une_date)` convertit la date ISO `une_date` en un objet python `date`. La fonction `objet_date.strftime(format)` convertit un objet `date` selon le format indiqué (voir l'[extrait des motifs de date](#) ci-dessus).

Exercice 2 Interrogation d'une BD avec Python

Les extraits de code ci-dessous ont pour objectif de se connecter à une base de données PostgreSQL.

```
config-bd.toml  
1 SERVER = "bd-pedago.univ-lyon1.fr"  
2 USER = "p1234567"  
3 PASSWORD = "Exemple2Mot2Passe!"  
4 DATABASE = "p1234567"
```

```
main.py  
1 from model import get_connexion  
2 config = load_toml('config-bd.toml')  
3 connexion = get_connexion(config['SERVER'], config['USER'],  
4     ↪ config['PASSWORD'], config['DATABASE'])
```

```
model.py  
1 import psycopg # package PostgreSQL  
2  
3 def get_connexion(host, username, password,  
4     ↪ db):  
5     try:  
6         connexion = psycopg.connect(host=host,  
7     ↪ user=username, password=password,  
8     ↪ dbname=db)  
9     except Exception as e:  
10        return None  
11    return connexion
```

1. Étudiez les extraits de code. Si l'on exécute `main.py`, quelle(s) instruction(s) faut-il ajouter dans le code pour se connecter à la base de données ?
2. Le code ci-contre permet de récupérer le nombre d'utilisateurs dans la BD, mais il contient des erreurs. Détectez-les et corrigez-les.
3. Écrivez une fonction qui permet de retourner les catégories des activités (attribut *type*), sans doublon et avec le nom des attributs du résultat (attribut `cursor.description`).

```
1 def count_users(connexion):
2     with connexion.cursor() as cursor:
3         try:
4             connexion.execute("SELECT COUNT(nom)
5                 ↪ FROM UTILISATRICES")
6             result = cursor.fetch()
7             return result
8         except psycopg.Error as e:
9             print(f"Error : e")
10            return NULL
```