

Développement Web

Serveur + Base de donnée + Cookies

Site du cours : <https://defelice.up8.site/devwebbdd.html>

Les exercices marqués d'un @ sont optionnels et à faire dans un second temps.

Exemple d'utilisation sqlite3 en python

```
import sqlite3 # importer la librairie
base = sqlite3.connect("monfichier.sqlite3")
interface = base.cursor()
interface.execute("""...ici requete sql... """)
interface.execute("INSERT INTO matable VALUES (?, ?)", ("Jean", "Sauveterre"))
cur.execute("...(:age,:nom)", {"age": "coucou", "nom": "Jean"})
interface.executemany("... (?)", [("nom1",), ("nom2",)])
interface.commit() # pour sauvegarder sur le disque les changements
base.close() # pour se deconnecter de la base
```

```
Set-Cookie: NOM=valeur; Expires=Wed, 21 Oct 2015 07:28:00 GMT;
Set-Cookie: NOM=valeur; Max-Age=7777;
openssl s_client -connect www.youtube.com:443 # sorte de necat ssl
En CGI : (les variables d'environnements)
    HTTP_COOKIE= // les cookies
    REQUEST_METHOD=GET // ou POST
        // dans le cas de POST les valeurs sont
        // envoyée sur l'entrée standard
        // dans le cas de get les valeur sont dans la variable
        // query_string
    QUERY_STRING=
```

Récupérer les variables d'environnement :

```
En python :
    import os
    os.environ // contient un dictionnaire des variables
    os.environ("CONTENT_TYPE")
En C :
    main(int n, char** arg, char** varEnv)
    // en C varEnv est un tableau d'adresse
    // où chaque adresse est un pointeur sur une chaine de caractere
    // Chaque chaine de caractère décrit une variable d'environnement
```

Exercice 1. *Message*

Même exercice que le dernier du TP précédent (Serveur Dynamique/Base de données simple) mais en utilisant une base de données sql au lieu d'un fichier. Le script doit s'appeler `/cgi-bin/messagesql` Rappel : Ecrire un programme exécuté par le serveur web CGI qui produit une page web avec un formulaire (ayant un champs de texte et un bouton submit) suivi d'une liste de messages. Soumettre le formulaire ajoute un message dans la liste des messages. Les messages sont à stocker dans une base de données (avec sqlite3 par exemple).

Exercice 2. *Nettoyer, Date*

Ajouter un bouton vider pour vider la liste des messages. Afficher la date d'ajout de chaque message et afficher les messages du plus récent (en haut) au plus ancien (en bas). Vous pouvez utiliser un champs hidden.

Exercice 3. *Mot de passe*

Modifier le serveur précédent avec une nouvelle page `/cgi-bin/message-passwd` et en ajoutant une table pseudo/mot de passe (que vous remplirez manuellement au départ) et un champs mot de passe. Ainsi un utilisateur ne peut ajouter un message que s'il donne indique son mot de passe.

Attention le protocole http (au contraire de https) ne chiffre pas les données de communications. Elles voyagent publiquement sur le réseau, notamment les mots de passe.

Exercice 4. *Inscription*

Ajouter une page `/inscription.html` pour pouvoir s'inscrire en choisant son pseudo et mot de passe. De plus faites en sorte de ne sauvegarder que la valeur hachée du mot de passe du coté du serveur.

Exercice 5. *Session*

Enrichissez votre serveur pour que l'utilisateur utilise un mot de passe de session et non à chaque message. Utilisez les cookies.

Exercice 6. *Modification*

Enrichissez votre serveur pour que l'utilisateur puisse modifier un champs s'il est (vous pouvez utiliser l'attribut `value` de la balise `input`).

Exercice 7. *Reponse*

Ajouter une page `/modification` pour pouvoir modifier son mot de passe. L'utilisateur voulant modifier son mot de passe doit donner son ancien mot de passe.