

Ce document est sous licence Creative Commons : [CC-BY-NC](#).

1 Présentation du problème

Nous allons nous intéresser à un exercice classique qui consiste à écrire un programme permettant de répondre "rapidement" à une question. Le plus souvent ce sera avec le protocole HTTP (donc sur une page web) mais cela peut se faire quelque que soit le protocole.

Nous allons voir le défi proposé par le site NEWBIECONTEST <https://www.newbiecontest.org/index.php?page=epreuve&no=134> dont l'énoncé est le suivant :

Description :

Ouvrez la page [prog1.php](#).

Elle vous attribue alors un nombre aléatoire que vous devez renvoyer à la page [verifpr1.php](#) en moins de 1 seconde.

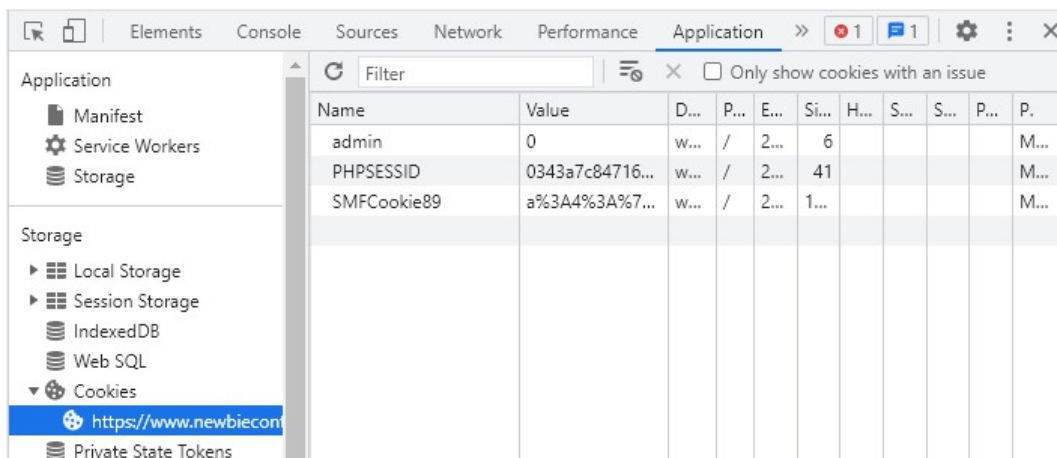
Vous devez renvoyer ce nombre sous la forme : `verifpr1.php?solution=nombre`. Il est nécessaire de coder un programme en local pour réussir cette épreuve.

Il faut donc "simplement" récupérer le nombre affiché et le renvoyer sur une autre page, en moins d'une seconde. Comme cela est multijoueurs, l'astuce ici est de s'assurer que l'on répond bien à son propre défi, et il faut donc gérer le **Cookie** qui est envoyé par le site.

2 Récupération du Cookie

Une fois sur la page du challenge, on ouvre la page du défi (<https://www.newbiecontest.org/epreuves/prog/prog1.php> qui affiche un nombre aléatoire.

On va regarder les Cookies présents (menu "Outils de développement", raccourci Ctrl+Maj+I) :



Name	Value	D...	P...	E...	Si...	H...	S...	S...	P...	P.
admin	0	w...	/	2...	6					M...
PHPSESSID	0343a7c84716...	w...	/	2...	41					M...
SMFCookie89	a%3A4%3A%7...	w...	/	2...	1...					M...

On en voit trois, et par déduction (ou essais-erreurs) celui qui nous intéresse est **SMFCookie89**. Il va donc falloir stocker la valeur de ce cookie, et la renvoyer dans la page de vérification.

3 Résolution en Python

Nous allons coder la solution en Python, par exemple dans l'environnement Jupyter (ou <https://colab.research.google.com/>).

Nous allons importer les deux bibliothèques que nous allons utiliser :

```
import requests
import re
```

La bibliothèque *requests* est utilisée pour effectuer des requêtes HTTP, tandis que *re* est utilisée pour les expressions régulières.

Le cookie nécessaire pour accéder au site Web est défini. Ce cookie est utilisé pour authentifier l'utilisateur lors des requêtes HTTP :

```
cookies = {'SMFCookie89': 'a%3A4...Copiez ici votre cookie de jeu...%7D'}
```

Une requête HTTP est effectuée sur l'URL du défi, et on envoie le cookie précédemment stocké. La réponse reçue (la page web) est stockée dans la variable *response* :

```
url = 'https://www.newbiecontest.org/epreuves/prog/prog1.php'
r = requests.post(url, cookies=cookies)
response = r.text
```

Une expression régulière est utilisée pour rechercher un nombre dans la réponse. Bien qu'on puisse faire la même chose plus facilement dans ce cas, c'est une bonne habitude à prendre pour des cas plus complexes.

L'expression régulière capture un nombre qui est ensuite stocké dans la variable *number*.

```
m = re.search('^.* est : (\d+)$', response)
number = m.group(1)
```

On effectue alors une nouvelle requête HTTP sur l'URL de réponse pour soumettre le nombre voulu. Les cookies sont de nouveau inclus dans la requête pour authentifier l'utilisateur. La réponse reçue après la soumission de la solution est imprimée et donne la réponse.

```
url = 'https://www.newbiecontest.org/epreuves/prog/verifpr1.php?solution='+str(number)
r = requests.post(url, cookies=cookies)
print(r.text)
```