



1ère NSI : Les formulaires avec Bottle

Bottle est un framework web pour Python. Il permet de créer des applications web (en python) à partir d'un simple fichier `bottle.py`.

Nous ne ferons qu'effleurer l'utilisation de Bottle, l'objectif étant simplement de revoir la notion de requête HTTP et de constater que les traitements côté serveur peuvent être exécutés par un code python au même titre qu'un code PHP.

Remarque: Si Bottle n'est pas installé sur votre PC, il suffit de charger le fichier `bottle.py` et de le placer dans le même dossier que les scripts qui seront écrits dans les pages qui suivent.

1 - Prise en main

Créez un fichier python (je l'ai nommé `main.py`):

```
import bottle

@bottle.route('/')
def index():
    return """
    <h1> Coucou </h1>
    <p> Première tentative Bottle. </p>
    """

bottle.run(host='localhost', port=8080, debug=True, reloader=True)
```

La fonction `run` présente en dernière ligne a pour rôle de lancer un serveur qui nous permettra d'exécuter des instructions qui ne sont normalement pas exécutables côté client. Tout se passera sur votre machine, mais il y a d'une part un client (votre navigateur) et d'autre part un serveur (lancé par Bottle).

lancez votre script, vous devriez obtenir une réponse dans le terminal telle que celle-ci:

Bottle v0.12.18 server starting up (using WSGIRefServer())... Listening on http://localhost:8080/

Ouvrez un navigateur à l'adresse indiquée (port 8080) `http://localhost:8080/`. Vous devriez voir une page avec un titre "Coucou" et un paragraphe "Première tentative Bottle."

Créons une seconde page. Complétez le code comme suit:

```
import bottle

@bottle.route('/')
def index():
    return """
    <h1> Coucou </h1>
    <p> Première tentative Bottle. </p>
    """

@bottle.route('/page2')
def page2():
```

```
return """
<h1> Autre page. </h1>
<p> Ceci est une seconde page. </p>
"""
```

```
bottle.run(host='localhost', port=8080, debug=True, reloader=True)
```

puis ouvrez le navigateur à l'adresse "http://localhost:8080/page2".

2 - Des paramètres

On va créer à nouveau des pages web via Bottle et l'on ajoutera à la main des paramètres (avec des valeurs) dans l'URL, comme nous l'avons déjà fait.

Exercice 1

On repart avec un code minimal pour créer une page web à l'aide de Bottle.

Écrire un fichier parametre_exo1.py qui contiendra:

```
import bottle

@bottle.route('/')
def index():
    return """
    <h1> Coucou </h1>
    <p> Première page. </p>
    """

bottle.run(host='localhost', port=8080, debug=True, reloader=True)
```

Ouvrez firefox à l'adresse indiquée par le terminal après avoir exécuté le script. A l'aide du raccourci F12 , ouvrir les "outils de développement" firefox. Sélectionner l'onglet "réseau". Rafraîchissez éventuellement la page du navigateur pour visualiser les requêtes HTTP.

La requête HTTP menant à cette page présente-t-elle des paramètres?



Exercice 2

On va maintenant ajouter des paramètres dans l'URL.

Entrer l'URL suivante dans la barre d'adresse firefox:

localhost:8080?nom=Duchmoll&prenom=Toto

Quelle modification est apportée à notre page?



Exercice 3

Nous pouvons donc passer des paramètres dans l'URL, comme nous le savions déjà. Mais peut-on demander un traitement de ces paramètres par Python ? La réponse est oui, bien sûr. Il nous faut pour cela écrire l'instruction qui va permettre au script de récupérer les valeurs des paramètres puis les traiter.

Récupérer les valeurs des paramètres. Pour savoir comment procéder, lire la documentation de Bottle Query Variables. Puis faites afficher les valeurs qui sont passées en paramètre. On pourra utiliser la fonction `format()` pour mettre en forme le texte.



Exercice 4

Modifier le script précédent pour qu'un entier écrit en base 10 en paramètre dans l'URL soit affiché en base 2 dans la page html.

3 - Formulaires

Plutôt que de passer les paramètres via l'URL comme dans les pages précédentes, on va maintenant les envoyer via un formulaire.

Exercice 1

En vous inspirant de l'exemple donné dans la documentation de Bottle, créez, via Bottle, deux pages.

- la première page `index.html` présentera un formulaire demandant le nom et le prénom du visiteur.
- La seconde page (affichée lorsque l'utilisateur cliquera sur le bouton de validation du formulaire) affichera un message (de votre choix) utilisant les nom et prénom qui auront été entrés par le visiteur.

Exercice 2

Modifier le script précédent pour que le formulaire demande deux entiers et que la page de traitement affiche le produit des deux entiers.