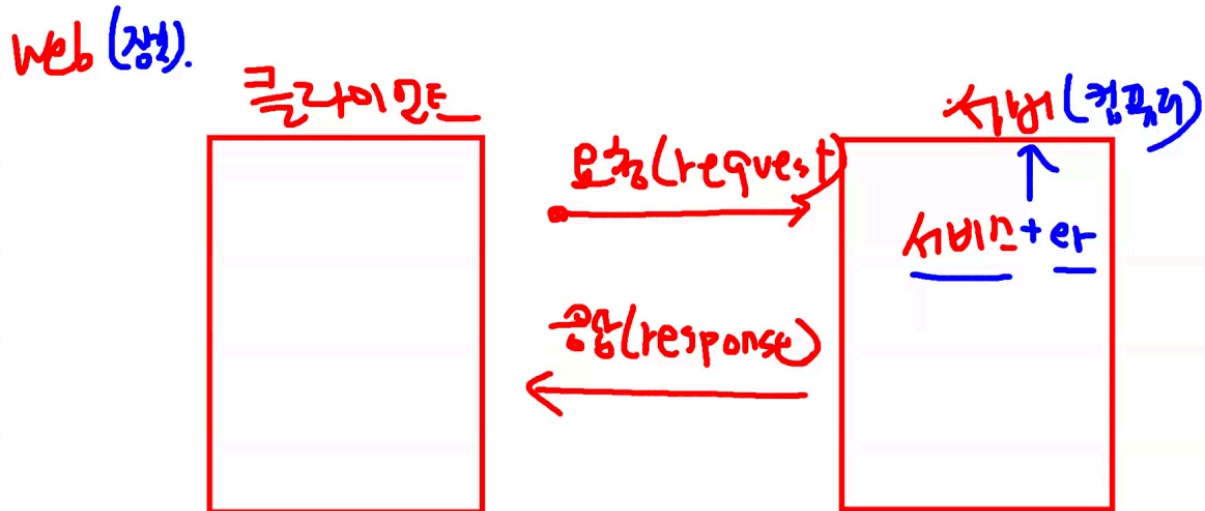


Python2 Day-02

웹 크롤링(스크래핑)

서버와 클라이언트

- 서버는 컴퓨터이다.
- 클라이언트들의 요청(request)에 맞는 응답(resoponse) 처리를 해 준다.



웹(web)

- 요청과 응답이 이루어지는 장소 (서버와 클라이언트의 공간)
- 수억명 수만명들이 요청과 응답을 받기 때문에 마구잡이로 거미줄 모양으로 되어있기 때문에 웹 이라고 부른다.

웹 브라우저

- chrome, edge 등 인터넷에서 웹 서버의 모든 정보를 볼 수 있도록 하고, 문서 검색을 도와주는 응용 프로그램이다

URI(Uniform Resource Identifier)

- 프로토콜://도메인:포트번호/경로
- ex)
<https://sports.news.naver.com/basketball/index>
프로토콜(http)://도메인:포트번호(sports.news.naver.com)/경로(basketball/index)
- 프로토콜부터 포트번호까지를 URL (Uniform Resource Locator)이라고 부르며, 경로만 특정할 때 URI라고 한다.

도메인

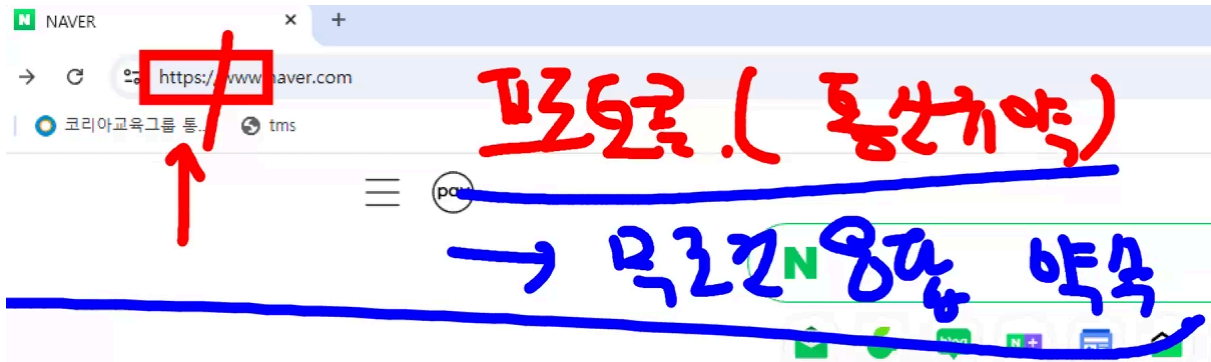
ex) naver.com
IP 대신에 사용하는 웹 상의 별칭

IP

PC의 고유한 주소값 ex) 198.234.321.32

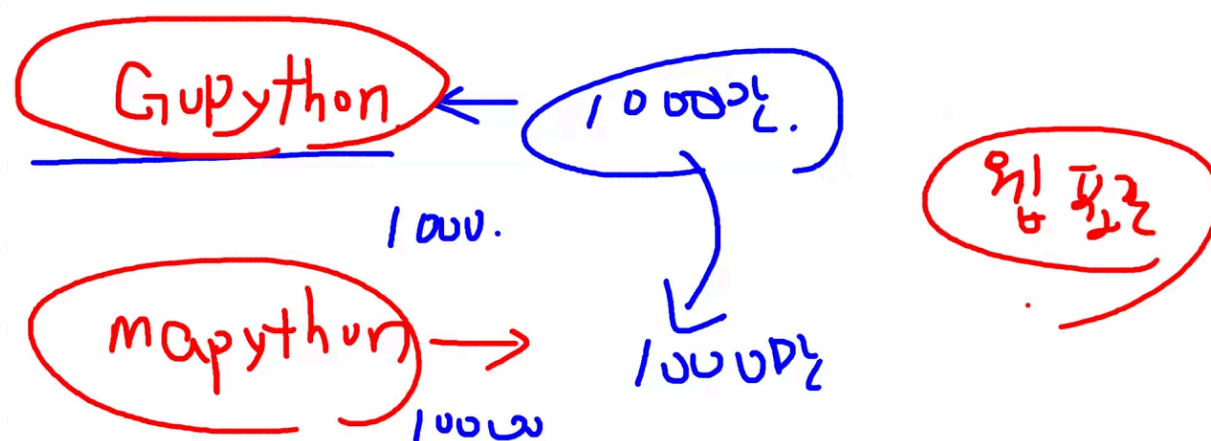
프로토콜

- 프로토콜이란 통신 규약이며, 무조건 응답 해주겠다는 약속이다.
- https라는 프로토콜이 붙어 있으므로 네이버는 어떤 값이든 응답을 해준다.
- https는 ID, PW 등 data를 암호화 시켜주는 프로토콜이다 (해커로부터 안전함).



WWW(World Wide Web), W3C

- www이란 웹표준을 지정을 한다.
- 웹 표준이라는것은 웹 안에서 약속을 지키자는 뜻이다.
- 웹 표준을 지정함으로써 웹 안에서는 모두가 공평하게 지정한 언어를 사용할 수 있다. (벤처사의 독식을 막을 수 있다)
 - ex) 구글, 마소 등에서 언어를 만들어 기업끼리 독식을 할 수 있다.
- W3C는 웹 표준을 정의하는 기관이다.



HTML(Hyper Text MarkUp Language)

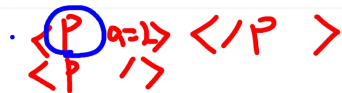
- 웹 페이지를 만드는 문법을 갖춘 언어, 태그(Tag)로 이루어져 있다.
- Markup: 웹 표준 (어디서든 똑같이 번역됨)
- Markdown: 웹 표준이 아님 (똑같이 번역되지 않음)

태그(Tag)

- 어떤 의미를 지니고 있다는 뜻
- 1) 여는 태그(Opening tag) : 요소의 이름과 열고 닫는 꺾쇠 괄호로 구성됨 <html>
- 2) 닫는 태그(Closing tag) : 요소의 이름 앞에 슬래시(/)를 써서 구성 </html>
- 3) 내용 (Content) 요소의 내용이며, 단순한 텍스트 이다.
- 4) 요소(Element) : 여는 태그와 닫는 태그, 내용을 통틀어서 이르는 말

ex)

- 밑의 사진에서 태그의 이름은 P 이다.
- 여는 태그는 <P>, 닫는 태그는 </P>
- 이 전체를 통틀어서 element 라고 부른다

· 

태그의 속성

- HTML은 구조화된 언어이다. 쉽게 말하자면 각 태그는 고유한 기능을 가지고 있다. 하지만 추가적으로 기능이 필요할 때마다 태그를 만든다면 태그들의 종류가 다양해질 것이다. 그래서 재사용과 사용 목적에 따라 다르게 사용할 수 있도록 속성이 존재한다.

<태그명 key=value ></태그명>

key, value 한 쌍으로 이루어져 있으며, value에 따라 적용되는 값이 다르다.

ex)

```
▼<div class="image">
  ::before
  
  ::after == $0
</div>
```

· 

- id : 중복되지 않은 태그의 특정한 속성 (고유 속성으로 하나만 존재한다)
- class : 태그들을 그룹화 하는 공통적인 속성 (공통 요소들을 묶기 위해서 사용한다)

ex) 여기서는 공통된 class = 'link_news_end' 를 사용한다.

```
<a href="https://m.sports.naver.com/wbaseball/article/382/0001184656" onclick="clickcr(this, 'pop.article', '', '', event);" class="link_news_end">아깝다 김혜성' 베츠, 도쿄시리즈 불기, '美' 소가 귀국</a> == $0
```

```
<a href="https://m.sports.naver.com/wbaseball/article/023/0003893900" onclick="clickcr(this, 'pop.article', '', '', event);" class="link_news_end">177kg 참치 해체쇼 선사한 오타니, 다저스 동료에 호화 일식 썼다</a> == $0
```

기본적인 HTML의 구조

```
<html>
  <head></head>
  <body></body>
</html>
```

크롤링이란?

- 여러 웹 페이지를 기계적으로 탐색하는 것

스크래핑이란?

- 특정한 하나의 웹 페이지를 탐색하고, 소스 코드 작성자가 원하는 정보를 얻어내는 작업

주의사항

- 크롤링을 무분별하게 사용하면 과부하를 일으킬 수 있다.
- 상업적인 용도나 불법적인 용도로 사용 시 법적 문제가 발생할 수 있다.

웹 크롤링 패키지

- requests : 파이썬에서 동작하는 작은 브라우저



```
1 import requests
2 url = 'https://www.naver.com'
3
4 # .get(): 페이지 요청
5 requests.get(url)
```



<Response [200]>

200: 서버 응답이 잘 넘어갔을때

400: 클라이언트쪽에서 응답이 안넘어갔을때

500: 서버쪽에서 문제가 생겼을때

requests의 메서드

- 1) .encoding() : 인코딩 설정
- 2) .status_code : 상태 코드
- 3) .text : 웹 페이지 소스
- 4) .context : 웹 페이지 소스(모든 문자)

BeautifulSoup

- html 태그들을 보기 쉬운 형태로 처리해주는 라이브러리
- 1) find('tagname') : 태그 중에서 태그명이 'tagname' 인 첫 번째 것
- 2) find('tagname').text : 위에 묶음 중 내용만 가져오기
- 3) find('tagname', class_='클래스속성명') : 'tagname' 인 것 중 클래스 속성명인 것의 첫 번째 것
- 4) find('tagname', id='아이디속성명') : 'tagname' 인 것중 아이디 속성명인 것
- 5) find_all('tagname') : 태그들 중 태그명이 'tagname' 인 모든 것을 리스트로 가져옴

User-Agent

- 사용자의 소프트웨어 식별 정보

user-Agent의 필요성

- 무분별한 크롤링으로 서버의 과부하를 막기 위해 프로그램을 통해서 접속하는 것을 차단하는 사이트 들이 있다. 그런 경우 원하는 정보를 추출할 수 없기 때문에 header에 user-agent 정보를 심어야한다.

내 User-Agent를 확인하기

- 1) 브라우저
 - a) <https://www.useragentstring.com/>
- 2) 운영체제
 - a) https://www.whatismybrowser.com/detect/what-is-my-user-agent/#google_vignette

코드 예시:

https://colab.research.google.com/drive/1vA-UpFB_mw9s27QXG2PQkyz1ajkmNuLQ#scrollTo=DnDzek2WEKGP