Crawler

A Web crawler or spider is an Internet bot which systematically browses web pages to get content. They are often used to perform Web indexing. `wget` is a very basic web crawler. If you have any question, use `man wget`.

1. Basic usage of `wget`

The following command downloads `URL` and stores the result in `res`.

```
wget -b -O res URL
```

`wget` can also be used to check if a resource exists and if it is available.

```
wget --spider URL
```

Instead of using `stderr` in the terminal, all the log of the command are stored in `wget.log`:

```
wget -o wget.log URL
```

`wget` also supports user agent to fake clients to servers.

```
wget -user-agent="iOS␣4_3␣-␣iPhone␣-␣Safari␣533.17.9"  URL
```

With the previous command, we act as an iPhone running iOS 4.3 and Safari 533.17.9. It is useful when we know that a server provides different contents depending on the client. Most browsers have nowadays a plugin, addon or extension to fake user agent.

It is very likely that some servers will blacklist you if you start to use `wget` extensively to prevent this it is possible to add extra delay between `wget` requests:

```
wget -w 5 --random-wait URL
```

The previous command wait 5 seconds between the requests plus an extra random delay. **Do not forget to use it !**

`wget` can also easily recover all files of a certain types on a URL:

```
wget -r -A.pdf URL
```

This command recovers all the pdf files referenced in the URL.

If you want to recover everything except a certain type of files, it is also possible. If you want everything except the `jpg` file, use the following command.

```
wget -r --reject=jpg URL
```

Question 1: Use `wget` to recover all the `pdf` files and all the `gz` files in the teaching section of my webpage. Create a `python` script to recover from each `pdf` files, the text it contains and then process the text to create a wordlist. To process `pdf` files, we have three choices `slate`, `pdfminer` and `PyPDF2`. Which package is the most satisfying ?

To install `slate`:

```
sudo pip install --upgrade --ignore-installed slate==0.3 pdfminer==20110515
```

To install `pdfminer` and how to use it:

`http://survivalengineer.blogspot.fr/2014/04/parsing-pdfs-in-python.html`

```
sudo pip install --upgrade pdfminer
```

To install `PyPDF2`:

```
sudo pip install PyPDF2
```

Finally to extract the word from the text, use regular expression (`re`) library. (hint) What is done by the following piece of code ?

```
import re
TEXT= 'Mr Lauradoux is the greatest of all the teacher !'
wordlist=[w for w in re.split('\W+', TEXT) if w]
print wordlist
```

Question 2: Same question but for `html` documents, you are free to use the package you want.

2. Advanced usage of `wget`

By default, `wget` is a nice bot: it obeys a site's `robots.txt` file and `no-follow attributes`.
If `wget -debug` output says something like

- `Not following toto.html because robots.txt forbids it`
- or `no-follow in index.html`

then it has followed the rules of the `robots.txt` file of the website. To play without respecting `robots.txt` files:

```
wget -erobots=off URL
```

Question 4: Explore the `robots.txt` of the university and of `http://www.robotstxt.org/`. Use the `user-agent` to test the different `robots.txt` files.

It is possible to make a full mirror of a website using `wget`:

```
wget --mirror -p --convert-links -P ./LOCAL-DIR URL
```

Be careful when you mirror a website, it can consume lot of resources for both the server and you !
If you do not want to get everything but only

```
wget -r -l3 -spider -D inria.fr http://planete.inrialpes.fr/~lauradou/
```

Let try to understand the option of the previous command:

- `-r` for recursive (follow the link)
- `-l3` indicates the number of levels to recurse (here 3)
- `-D inria.fr` list of domains (separated by comma) for which we allow crawling.

Question 4: From the script you have already written, create a wordlist of all the `html` and `pdf` files available on a webpage.

Question 5: the Custom Word List generator `cewl` is often mentioned to build good wordlist for `john`. Use it on my webpage and compare the result with your own tool. Are you beating `cewl` ? How to improve your script.

```
sudo apt-get install cewl
```

3. SCRAPY

`wget` is a great tool and most of the time it is better to use it than to try to build your own tool. Still, building your own crawler can be interesting. We are going to use `scrapy` so first install it. Follow the instructions and manage all the dependencies (`https://doc.scrapy.org/en/latest/intro/install.html#intro-install`).

```
sudo pip install scrapy
```

Then, follow the tutorial `https://doc.scrapy.org/en/latest/intro/tutorial.html` to build your first crawler.