

# TD 9

## Ressources Web : vers la lecture de flux RSS

### Chercher des trucs sur le Web : la bibliothèque *requests*

<https://requests.readthedocs.io/en/latest/user/quickstart/>

```
import requests
response = requests.get("http://inalco.fr/")

print(response.status_code)
print(response.headers)
print(response.content) # bytes
print(response.text)    # response.content as str
print(response.json())  # converts JSON to list/dictionary
```

### Exceptions en Python

<https://www.geeksforgeeks.org/python-exception-handling/>

```
try:
    # Some Code....

except:
    # optional block
    # Handling of exception (if required)

else:
    # execute if no exception

finally:
    # Some code .....(always executed)
```

Application à *requests* :

```
try:
    url = "http://inalco.fr/"
    response = requests.get(url)
except requests.ConnectionError:
    print("Erreur de connexion à "+url)
except requests.Timeout:
    print("Délai dépassé pour "+url)
except requests.RequestException:
    print("Erreur pour connexion à "+url)
```

## Les fichiers JSON

[https://fr.wikipedia.org/wiki/JavaScript\\_Object\\_Notation#Exemple](https://fr.wikipedia.org/wiki/JavaScript_Object_Notation#Exemple)

## Les fichiers YAML

<https://fr.wikipedia.org/wiki/YAML#Exemple>

```
load vs safe_load
dump vs safe_dump

with open('test.yaml', 'w') as file:
    yaml.dump(mydata, file)

with open('test.yaml', 'r') as file:
    mydata = yaml.safe_load(file)
```

## Les fichiers *pickle*

```
with open('test.pickle', 'wb') as file:
    pickle.dump(requests.get("http://www.inalco.fr/rss.xml"), file)

with open('test.pickle', 'rb') as file:
    myrequest = pickle.load(requests.get(file))
```

## Le hachage SHA256

```
from hashlib import sha256
bytes = "blabla".encode('utf-8') # Conversion string -> bytes
sha256().hexdigest()
```

## Âge d'un fichier ou d'un dossier

```
import os
st=os.stat('/tmp')
mtime=st.st_mtime # Date de la dernière modification en secondes
```

## Exercice 1

Écrire une fonction `http_get(url)` qui retourne juste le résultat de `requests.get(url)` (= un objet `Response`), **mais...**

- La fonction utilise un cache d'une durée de vie de 10 min. C'est à dire que la fonction `request.get(url)` n'est réellement appelée qu'une fois toutes les 10 minutes au maximum par URL ; sinon on renvoie le contenu du cache.
- Elle retourne aussi le contenu du cache en cas de problème de connectivité réseau (pas de réseau, site indisponible, etc.).

## Les flux RSS

[https://fr.wikipedia.org/wiki/RSS#Exemple\\_simple](https://fr.wikipedia.org/wiki/RSS#Exemple_simple)

## Exercice 2

Écrire un script qui lit un fichier de configuration en YAML, qui contient une liste de flux RSS et la durée du cache :

```
feeds:  
- http://www.inalco.fr/rss.xml  
- https://www.france24.com/fr/rss  
- https://radiofrance.fr/franceculture/rss  
- https://korben.info/feed  
- https://mastodon.social/@brownpau.rss  
cache_ttl: 600
```

... il affichera la date, le titre et le lien de l'item le plus récents des flux.

Pour la conversion de la date, voir : <https://stackoverflow.com/questions/1568856/how-do-i-convert-rfc822-to-a-python-datetime-object>