21 Python基礎

インターネットへのアクセス

- インターネットへのアクセスを行う場合、以下の点に注意
  - 同一のサイトに連続してアクセスしない（DoS攻撃とみなされる場合がある）
    - 連続アクセスする場合は、一定の間隔をあけてアクセスすること
  - robots.txtを参照して、取得しないページへのアクセスは避ける
    - robots.txtとは何か？

ライブラリ

urllib.requestモジュール

- 標準モジュールとして用意されている
  - 低レベルなモジュールなので、open,close などプロトコルの手続きを正しく記述する必要がある

Requestsモジュール

- 標準モジュールではないので、各自インストールする必要がある

```
conda install requests
```

  - Requestモジュールの使い方を調べてください。
    - どのようなメソッドが存在するのか？
    - 具体的にどのように利用するのか？

> https://requests.readthedocs.io/en/latest/

【requet1.py】

```
import requests

url = 'https://www.yahoo.co.jp/'
response = requests.get(url, timeout=5)

print(response)
print('-='*10)
print(response.status_code)
print('-='*10)
print(response.headers)
print('-='*10)
print(response.text)
print('-='*10)
print(response.content)
print('-='*10)
print(response.encoding)
print('-='*10)
print(response.cookies)
```

- どのような結果が得られるだろうか？
    - `.text` と `.content` の違いはなにか？

取得後の処理

- requestを使用して取得したサイトデータを処理するには、どうすればよいか？
- 「18_Python基礎」の練習1で扱ったのは、ファイルから処理した
    - 上記の例のような処理を行った場合、ファイルを取得するのがrequestの処理になる

練習

- テストには `https://sat.f5.si/~yoshimura/nt/` を使用すること！
- `https://news.yahoo.co.jp/` から直接データを取得し、18_Python基礎で作成したren1_5.pyを元に動作するようにしてください。（ren1_1.py）
    - URLはコマンドライン引数で渡すものとする
    - URLは1つのみ与えるものとする
    - エラー処理を組み込むこと
    - 出力は以下の通りとする

    - 多く含まれるURLをカウントし、件数の多い方から最大10件表示する（ren1_5.pyと同様）

HTML(ECMAScript/JavaScript) DOMとは

- DOMについて確認してください
- htmlデータを正規表現だけ、if文だけで処理するのは難しい場合が出てくる
    - そこで、構造を元に処理できると便利

Beautiful Soupモジュール

- 標準モジュールではないので、各自インストールする必要がある

```
conda install beautifulsoup4
```

```
$ conda install beautifulsoup4
Channels:
 - defaults
 - conda-forge
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /home/yoshimura/anaconda3/envs/py39

  added / updated specs:
    - beautifulsoup4


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    beautifulsoup4-4.12.3      |   py39h06a4308_0         217 KB
    ------------------------------------------------------------
                                           Total:         217 KB

The following NEW packages will be INSTALLED:
```

```
  beautifulsoup4     pkgs/main/linux-64::beautifulsoup4-4.12.3-py39h06a4308_0
  soupsieve          pkgs/main/linux-64::soupsieve-2.5-py39h06a4308_0


Proceed ([y]/n)?


Downloading and Extracting Packages:


Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

- BeautifulSoupの使い方を調べてください
    - 具体的な使い方を簡単にまとめてください。

> https://machine-learning-skill-up.com/knowledge/conda%E7%92%B0%E5%A2%83%E3%81%A7beautifulsoup%E3%82%92%E4%BD%BF%E3%81%86%E6%96%B9%E6%B3%95
>
> https://udemy.benesse.co.jp/development/python-work/web-scraping.html

練習

- 以下のプログラムを作成してください。(ren2_1.py)
    - RequestおよびBeautifulSoupを使用する
        - テストには `https://sat.f5.si/~yoshimura/nt/` を使用すること！
            - `https://computer.trident.ac.jp/` のトップページ内をスクレイピングする
        - リンク先を取得し、一覧を作成してください
        - 同一のURLは、畳み込んでunique(一意)にしてください

- 上記プログラムを修正してください（ren2_2.py)
    - リンク先が、同一サイト内の場合、FullURLになっていない場合がある
        - この場合、Base URL `https://computer.trident.ac.jp/` を付加して処理してください

        - 現在 tridentのSSL証明書の処理が古いため、正常に取得できない
        - コピーを用意したので、こちらでテスト
            - `https://sat.f5.si/~yoshimura/nt/trident.html`
    - `urllib.parse.urljoin` や `urllib.parse.urlparse` は調査の上使用しても良い
  - 同一ドメイン内へのリンクの場合、もう1階層全ページを取得し、一覧を作成してください（全部で2階層）
    - トップページ内のリンク
        - 上記リンク先のページ内のリンク
    - 外部へのリンクの場合は、取得しないでください
  - 出力を分かり易くしてください

- 以下のプログラムを作成してください。(ren2_3.py)
  - ホームページの変更を検知するプログラムを作成する
    - 先のURL `https://sat.f5.si/~yoshimura/nt/` は、適当なタイミングでページ内容が書き換わるようになっている
    - アクセス時にアクセス先の内容に関する記録を作成（ファイルに保存）する
    - 再度、アクセスしたときの内容と比較し、異なっていれば、その旨を出力する
    - 上記の内容を、一定の間隔で実行する
  - 複数のURLを登録できるようにしたい
  - 定期的な実行は、プログラム内で間隔を処理するもしくはタスクスケジューラ（Linuxの場合cron）を使用する方法でも良い