파이썬 스터디 활동보고서 DAY 8

기본 활동 정보

날짜 : 2018년 7월 25일 (수)
시간 : 6시 30분 - 8시
장소 : 국민대학교 북악관 카페테리아

참여 인원 : 3명
우현웅 20153195 소프트웨어 전공
최승호 20142772 소프트웨어 전공
김선필 20143038 소프트웨어 전공

1. 스터디 과제 검사

강의 : Edwith 파이썬을 이용한 웹 스크래핑 챕터12 파트 1 ~ 6 듣고 실습하기
과제 : 없음

2. 강의 리뷰

새롭게 알게된 내용들 정리

파트 1 : 소켓 모듈을 통한 네트워크 연결

TCP : 소프트웨어 사이를 연결하는 Pipe 역할
Socket : 응용프로그램이 TCP 를 이용하는 창구 역할
> 응용 프로그램과 소켓 사이의 인터페이스를 소켓 인터페이스라 한다.

포트 넘버 : 기능에 따라 구분 서비스 번호라고 생각하면 될 듯

import socket
> 소켓 사용 패키지

my_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
> 소켓 객체의 생성

my_socket((‘data.pr4e.org’, 80))
> 소켓 객체를 통한 연결 튜플로 호스트와 포트번호를 추가해서 넣어야 한다!

파트 2. HTTP 를 이용해 서버에 요청 보내기

주로 사용하는 프로토콜은 http이다. 프로토콜은 규칙의 모음이다. telnet이나 dns도 프로토콜이다.

링크 클릭하면 서버에 요청 전송하고 서버로부터 파싱/렌더링을 통해 html문서를 받는다.

http 요청
ex) GET http://www.dr-chuck.com/page1.htm HTTP/1.0

팁 : command창에서 ctrl + shift + c, v 를 통하여 복사 붙여넣기 가능

파트 3-1. 파이썬을 이용해 웹 데이터 읽어오기

```
import socket
mysock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
mysock.connect((‘data.pr4e.org’, 80))
```
> 소켓 생성 과정
> SOCK_STREAM이 socket을 스트림으로 받아온다

```
cmd = ‘GET http://data.pr4e.org/romeo.txt HTTP/1.0\n\n’.encode()
mysock.send(cmd)
```
> 웹 데이터 읽기 명령문 저장 및 실행
\n http 요청을 하기 위해서는 개행문자가 필요하다
\r은 커서를 맨앞으로 가게해준다

```
while True:
  data = mysock.recv(512)
  if (len(data) < 1 ) :
    break
  print(data.decode())
```
> 소켓을 통해 512문자를 전달받아 data에 저장.
> 만약 data 가 비었다면, 종료. 비지 않았다면 출력 후 다시 전달
> 1로 바꾸면 1byte씩 받아와 한글자씩 출력하고 만약에 큰 숫자가 오면 \n기준으로 한줄씩 출력한다.
> 헤더와 바디 따로 받아오고 한줄로 구분한다. 일종의 규칙이다.

```
mysock.close
```
> 소켓 연결 종료

파트 4 : 문자의 표현과 인코딩 / 디코딩

ASCII : 아스키코드
> 128개의 라틴어 문자 집합
> 하나의 문자당 8비트 = 1바이트를 차지함
> 각각의 문자가 숫자와 대응된다
> 라틴어만 표현 가능하므로, 다른 문자열의 표현이 가능한 방법의 개발이 필요해졌다

UNICODE : 유니코드
> 아스키코드의 문제점을 개선. 수많은 문자의 표현이 가능.
> UTF 라고 불리며, 이는 Unicode Transformation Format의 약자이다.
> UTF - 32 는 한 글자당 4바이트를 할당하며, 문제점은 전송 시 너무 많은 자원을 필요로 한다는 것이다.
> 이를 압축한 것이 UTF-16 혹은 UTF-8이며, UTF-8이 인터넷에서 사용하기 가장 적합하다.
> UTF-8은 동적으로 1~4 바이트를 할당한다.
> 이로 인해 인터넷 자료의 대부분 인코딩은 UTF-8 을 사용하게 된다.

파이썬의 문자열
> 파이썬 2에서는 문자를 ASCII 코드로 나타낸다.
> 파이썬 3에서는 문자열을 유니코드로 나타낸다.
> 그래서 파이썬 3에서는 모든 문자열이 str로 뜬다. 유니코드기 때문에

소켓의 인코딩 / 디코딩
> 인코딩은 unicode -> utf-8 , 디코딩은 utf-8 > unicode
> 데이터를 외부로 보내거나 받을 때, 인코딩 / 디코딩을 하는 것은 소켓의 역할이다.

한글 사용 에러 해결법
> 아톰에서 한글 사용시 에러
-*- coding: utf-8 -*-
> 주석을 이용해 기본적으로 utf-8 을 사용함을 명시

파트 5 : urllib 를 이용한 웹 데이터 읽기

```
import urllib.request, urllib.parse, urllib.error

fhand = urllib.request.urlopen('http://http://data.pr4e.org/romeo.txt')
for line in fhand:
  print(line.decode().strip())
```

> terminal 에서 python 3 로 접속하여 코드를 작성하면 정상적으로 코드가 작동함!!
> 소켓이 하는 일을 대신해준디 ㅏ
> urlopen이 파일의 헤더를 자동으로 해석과 처리를 해준다.
> urlopen 파라미터에서 url을 자동으로 인코딩해서 보내주고 객체로 리턴해준다

실습 내용 : 단어를 찾아, 단어별 개수를 세는 것

```
for line in fhand :
  words = line.decode().split()
```
> 인코딩은 자동으로 해주지만 디코딩은 안된다. 그래서 decode()를 해줘야 한다.

> 한 줄을 읽어와 해당 줄을 띄어쓰기 단위로 나눈 리스트를 words 에 저장한다.
```
  for word in words:
    counts[word] = counts.get(word, 0) + 1
```
> words 의 값들을 word 로 하나씩 뽑아와, counts 딕셔너리에 개수를 센다.
> get 은 딕셔너리에 대한 메소드로, 해당 word 에 대한 value 인 개수를 가져온다.

> word, 0 으로 작성한 이유는, 만약 value 가 없을 경우 default 값을 0 으로 지정한다는 의미를 가진다.
> 만약 value가 존재한다면, 해당 밸류에 대해 +1 한 값을 value 로 다시 저장하는 것이다.

최종적으로 counts 리스트에는 각각의 단어와 해당 단어의 빈도가 key - value 로 저장된다.

파트 6 : BeautifulSoup 를 이용한 웹 데이터 스크래핑

웹스크래핑 = 크롤링 = 스파이더링 - 프로그램이 페이지를 살펴보고 정보를 추출하고 조사하는 것을

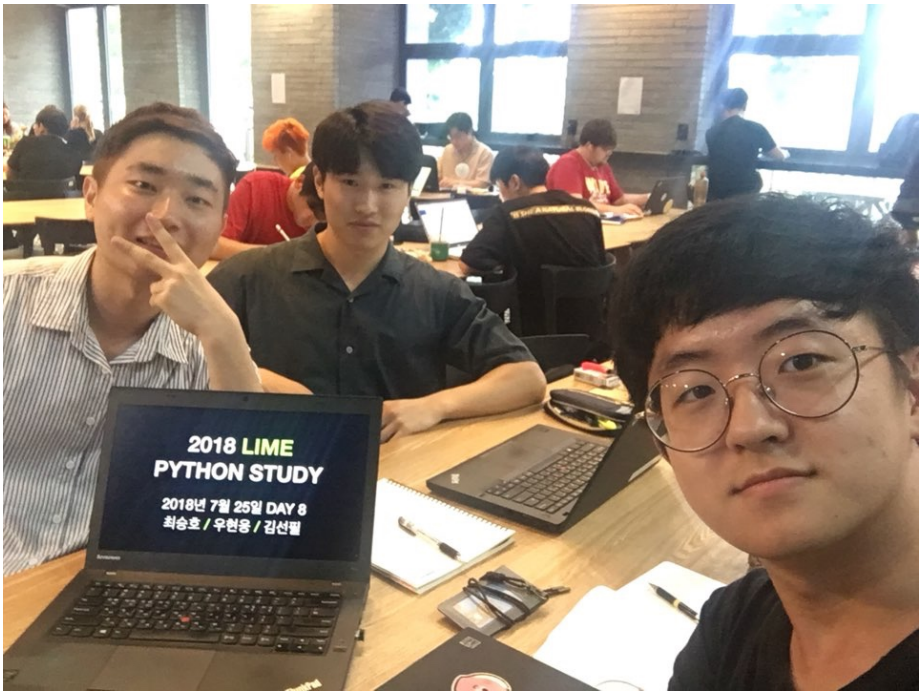
스크래핑을 허용하지 않는 사이트가 존재할 수 있다! 주의할 것

BeautifulSoup : 웹페이지의 다양한 문제를 해결해주는 패키지
BS 는 하나의 객체를 가지며, 해당 객체에 대한 메소드가 존재한다.
이 메소드를 이용하여 특정 태그에 손쉽게 접근할 수 있다!

3. 다음시간 강의 / 과제

다음시간 : 7월 27일 (금)
그 이후의 스터디 : 7월 31일 / 8월 2일 / 8월 8일 / 8월 10일 ...

1) edwith 파이썬 웹 크롤링 챕터 13 파트 1 ~ 4 JSON 사용하기까지 듣고 실습하기
2) Beautiful Soup 기능 사용법 / 사용 예시 자율적으로 알아오기



< 2018년 7월 25일 스터디 DAY8 >