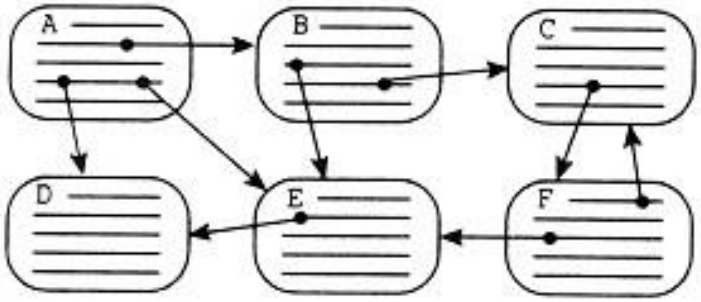
Web Crawling in Python
참고자료

https://github.com/nhkim55/bigdata_fintech_python

웹(Web)

웹

- World Wide Web
- 인터넷 상에서 동작하는 하나의 서비스
- 인터넷 상의 정보를 하이퍼텍스트 방식과 멀티미디어 환경에서 검색할 수 있게 해주는 정보검색 시스템
- 하이퍼텍스트란 링크를 통해서 다른 문서들끼리 연결할 수 있는 텍스트

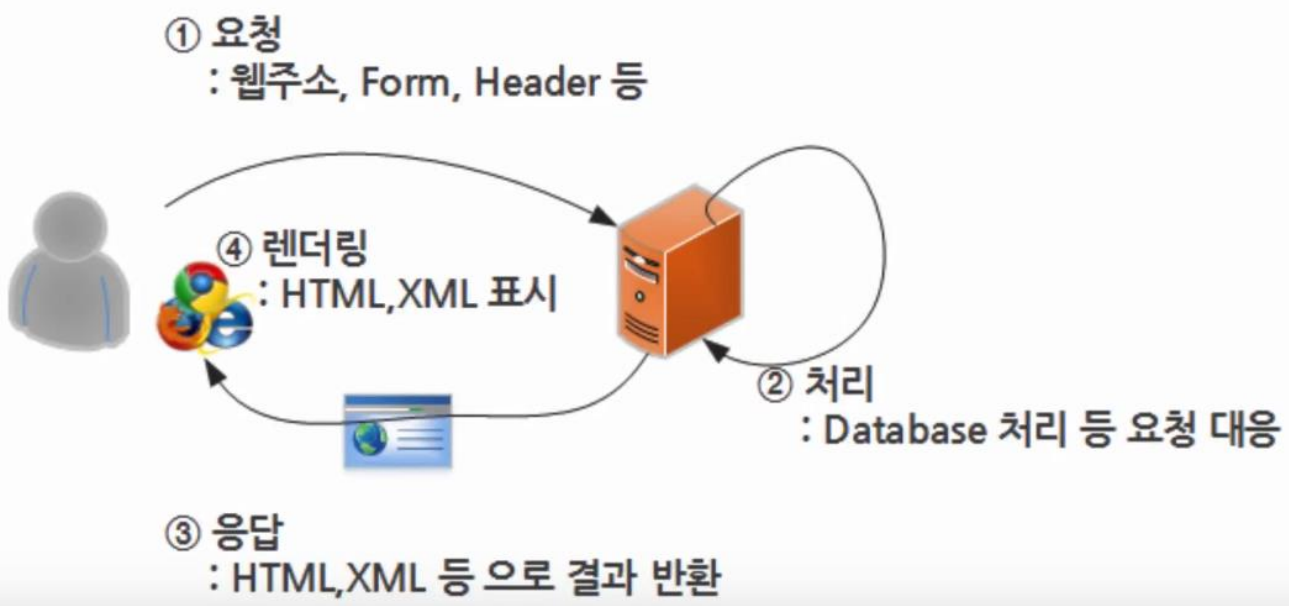


  - 여섯 개의 노드와 아홉 개의 링크로 이루어진 하이퍼텍스트 구조
- HTML(Hyper Text Mark-up Language)라는 언어를 사용

웹(Web)

Web의 동작 원리

- 인터넷 브라우저 등에 웹 주소를 입력하면 원격지에 있는 서버(물리적 공간)를 찾아 접속
- 필요한 정보가 다시 우리 컴퓨터로 돌아와 다운로드
  - html, xml 등의 문서
- 브라우저는 다운로드된 HTML 문서 등을 우리가 눈으로 볼 수 있도록 해석하여 변환(렌더링)



웹(Web)

HTML

- 웹 상의 정보를 '구조적'으로 표현하기 위한 언어
- 제목 단락 링크 등 요소 (node) 표시를 위해 Tag (<>) 사용
- 모든 요소들은 꺽쇠 괄호 안에 둘러쌓여 있음
  - <title> Hello world </title> #제목 요소, 값은 Hello World
- 모든 HTML은 트리 구조의 포함관계를 가짐

```
<!doctype html>
<html>
  <head>
    <title>Hello HTML</title>
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```

HTML 구조 (tree)

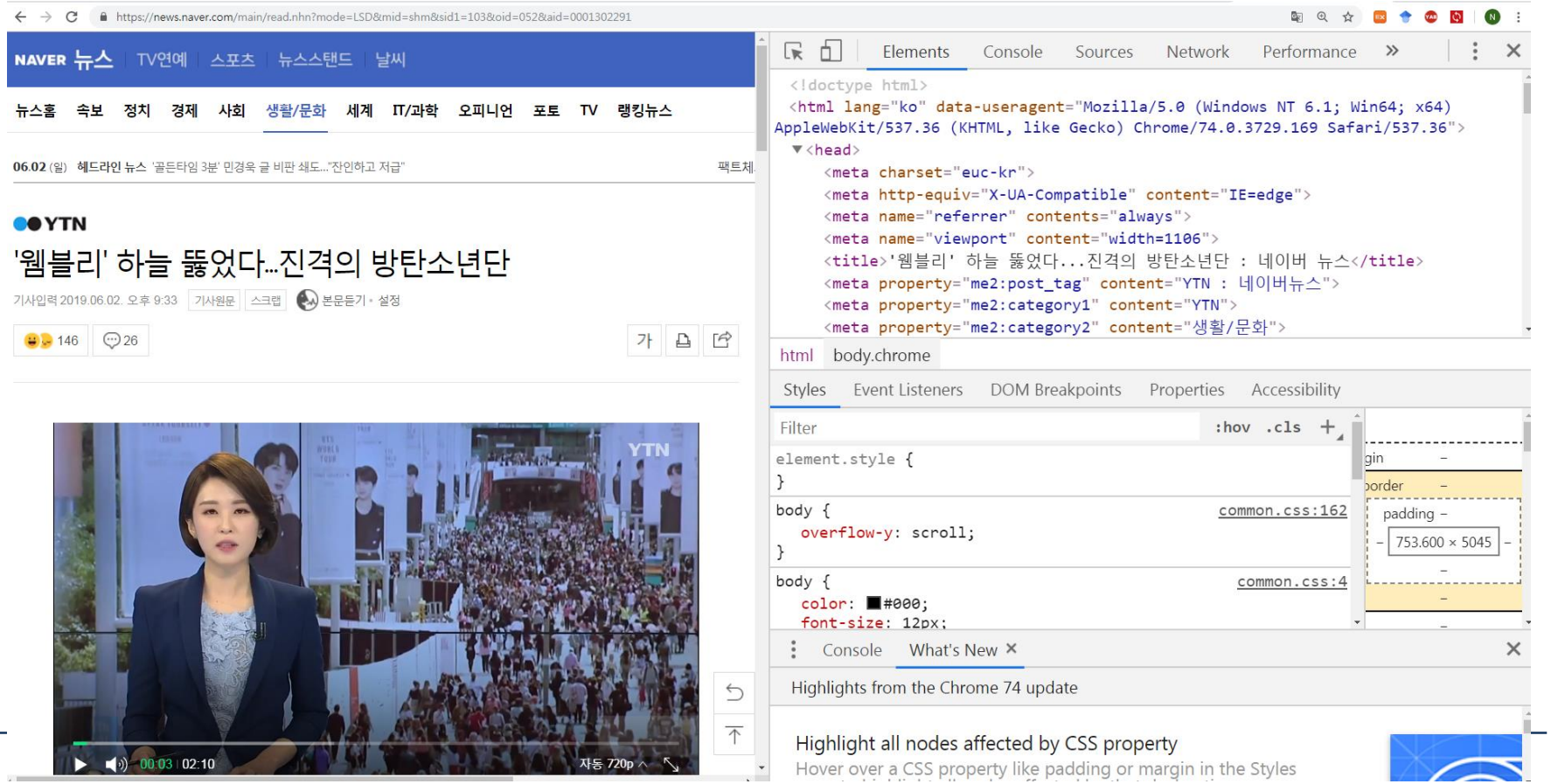
<html> – <head> – <title>
– <body> – <p>

Element, Attribute Value 이루어짐

<tag attribute1= " att_value1" attribute2="
att_value1 ">
보이는 내용(Value)
</tag>

웹(Web)

- HTML 예시
  - Chrome 오른쪽버튼 "검사(Inspect)" 또는 F12 버튼. Explorer 오른쪽 버튼 "소스보기"
  - https://news.naver.com/main/read.nhn?mode=LSD&mid=shm&sid1=103&oid=052&aid=0001302291



웹 크롤링

웹 상의 자료를 추출하는 것

- 다양한 정보들이 웹을 통해 공유됨
  - 환율정보 https://finance.naver.com/marketindex/?tabSel=exchange
  - 날씨정보 http://www.weather.go.kr/weather/main.jsp
  - 주가정보 https://finance.yahoo.com/
- HTML도 일종의 프로그래밍 언어로 페이지 생성 규칙이 존재
  - 규칙을 분석하여 데이터 추출 가능
- 추출된 데이터를 바탕으로 다양한 분석 가능

HTML Parsing

HTML Parsing

- 웹으로부터 데이터를 추출해 내는 행위
- 대부분의 웹은 사용자 요구에 따라 동적으로 생성됨
  - 예시) http://finance.naver.com/item/main.nhn?code=005930 (삼성전자)
  - 위에서 ? 앞부분이 페이지를 생성하는 파일, 그 뒤는 해당 프로그램의 변수
  - 뒤에 코드명만 바꾸어주면 다양한 주식정보를 받을 수 있다
- 이런 번호들의 리스트를 가지고 있다면 주식 데이터를 컴퓨터로 옮길 수 있음
- HTML 파싱을 위해서는 HTML 생성 규칙 파악
- HTML은 트리 구조 - 구조 파악 필요

HTML Parsing

- HTML 규칙 파악하기
  - http://finance.naver.com/item/main.nhn?code=005930
    - 참고: url 주소의 마지막 숫자는 종목 코드를 의미
    - 종목코드 검색 https://www.ktb.co.kr/trading/popup/itemPop.jspx

  - HTML 열어서 분석하기
    - 마우스 우클릭 – 페이지 소스 보기
  - HTML 파일에서 유일하게 위 데이터를 나타낼 수 있는 패턴을 찾아야 함
    - '종목 시세 정보'

HTML Parsing

HTML 규칙 파악하기

- <dl class="blind"> ~</dl> 사이에 데이터가 존재
- 각 데이터는 <dd> ~ </dd>로 나타내며 데이터 생성 순서는 일시, 종목 명 ~ 거래 대금 순서
- 이러한 구조를 파악할 수 있다

```
▼<dl class="blind">
    <dt>종목 시세 정보</dt>
    <dd>2022년 01월 19일 16시 11분 기준 장마감</dd>
    <dd>종목명 삼성전자</dd>
    <dd>종목코드 005930 코스피</dd>
    <dd>현재가 76,300 전일대비 하락 700 마이너스 0.91 퍼센트</dd>
    <dd>전일가 77,000</dd>
    <dd>시가 76,500</dd>
    <dd>고가 76,900</dd>
    <dd>상한가 100,000</dd>
    <dd>저가 76,100</dd>
    <dd>하한가 53,900</dd>
    <dd>거래량 10,543,644</dd>
    <dd>거래대금 805,979백만</dd>
  </dl>
```

HTML Parsing

- HTML Parsing 방법

  1) 파이썬 제공 HTML 파싱 모듈 활용

  2) 정규식 이용

HTML Parsing

BeautifulSoup을 이용한 parsing

- 파싱(Parsing)이란 웹 문서에서 원하는 패턴이나 순서로 자료를 추출해 가공하는 것을 의미

기본 모듈 임포트

```
# 기본모듈 임포트

## urllib은 웹에서 얻은 데이터를 다루는 파이썬 패키지. request는 웹 문서를 열어 데이터 읽어오는 모듈
import urllib.request as ur
## 웹문서를 구성하는 HTML과 XML문서에서 원하는 정보를 쉽게 추출할 수 있는 모듈을 모아놓은 패키지
from bs4 import BeautifulSoup as bs
```

HTML Parsing

- BeautifulSoup을 이용한 parsing

1. urlopen으로 웹사이트 정보 가져오기

웹 사이트에서 원하는 정보를 요청해서 그 결과물을 반환하는 명령

```
urllib.request.urlopen('URL주소')
```

```
[ ]  # 접속하고 싶은 웹 사이트의 url주소를 url 객체에 문자열로 저장
     url = 'http://finance.naver.com/item/main.nhn?code=005930'

     html = ur.urlopen(url)
     html
```

```
<http.client.HTTPResponse at 0x7ff8dce83f60>
```

html에 어떤 내용이 있는지 read()로 간단히 살펴볼 수 있음

```
[ ]  html.read()[:100]
```

```
b"₩n₩n₩t₩n₩t₩n₩t₩n₩t₩n<html lang='ko'>₩n<head>₩n₩n₩n₩t₩n₩t₩t<title>₩xbb
```

HTML Parsing

BeautifulSoup을 이용한 parsing

2. 뷰티풀수프로 자료형 변환

html 객체에 저장한 자료를 정보를 쉽게 추출할 수 있는 형태, 즉 파싱(parsing)하기 쉬운 형태로 변환

```
bs(html.read(), 'html.parser')
```

- 파싱(parsing)이란 웹 문서에서 원하는 패턴이나 순서로 자료를 추출해 가공하는 것을 말함

```
[ ]  soup = bs(html.read(), 'html.parser')
```

```
[ ]  type(html), type(soup)
```

```
(http.client.HTTPResponse, bs4.BeautifulSoup)
```

```
[ ]  # 위의 과정은 다음과 같이 한 줄로 표현가능
     soup = bs(ur.urlopen(url).read(), 'html.parser')
```

HTML Parsing

- BeautifulSoup을 이용한 parsing

3. 특정 태그에서 텍스트만 추출

HTML 구조 살펴보기

```
<HTML>
<head>
  <title> 페이지 제목 </title>
</head>
<body>
  <h1> 글 제목 </h1>
  <p> 글 본문 </p>
</body>
</HTML>
```

이 구조를 참고해 웹 문서자료에서 원하는 요소를 찾음

HTML Parsing

- BeautifulSoup을 이용한 parsing

3. 특정 태그에서 텍스트만 추출

find_all로 원하는 태그만 모으기

- soup에 find_all메서드 이용하여 특정 태그만 모을 수 있음

```
soup.find_all(찾아낼 태그)
```

```
▼<dl class="blind">
   <dt>종목 시세 정보</dt>
   <dd>2021년 01월 14일 16시 10분 기준 장마감</dd>
   <dd>종목명 삼성전자</dd>
   <dd>종목코드 005930 코스피</dd>
   <dd>현재가 89,700 전일대비 보합 0  0.00 퍼센트</dd>
   <dd>전일가 89,700</dd>
   <dd>시가 88,700</dd>
   <dd>고가 90,000</dd>
   <dd>상한가 116,500</dd>
   <dd>저가 88,700</dd>
   <dd>하한가 62,800</dd>
   <dd>거래량 26,127,127</dd>
   <dd>거래대금 2,332,652백만</dd>
 </dl>
```

HTML Parsing

BeautifulSoup을 이용한 parsing

3. 특정 태그에서 텍스트만 추출

- <dl> 태그를 모두 추출하기

```
soup.find_all('dl')
```

```
[<dl class="blind">
<dt>종목 시세 정보</dt>
<dd>2021년 01월 14일 16시 10분 기준 장마감</dd>
<dd>종목명 삼성전자</dd>
<dd>종목코드 005930 코스피</dd>
<dd>현재가 89,700 전일대비 보합 0  0.00 퍼센트</dd>
<dd>전일가 89,700</dd>
<dd>시가 88,700</dd>
<dd>고가 90,000</dd>
<dd>상한가 116,500</dd>
<dd>저가 88,700</dd>
  ...
</dl>, <dl>
<dt>EPS(지배주주지분)</dt>
<dd>지배주주지분 당기순이익/지배주주 평균발행주식수(우선주+보통주)</dd>
<dt>BPS(지배주주지분</dt>
<dd>지배주주지분 귀속 순자산/지배주주 기말발행주식(우선주+보통주)</dd>
</dl>]
```

HTML Parsing

BeautifulSoup을 이용한 parsing

3. 특정 태그에서 텍스트만 추출

- class가 "blind"인 <dl> 태그 모두 추출하기

```
info = soup.find_all('dl',{"class": "blind"})
for i in info:
  print(i.text)
```

```
종목 시세 정보
2021년 01월 14일 16시 10분 기준 장마감
종목명 삼성전자
종목코드 005930 코스피
현재가 89,700 전일대비 보합 0  0.00 퍼센트
전일가 89,700
시가 88,700
고가 90,000
상한가 116,500
저가 88,700
하한가 62,800
거래량 26,127,127
거래대금 2,332,652백만


삼성전자
오늘의시세 89,700 포인트
0 포인트 보합
0.00%
```

- 두 개가 추출되는 것을 확인할 수 있으며 우리가 원하는 것은 첫번째 태그인 것을 확인할 수 있음
- 다음과 같이 첫번째 태그만 이용 가능

```
print(info[0].text)
```

- 또는 find() 메서드를 이용해서 첫번째 태그만 추출 가능

```
info = soup.find('dl',{"class": "blind"})
print(info.text)
```

HTML Parsing

BeautifulSoup을 이용한 parsing

- 속성과 속성값
  - 앞의 예제에서 주가 정보는 <dl>태그에 저장되어있었으며, 시작 태그를 보면 <dl class="blind"> 라고 작성되어 있음.
  - 이 때 class를 이 <dl> 태그의 속성(attribute)이라 하고, "blind"는 속성값(attribute value)이라고 함

  - 속성이란 HTML 요소에 좀 더 구체적인 기능을 추가하기 위해 시작 태그 안에 지정하는 것 의미
  - 이 속성에 입력할 구체적인 값이 속성값임

  - 속성과 속성값을 이용하면 원하는 정보를 좀 더 쉽게 찾을 수 있음

Regular Expression

정규식 – Regular Expression

- 정규 표현식, regexp 또는 regex 등으로 불림
- 복잡한 문자열 패턴을 정의하는 문자 표현 공식
- 특정한 규칙을 가진 문자열의 집합을 추출

예시	형식	정규식
전화번호	010-0000-0000	^\d{3}\-\d{4}\-\d{4}$
IP 주소	203.252.101.40	^\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}$

- ^는 시작 $는 끝을 의미
- d{3} : digit이 3개, 즉 숫자 3개를 의미
- d{1,3} : 숫자가 1개~3개까지 존재할 수 있음을 의미

Regular Expression

정규식 for HTML Parsing

- 주민등록 번호, 전화번호, 우편번호, 도서 ISBN 등 형식이 있는 문자열을 원본 문자열으로부터 추출함
- HTML 역시 tag(<>)를 사용한 일정한 형식이 존재하여 정규식으로 추출이 용이함

  -관련자료 : http://www.nextree.co.kr/p4327/

정규식 문법

- 문법 자체는 매우 방대, 스스로 찾아서 하는 공부 필요
- 필요한 경우 인터넷 검색을 통해 찾을 수 있음
- 기본적인 것을 공부한 후 넓게 적용하는 것이 중요

Regular Expression

정규식 연습장 사용하기

- 1) 정규식 연습장 (http://www.regexr.com/) 으로 이동
- 2) 테스트하고 싶은 문서를 Text 란에 삽입
- 3) 정규식을 사용해서 찾아보기

정규식 기본 문법 #1

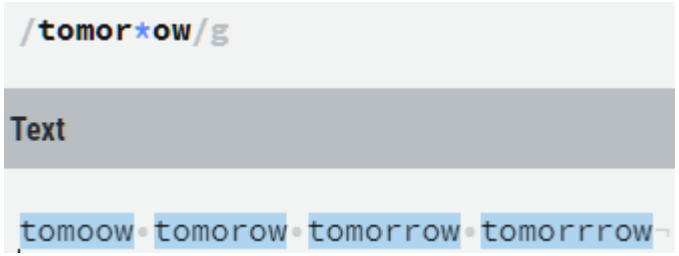
- 문자 클래스 [ ] : [와] 사이의 문자들과 매치한다는 의미
  - 예) [abc] : 해당 글자가 a,b,c중 하나이다.

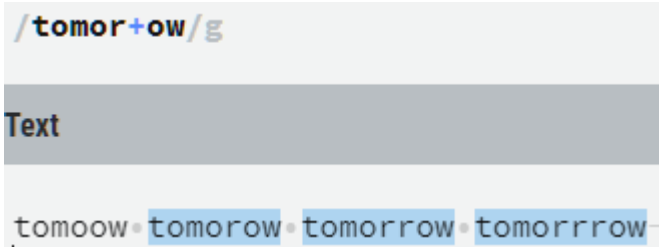
```
RegExr was created by gskinner.com, and is proudly hosted by Media Temple.
```

- '-'를 이용해 사용 범위를 지정할 수 있음
  - 예) [a-zA-Z] – 알파벳 전체 (대, 소문자 구분 X), [0-9] – 숫자 전체

정규식 기본 문법 – 메타 문자

- 정규식 표현을 위해 원래 의미가 아닌 다른 용도로 사용되는 문자
  - Ex) . ^ $ * + ? { } [ ] ₩ | ( )
- '.' : 줄바꿈 문자인 ₩n을 제외한 모든 문자와 매치
  - a . b : a로 시작해서 b로 끝나는 모든 글자
- '*' : 앞에 있는 글자를 반복해서 나올 수 있음
- '+' : 앞에 있는 글자를 최소 1회 이상 반복





Regular Expression

정규식 기본 문법 – 메타 문자

- 정규식 표현을 위해 원래 의미가 아닌 다른 용도로 사용되는 문자
  - Ex) . ^ $ * + ? { } [ ] ₩ | ( )
- {m, n} – 반복 횟수를 지정 (최소 m개, 최대 n개)
  - IP주소 : [0-9]{1,3}₩.[0-9]{1,3}₩.[0-9]{1,3}₩.[0-9]{1,3}
- '?' : 반복 횟수가 0-1회 {0,1}을 의미
  - 전화번호 : 01[01]?-[0-9]{4}-[0-9]{4}
- '|' : or
- '^' : not