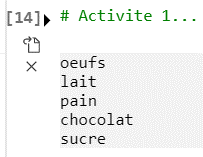
CORRECTION du TP « TODOLIST »

DR1 – Interagir en Python avec une base de données

**Activité 1**

*Résultat attendu*



import sqlite3

from flask import Flask

# Initialise l'application Flask

app = Flask(\_\_name\_\_)

donnee=('Courses',)

# A compléter

**curseur.execute("SELECT sujet FROM todo WHERE categorie = ?", donnee)**

**lstCourses = curseur.fetchall()**

**testreq.py**

**for course in lstCourses:**

**print(course[0])**

DR2 – Création d’un serveur Web avec le micro Framework Bottle

**Activité 2**



from flask import Flask

# Initialise l'application Flask

app = Flask(\_\_name\_\_)

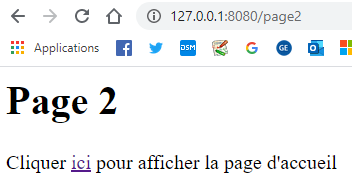
@app.route('/') # Définition d'une route et d'un gestionnaire

def accueil():

pages.py

    # 1. Construction de la page





    html = """<h1>Page d'accueil</h1>

    <p>Cliquer <a href='/page2'>ici</a> pour afficher la page 2</p>

    """

    # 2. Transmission de la page au navigateur

    return html # Envoi du html au navigateur

# A compléter

@app.route('/page2')

def page2():

**# 1. Construction de la page**

**html = """<h1>Page 2</h1>**

**<p>Cliquer <a href='/'>ici</a> pour afficher la page d'accueil</p>**

**"""**

**# 2. Transmission au navigateur**

**return html # Envoi du html au navigateur**

# run() démarre le micro serveur Web Werkzeug sur: http://127.0.0.1:port/

app.run(host='0.0.0.0', port=5000, debug=True)

DR3 – Affichage des données issues de la base « Todo » dans une page Web

**Activité 3** **(solution 1 : page web dans le code Python)**

import sqlite3

Todolist2.py

from flask import Flask

# Initialise l'application Flask

app = Flask(\_\_name\_\_)



# Définition d'une route et d'un gestionnaire pour la page d'accueil

@app.route('/')

def accueil():

    # 1. Connexion à la base

    connexion = sqlite3.connect('bdd/todo.db')

    # 2. Création d'un curseur pour exécuter les requêtes

    curseur = connexion.cursor()

    # 3. Exécution de requêtes

    curseur.execute("SELECT \* FROM todo ORDER BY categorie,sujet")

    # Le résultat renvoyé par la requête est une liste de tuples

    resultat = curseur.fetchall()

    # 4. Déconnexion

    curseur.close()

    # 5. Mise en forme pour l'affichage dans le navigateur

    html = '<h2>Liste des "choses à faire"</h2>'

**categorie = ""**

**for tuple in resultat:**

**if categorie != tuple[0]:  # Nouvelle catégorie => nouveau titre dans la page**

**categorie = tuple[0]**

**html += "<h3>" + categorie + "</h3>\n"**

**html += "<li>" + tuple[1] + "</li>\n"**

    # 6. Transmission de la page au navigateur

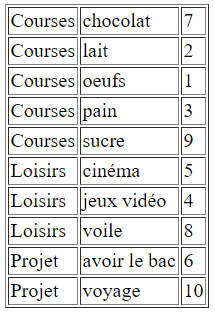
    return html

# run() démarre le micro serveur Web Werkzeug sur: http://127.0.0.1:port/

app.run(host='0.0.0.0', port=5000, debug=True)

pagev1.html

**Activité 4** **(solution 2 : template)**



<h2>2. Présentée sous la forme d'une table</h2>

<table border='1'>

        {% for ligne in table %}

        <tr>

            {% for colonne in ligne %}

            <td>{{ colonne }}</td>

            {% endfor %}

        </tr>

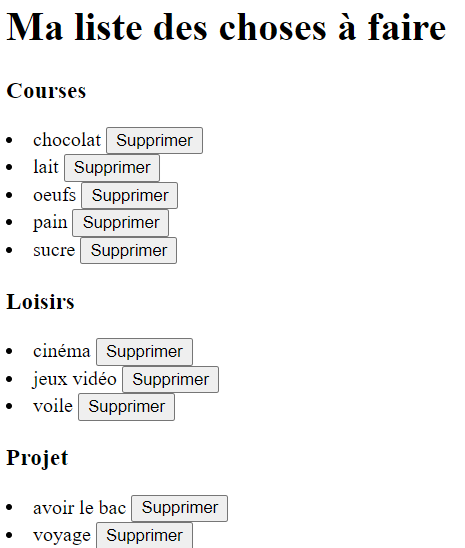
        {% endfor %}

    </table>

DR4 – Mise à jour de la base Todo avec des formulaires

**Activité 5**

todolist4.py



@app.route('/retrait', method='POST')

def retirer():  # Retrait d'un élément dans la table todo

    idelem = request.forms.get("supprimeItem").strip()

    # b1. Connexion à la base de données

**connexion = sqlite3.connect('bdd/todo.db')**

    # b2. Création d'un curseur pour exécuter les requêtes

**curseur = connexion.cursor()**

    # b3. Exécution de la requête

    sqlstr = f"DELETE FROM todo WHERE id={str(idelem)}"

    curseur.execute(sqlstr)

    # b4. Mise à jour de la table

**connexion.commit()**

    # b5. Deconnexion

    curseur.close()

    # b6. Retour à la page principale pour afficher la liste modifiée

    return redirect("/")



**Activité 6**

        <hr>

        <div id="rem2">Compléter le formulaire pour ajouter une nouvelle tâche.</div>

pagev2.html

        <form action="/ajout" method="POST">

            <select name="nouvelleCategorie">

                <option value="Courses" selected>Courses</option>

**<option value="Loisirs">Loisirs</option>**

**<option value="Projets">Projets</option>**

            </select>

            <input type="text" size="25" name="sujet">

            <input type="submit" name="save" value="Sauvegarder">

        </form>

**Activité 7**

@bottle.route('/ajout', method='POST')

def ajouter():  # Ajout de nouveaux éléments dans la table todo

todolist4.py

    # Acquisition des données transmises par le formulaire

    nouveauSujet =  request.forms.get("sujet")

    nouvelleCategorie = request.forms.get("nouvelleCategorie")

    if nouveauSujet != "":

        # 1. Connexion à la base de données

        connexion = sqlite3.connect('bdd/todo.db')

        # 2. Création d'un curseur pour exécuter les requêtes

        curseur = connexion.cursor()

        # 3. Exécution de la requête

        curseur.execute(

            "INSERT INTO todo (categorie,sujet) VALUES (?,?)", (nouvelleCategorie, nouveauSujet))

        # 4. Mise à jour de la table

        connexion.commit()

        # 5. Deconnexion

        curseur.close()

        # 6. Retour à la page principale pour afficher la liste modifiée

        return redirect("/")