Lab 6 - Internet Crawling

Introduction

In ex6 you will build a simple search engine, which will search different Internet pages. The search engine will be based on *crawling* - a process in which we automatically go through different webpages, and continue to go between the pages which refer one to the other.

Note - most Internet sources block users who attempt to crawl and download too many of their webpages, and thus in ex6 we limit you to search for 12 pages only.

The Problem

In this lab, we will practice how to download a single Internet page, and print all the addresses of the pages to which it refers to.

At the end of the lab you will submit a single python file **`crawler.py`** which will contain the code for the lab assignment. The crawler will receive from the command-line a single webpage from Wikipedia.

Stage 0 – Installing BeautifulSoup4 and requests

First, we will install the packages we need to accomplish our mission. We need the **requests** package to download the html code of our internet page, and we will use the BeautifulSoup4 package to find the HTML tags we are interested in. To install these packages, run the following command from the terminal:

```
pip install requests bs4
```

Stage 1 - Downloading the webpage

First, we will download the page by using the **`requests`** module. For the string url which is a string of an **html** page, we can download the page in the following way:

```
import requests
response = requests.get(url)
html = response.text
```

Now, the **`html`** variable will include all the data from the given webpage: including the html code of the webpage. Next, we will search the webpage for references to other webpages (actually, these are links that you can visit directly, by surfing this page and clicking on them!).

Stage 2 - Searching links

This part will be done with the **`BeutifulSoup4`** module. This module gets a string representing the html code of the webpage (we got this at stage 1) and searches for specific types of data.

In our case, we will be interested in links which are represented as **`<a>`** tags, inside paragraphs (which themselves are represented as **`<p>`** tags).

We can start by importing the module, and generating a *BeautifulSoup* object, which receives the html string we got at the previous step.

```
import bs4

soup = bs4.BeautifulSoup(html)
```

Next, we will iterate over all the paragraphs in the original webpage, meaning all the data between **`<p>`** parenthesis.

```
paragraphs = soup.find_all("p")
```

This creates a paragraphs object, which will allow us to iterate over, and extract all the links to the output pages, effectively searching for all the data in between **`<a>`** parenthesis.

```
for p in paragraphs:

     links = p.find_all("a")
```

Assignment

Write a program **`crawler.py`** which receives as a command-line argument a single html link from Wikipedia. The program will download the webpage data, and print all the links which the webpage refers to.

Notice that:

- The webpage can refer to itself
- The webpage can refer to another page multiple times (each time it should be printed separately)

- The printouts to the referred webpages must have an absolute domain address, such as ***www.wikipdia.org/local_address_of_referred_page***. You main need to manually add this.

Good Luck!