

Développement Web

2024-2025

Serveur Dynamique

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

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

Rappel

- La commande `$ python3 -m http.server --cgi 12345` lance dans le dossier courant un serveur web dynamique CGI qui écoute toutes les interfaces du système sur le port 12345.
 - Les scripts/programmes de ce serveur doivent être situés dans un sous-dossier `cgi-bin`.
 - Les scripts doivent être exécutables.
 - Les premières lignes produites (dans l'entrée standard) par le script (l'entête) doivent contenir `Content-type:text/html` suivis d'éventuelles autres lignes et finalement séparées du contenu de la page web par une ligne vide.
 - Le serveur CGI communique des paramètres (par exemple les valeurs des formulaires ou les cookies) au script à travers les variables d'environnement.
- `chmod u+x script.py` rend le fichier `script.py` exécutable par l'utilisateur.
 - Dans le cas d'un script interprété la première ligne doit indiquer le chemin de l'interpréteur. Pour un script python la première ligne est souvent `#!/usr/bin/python3`.
 - En python vous pouvez récupérer les valeurs du formulaire avec `import cgi`
`form=cgi.FieldStorage()`
`form['nomCle'].value` et la liste des champs remplis avec `form.keys()`.
- Exemple type d'un formulaire

```
<form action="http://....." method="POST">
  <input type="text" name="nom">
  <input type="checkbox" name="nom">
  <input type="submit" value="Envoyer">
</form>
```

Exercice 1. Simple

Concevoir et tester un serveur web avec un script `/cgi-bin/bonjour` (écrit dans le langage de votre choix) qui renvoie une page web affichant : Bonjour !

Exercice 2. Heure

Concevoir et tester un serveur web avec un script `/cgi-bin/date` qui renvoie une page web affichant la date du serveur. (vous pouvez utiliser le langage `bash` et la commande `date`).

Exercice 3. Get

Concevoir et tester un serveur web avec un script `/cgi-bin/get.py` (un script écrit en python par exemple) qui récupère la valeur `param1` d'une requête get et l'affiche. Exemple d'url :

`http://192.168.200.152:2022/cgi-bin/get.py?param1=test`

Exercice 4. Formulaire get

Concevoir un serveur web avec fichier, une page web : `formG.html` et un script `getTraitement.py`. La page web `formG.html` doit contenir un formulaire avec un champ demandant un texte. Après la soumission, qui doit se faire avec la méthode `GET`, le serveur doit renvoyer une page web affichant ce texte. Interdiction d'utiliser javascript.

Exercice 5. Formulaire post

La même chose que précédemment mais avec un formulaire `POST` le fichier doit s'appeler `formP.html`.

Exercice 6. Protocole Post

Donner dans un fichier `exemplePost.http` un exemple d'envoi d'une requete http *POST* envoyé par le navigateur (à la suite de la soumission d'un formulaire par exemple). (Aide : utiliser `netcat` et/ou voir le TP HTTP).

Exercice 7. *Base de données simple*

Concevoir et exécuter un serveur web avec une seule adresse : `/cgi-bin/bddsimple`. La page/script `bddsimple` doit contenir un formulaire texte que l'utilisateur peut soumettre. Le message soumis est enregistré et doit apparaître ensuite à la suite du formulaire. Les messages peuvent ainsi s'accumuler. Vous pouvez sauvegarder les messages dans un fichier texte.