

1 ANALYSE DU TD3

Cette analyse du code HTML et du CSS sera faite ensemble en début de séance. De nombreux éléments seront aux devoirs. **Prenez des notes et soyez attentifs !**

Les points que nous allons balayer sont :

- L'utilisation d'une icône dans l'onglet du navigateur à côté du titre ;
- L'ensemble des sélecteurs css et un certain nombre de propriétés.

2 FLASK - BASE

Q1. Reprenez ce qui a été vu en cours pour avoir un Hello World fonctionnel. Vous êtes invité à lancer votre app flask avec l'option debug (le temps du développement) :

```
flask --app hello run --debug
```

Q2. Nous avons déjà vu la fonction suivante en cours :

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        return "<h1>show_the_login_form()</h1>"
    else:
        return "<h1>do_the_login()</h1>"
```

Testez cette fonction.

Q3. On remplace maintenant la fonction précédente par les deux suivantes :

```
@app.get('/login')
def login_get():
    return "<h1>do_the_login()</h1>"

@app.post('/login')
def login_post():
    return "<h1>show_the_login_form()</h1 >"
```

Y a-t-il une différence entre les questions Q2 et Q3 ?

3 FLASK – BDD

Nous allons spécifiquement nous intéresser à la communication de notre application Flask avec une base de données (BDD). Dans un souci de simplicité, nous allons utiliser une base de données sqlite car c'est une BDD qui tient dans un fichier et qui pourra tourner sans difficulté sur votre machine.

Sur le Moodle, dans le *bdd.zip*, vous trouverez l'exécutable *sqlite3.exe* (pas besoin d'install, c'est ça qui est cool) et le fichier *chinook.sql*. Ce dernier fichier sert à créer une BDD déjà toute prête. Chinook, au même titre que NorthWind sont deux bases de données disponibles sur internet qui servent à faire des tests ou TP comme celui-ci sur des données fictives sans s'embêter à créer une base de données soit-même.

Instructions : Dézipper *bdd.zip* et copier *sqlite3.exe* dans votre dossier de travail. Mettez dans ce même dossier aussi l'autre fichier, *chinook.sql*. Ouvrez un terminal dans le dossier (le plus simple, dans votre navigateur de fichier est de taper ctrl+L puis *cmd*; le terminal s'ouvrira dans votre dossier de travail). Tapez *sqlite3 Chinook.db* pour créer une base de données vide qui s'appellera Chinook. Il ne reste plus qu'à remplir la base de données avec les instructions contenues dans le fichier sql :
.read chinook.sql.

Cela prend du temps, soyez patient !

Pour vérifier que l'opération est bien terminée, tapez la commande suivante (la commande est case-insensitive) :

```
SELECT * FROM Artist LIMIT 10;
```

Vous devez observer ceci :

```
1|AC/DC
2|Accept
3|Aerosmith
4|Alanis Morissette
5|Alice In Chains
6|Antônio Carlos Jobim
7|Apocalyptica
8|Audioslave
9|BackBeat
10|Billy Cobham
```

Vous avez la possibilité de voir le contenu de la BDD grâce à des outils gratuits trouvable sur le Microsoft Store.

Q4. Reprenez le code de la section 2 Flask - base et rajouter l'instruction suivante dans *hello.py* :

```
app.config.from_mapping(
    SECRET_KEY='dev',
    DATABASE=os.path.join(app.instance_path, 'Chinook.db'),
)
```

Il se peut que vous ayez besoin de faire un import. A vous de voir.

Il faut aussi que vous mettiez le fichier *Chinook.db* dans un dossier nommé « *instance* ».

L'objet de cette instruction est de créer des variables (SECRET_KEY et DATABASE) pour y accéder depuis d'autres endroits de votre code Flask.

Q5. Créez un nouveau fichier *db.py* et insérez-y le code issu de la documentation officielle de Flask (<https://flask.palletsprojects.com/en/2.2.x/tutorial/database/>) :

```

import sqlite3
from flask import current_app, g

def get_db():
    if 'db' not in g:
        g.db = sqlite3.connect(
            current_app.config['DATABASE'],
            detect_types=sqlite3.PARSE_DECLTYPES
        )
        g.db.row_factory = sqlite3.Row

    return g.db

def close_db(e=None):
    db = g.pop('db', None)

    if db is not None:
        db.close()

```

Ce code gère la connexion à la BDD et la fermeture propre de la connexion lorsque le programme se termine. Il n'y a rien de particulier à changer dedans (**ATTENTION**, en TP, vous utiliserez une BDD PostgreSQL et non SQLite, il y aura donc une modification à faire à ce moment-là dans ce fichier).

Vous allez maintenant récupérer dans le *bdd2.zip* les fichiers *base.html* et *list_artists.html*, que vous allez bien évidemment mettre dans le bon dossier.

Ajoutez le code suivant à votre fichier principal :

```

@app.get('/read_db')
def read_db():
    db = get_db()
    artists = db.execute("select * from artist limit 10").fetchall()
    return render_template('list_artists.html', artists=artists)

```

Q6. Observez et surtout COMMENTEZ tout ce qui se passe, et dans le DETAIL !!