

Un des moyens d'interagir avec une base de données est d'utiliser un serveur Web et d'utiliser une interface Web pour accéder à la base.

- ✂ Créer un répertoire `bd_python` pour enregistrer tous les fichiers de cette activité.
- ✂ Copier la base `base_livres.db` dans ce répertoire.

1. Créer un serveur Web

- ✂ Enregistrer le script ci-dessous dans un fichier `serveur.py`.

```
import http.server

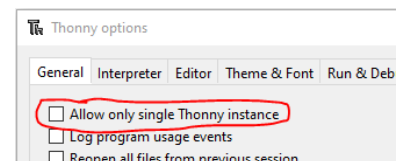
PORT = 80
server_address = ("", PORT)

server = http.server.HTTPServer
handler = http.server.CGIHTTPRequestHandler
handler.cgi_directories = ["/"]
print("Serveur actif sur le port :", PORT)

httpd = server(server_address, handler)
httpd.serve_forever()
```

- ✂ Lancer le serveur en exécutant le fichier `serveur.py`.

Tant que le programme n'est pas interrompu, le serveur est actif sur le port 8888. Mais ceci bloque le terminal dans lequel le serveur a été lancé. Si nécessaire, pour pouvoir lancer un autre terminal, il faut sur **Thonny** dans la fenêtre **Thonny options** (menu **Tools, Options**), décocher **Allow only single Thonny instance**.



2. Afficher les informations d'une base de données

- ✂ Enregistrer le script ci-dessous dans un fichier `affich_base.py`.

```
import sqlite3

def affich_base():
    print("""
    <!DOCTYPE html>
    <html lang = "fr">
    <head>
    <meta charset="utf-8"/>
    </head>
    <body>""")

    # connexion à la base
    connexion = sqlite3.connect('base_livres.db')
    curseur = connexion.cursor()
    # sélection de tous les attributs de la table illustateur
    curseur.execute("""SELECT * FROM Livres""")
```

```
print("<h1>Contenu de la table Livres</h1>") # titre
print("<table border = '1'>") # tableau
print("<tr><th colspan = '4'> Livres </th></tr>") # nom de la table
print("<tr><th>Identifiant</th><th>Titre</th><th>Auteur</th></tr>") # en-tête

for tuple in curseur: # affichage des tuples
    print("<tr>")
    liste = list(tuple)
    for champ in liste:
        print("<td>" + str(champ) + "</td>")
    print("</tr>")
print("</table>")
print("</body> </html>")

connexion.close()
```

affich_base()

☞ Afficher la page correspondante au script en tapant l'URL suivante dans la barre d'adresse d'un navigateur :

http://localhost:80/affich_base.py

- localhost désigne la machine courante (donc ici en local).
- 8888 désigne le port sur lequel le serveur est lancé (fichier `serveur.py`).
- `affich_base.py` désigne le nom du script dans le répertoire du serveur.

Contenu de la table Livres

Le résultat attendu est ci-contre :

Livres		
Identifiant	Titre	Auteur
1	Les Misérables	Hugo
2	La Peste	Camus
3	Les raisins de la colère	Steinbeck

Il est possible d'appliquer un style à la page Web obtenue.

☞ Enregistrer le script ci-dessous dans un fichier nommé `script.py`.

```
# ajout d'un style
def css():
    print("""
    <style>
        body {
            background-color: azure;
        }

        h1 {
            color: red;
        }
    </style>
    """)
```

☞ Compléter la fichier `affich_base.py` avec le script ci-dessous.

```
import style # à placer au début du programme
style.css() # à placer après affich_base()
```

☞ Vérifier le résultat obtenu.

3. Compléter une base de données

Un formulaire va être utilisé pour compléter la base de données.

✍ Enregistrer le script ci-dessous dans un fichier nommé `formulaire.py`.

```
import style

def affich_formulaire():
    print("""
    <!DOCTYPE html>
    <html lang = "fr">
        <head>
            <meta charset="utf-8"/>
            <title> Formulaire </title>
        </head>
        <body>
            <form action = "/compl_base.py" method = "post">
                <fieldset>
                    <legend> Livres </legend>
                    <p>
                        <label for = "titre"> Titre </label>
                        <input type = "text" id = "titre" name = "titre">
                        <label for = "auteur"> Auteur </label>
                        <input type = "text" id = "auteur" name = "author">
                    </p>
                    <p>
                        <button type="submit">Compléter la base</button>
                        <button type="reset">Effacer</button>
                    </p>
                </fieldset>
            </form>
            <a href = "http://localhost:8888/affich_base.py">Afficher la base</a>
        </body>
    </html>""")

    affich_formulaire()
    style.css()
```

La page obtenue contient un lien vers l'affichage du contenu de la base.

Lors de la l'appui sur le bouton de type `submit` (**Compléter la base**), les données du formulaire (`titre` et `author`) sont envoyées au fichier `compl_base.py` (voir balise `form`).

✍ Enregistrer le script ci-dessous dans un fichier nommé `compl_base.py`.

```
import cgi
from formulaire import *
import sqlite3

# récupération des données du formulaire
formulaire = cgi.FieldStorage()
titre = formulaire.getvalue("titre")
auteur = formulaire.getvalue("author")

# insertion des données du formulaire dans la base
```

```
connexion = sqlite3.connect('base_livres.db')
curseur = connexion.cursor()

donnees = (titre, auteur)
curseur.execute("""INSERT INTO Livres(titre, auteur) VALUES(?, ?)""", donnees)

connexion.commit()
curseur.close()
connexion.close()

formulaire() # lancement de la page formulaire
```

✍ Vérifier que maintenant, il est possible de consulter et de compléter la base.

L'interface **CGI** (Commun Gateway Interface) est un composant de la plupart des logiciels serveurs de pages web. Il s'agit d'une passerelle qui leur permet de communiquer avec d'autres logiciels tournant sur le même ordinateur. Ici, elle permet la récupération des données du formulaire.