

데이터 수집

BUSINESS DATA ANALYSIS AND VISUALIZATION USING AI

02

APPLICATION

ARTIFICIAL INTELLIGENCE

ARTIFICIAL INTELLIGENCE (AI) REFERS TO THE SIMULATION OF HUMAN INTELLIGENCE IN MACHINES THAT ARE PROGRAMMED TO THINK LIKE HUMANS AND Imitate THEIR ACTIONS

THE TERM MAY ALSO BE APPLIED TO ANY MACHINE THAT PERFORMS TASKS ASSOCIATED WITH A HUMAN, SUCH AS LEARNING AND PROBLEM-SOLVING

Artificial intelligence (AI) refers to the simulation of human

Notice

© All Rights Reserved. All Rights Reserved.

ARTIFICIAL INTELLIGENCE (AI) REFERS TO THE SIMULATION OF HUMAN INTELLIGENCE IN MACHINES THAT ARE PROGRAMMED TO THINK LIKE HUMANS AND Imitate THEIR ACTIONS

이 교육과정은 교육부 ‘성인학습자 역량 강화 교육콘텐츠 개발’ 사업의 일환으로써
교육부로부터 예산을 지원 받아 고려사이버대학교가 개발하여 운영하고 있습니다.
제공하는 강좌 및 학습에 따르는 모든 산출물의 저작권은 교육부, 한국교육학술정보원,
한국원격대학협의외와 고려사이버대학교가 공동 소유하고 있습니다.

학습목표

Artificial intelligence (AI) refers to the simulation of human

GOALS

ARTIFICIAL INTELLIGENCE (AI) REFERS TO THE SIMULATION OF HUMAN INTELLIGENCE IN MACHINES THAT ARE PROGRAMMED TO THINK LIKE HUMANS AND MIMIC THEIR ACTIONS

THE TERM MAY ALSO BE APPLIED TO ANY MACHINE THAT EMULATES TASKS ASSOCIATED WITH A HUMAN MIND SUCH AS LEARNING AND PROBLEM-SOLVING

1 정형데이터와 비정형데이터의 수집 방법과 기술을 이해하고 이를 비교하여 설명할 수 있다.

2 파이썬을 활용 분석 위한 데이터를 수집할 수 있다.



1 정형데이터 수집

2 비정형데이터 수집

3 데이터 수집 실습

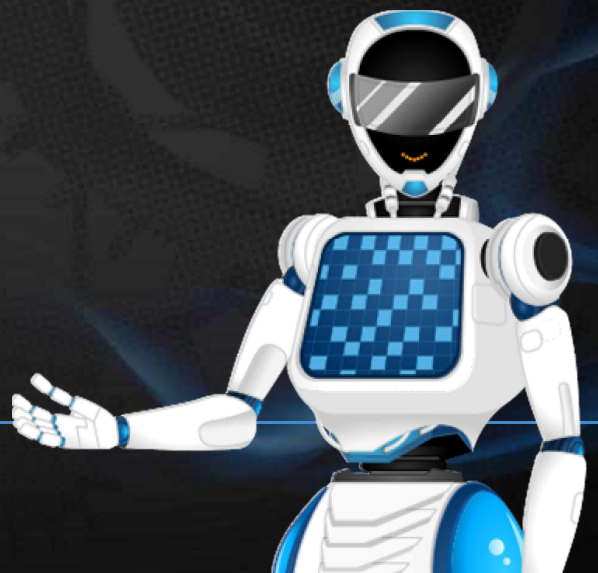
THE TERM MAY ALSO BE APPLIED TO ANY MACHINE THAT EMULATES TASKS ASSOCIATED WITH A HUMAN MIND SUCH AS LEARNING AND PROBLEM-SOLVING

ARTIFICIAL INTELLIGENCE (AI) REFERS TO THE SIMULATION OF HUMAN INTELLIGENCE IN MACHINES THAT ARE PROGRAMMED TO THINK LIKE HUMANS AND MIMIC THEIR ACTIONS

CONTENTS

학습내용

Artificial intelligence (AI) refers to the simulation of human



Artificial intelligence (AI) refers
to the simulation of human

APPLICATION

01

정형 데이터 수집



01 정형 데이터 수집

01 분석 목적 설정

가트너 그룹 빅데이터 분석의 목적

• The Big Data Value Model (2015) •

- ① 고객 인사이트 Customer Insight
- ② 제품 및 절차 효율성 Product & Process Efficiency
- ③ 디지털 제품 및 서비스 Digital Products & Service
- ④ 운영의 탁월성 Operational Excellence
- ⑤ 디지털 마케팅 Digital Marketing
- ⑥ 위기 관리시스템 Risk Management and Compliance

02 데이터 확보 계획 수립

1. 필요 데이터 정의

- ▶ 빅데이터 분석 목적에 적합한 데이터 목록 작성, 확보 가능여부 점검

2. 기관 내 데이터 확보 방안 검토

- ▶ 필요 데이터에 대한 기관 보유 데이터 현황조사를 수행
- ▶ 내부 보유 데이터라도 데이터 품질성, 업데이트 여부, 축적된 데이터량, 코드 변경 등 확인 필요
- ▶ 공유 가능 여부 확인 / 법, 보안, 개인정보 문제 확인

02 데이터 확보 계획 수립

3. 기관 외 데이터 수집 방안 검토

- ▶ 법적 제약사항이 없는지 확인
- ▶ 데이터 구매비용 고려

4. 직접 수집 및 공공기관 오픈 API

- ▶ 공공데이터포털(<http://www.data.go.kr>)
- ▶ 기상자료개방 포털(data.kma.go.kr/cmmn/main.do)
- ▶ 공간정보 오픈플랫폼(www.vworld.kr)

5. 데이터 확보 계획 수립 시 고려 사항

- ▶ 다양한 데이터 원천을 파악 정도
- ▶ 외부 기관과의 협의 여부
- ▶ 외부 데이터와 내부 데이터간의 통합이 가능 여부
- ▶ 수집관련 보안사항 및 개인정보보호 관련 문제 등 점검

- 1 보유 데이터와 분석 방법론을 활용하여 분석 목적을 해결하기 위한 분석 방안 제시
- 2 분석 방안은 기술적 해결책과 무관하게 정책 결정자와 최종 사용자 위주로 수립
- 3 분석 방안이 논리적으로 이해 가능하고, 구체적으로 수립되었는지 확인
- 4 분석 결과에 대한 성과 목표와 활용 방안 도출

03 데이터 분석 방안 설정

분석 방안 수립 시 세부 고려 사항

- 분석과 관련된 현재 이슈에 대한 기초 분석 단계를 설정하고 기초 분석 단계에서는 해당 이슈 관련 현황을 도출
- 어떠한 데이터를 활용해서 어떻게 이슈를 해결할 것인지 분석 목표와 이에 따른 방안을 제시
- 과제 관련 학습을 통해 과제의 이해도를 높이고, 국내외 선행 사례 또는 전문가와 함께 실현 가능한 분석 방안 구성과 성과 목표를 수립
- 데이터를 추가하거나 기존의 데이터를 변경하고, 선행 사례의 분석 방법론에 대한 변경 사항들을 고려하여 이슈해결 방안을 모색

04 데이터 유형

데이터 유형	데이터 종류	수집기술
정형 데이터	RDB, 스프레드 시트	ETL, FTP, Open API
반정형 데이터	HTML, XML, JSON, 웹문서, 웹로그, 센서 데이터	Crawling, RSS, Open API, FTP
비정형 데이터	소셜 데이터, 문서(워드, 한글), 이미지, 오디오, 비디오, IoT	Crawling, RSS, Open API, Streaming, FTP

05 RDB: 관계형데이터베이스

빅데이터 활용 단계별 업무절차 및 기술 활용 매뉴얼 (Version 1.0) 중

RDB: 관계형데이터베이스

- ① 관계형 데이터를 저장하거나, 수정하고 관리할 수 있게 해주는 데이터베이스
- ② SQL 문장을 통하여 데이터베이스의 생성, 수정 및 검색 등 서비스를 제공
- ③ oracle, mssql, mySQL, sybase, MPP DB

06 CSV : 스프레드 시트

CSV : 스프레드 시트

- ① .CSV 는 구분자(일반적으로 쉼표)로 데이터 내용을 나누어 놓은 엑셀과 같은 형태
- ② 외부 파일로 존재할 수도 있고, 프로그래밍을 완성한 데이터를 다시 하나의 파일로 저장할 수 있음
- ③ 한 개 이상의 레코드를 갖고 있고 레코드는 한 개 이상의 필드를 보유
- ④ 레코드는 줄바꿈으로 구분

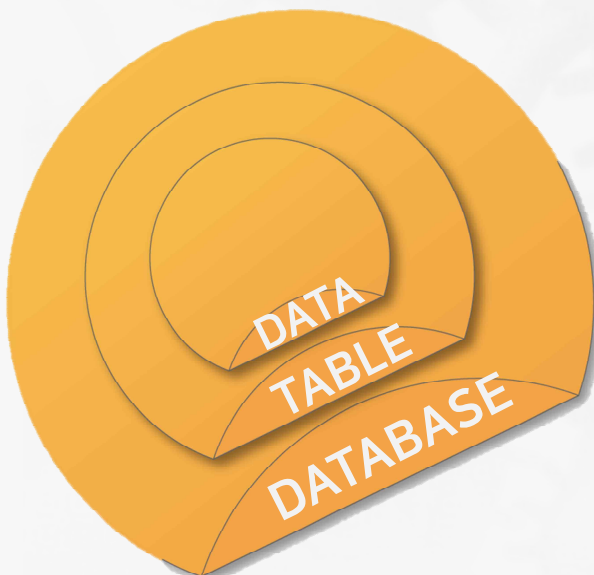
07 RDB : 관계형 데이터베이스

RDB : 관계형 데이터베이스

2개 이상의 데이터베이스 또는 테이블을 연결하기 위해 고유한 식별자를 사용하는 데이터베이스(DB)로서, 각각의 항목과 그 속성이 다른 모든 항목 및 그의 속성과 연결될 수 있도록 구성된 자료 구조

- 자료가 다중 연결되어 있어 각각의 다른 필드들과 연결되도록 하는 강력하고 유연성 있는 데이터베이스의 종류
- 보통 관계형 질의(Query)는 하나 이상의 필드에 특정 조건을 주어 그것들을 만족시키는 레코드를 찾게 함

07 RDB : 관계형 데이터베이스



07 RDB : 관계형 데이터베이스

MySQL

- ① 2016년 기준 80% 이상의 시장 점유율을 차지하고 있는 관계형 데이터베이스 관리 시스템(RDBMS)로 오픈소스로 개발됨
- ② GNU GPL(GNU General Public License)과 상업용 라이선스의 이중 라이선스로 관리 중임
- ③ 표준 데이터베이스 질의 언어인 구조화 질의 언어(SQL: Structured Query Language)를 사용

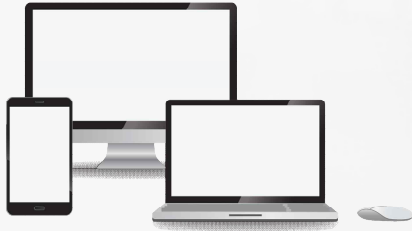
07 RDB : 관계형 데이터베이스

MySQL

- ④ 빠르고, 유연하며, 사용하기 쉬우며 다중 사용자, 다중 스레드(thread)를 지원
- ⑤ C, C++, 에펠(Eiffel), 자바, 펄, PHP, 파이썬(Python) 스크립트 등을 위한 응용 프로그램 인터페이스(API) 제공
- ⑥ 유닉스나 리눅스, 윈도우 운영 체제 등에서 사용

07 RDB : 관계형 데이터베이스

MySQL Client(s)



MySQL Server
(IPAddress: port)



Artificial intelligence (AI) refers to the simulation of human

APPLICATION

02

비정형 데이터 수집

02 비정형 데이터 수집

01 데이터 유형

데이터 유형	데이터 종류	수집기술
정형 데이터	RDB, 스프레드 시트	ETL, FTP, Open API
반정형 데이터	HTML, XML, JSON, 웹문서, 웹로그, 센서 데이터	Crawling, RSS, Open API, FTP
비정형 데이터	소셜 데이터, 문서(워드, 한글), 이미지, 오디오, 비디오, IoT	Crawling, RSS, Open API, Streaming, FTP

01 데이터 유형

빅데이터 활용 단계별 업무절차 및 기술 활용 매뉴얼 (Version 1.0) 중

NoSQL

- ① Not-Only SQL의 약자이며, 비관계형 데이터 저장소로, 기존의 전통적인 방식의 관계형 데이터베이스와는 다르게 설계된 데이터베이스
- ② 테이블 스키마(Table Schema)가 고정되지 않고, 테이블 간 조인(Join) 연산을 지원하지 않으며, 수평적 확장(Horizontal Scalability)이 용이
- ③ key-value, Document key-value, column 기반의 NoSQL이 주로 활용 중
- ④ MongoDB, Cassandra, HBase, Redis

03 텍스트 파일의 종류 및 유형

1. 텍스트 파일

텍스트 데이터를 포함하고 있는 파일

2. 바이너리 파일

텍스트가 아닌 데이터를 포함하고 있는 파일

3. 플레인 텍스트 파일

Sequence, EOF 등 몇가지 조건에 부합하는 파일

4. 비정형 파일

소셜, 이메일, SNS 등 라인 구분 외의 별다른 구조적 속성이 보이지 않는 텍스트 파일

03 텍스트 파일의 종류 및 유형

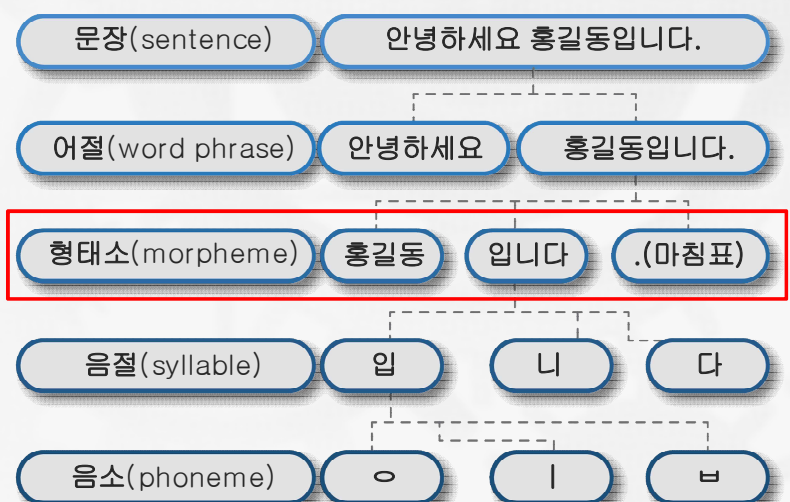
5. 반정형 파일

html, xml, yaml, json 등의 구조적 형태 안에서 자유롭게 쓰인 텍스트 파일

6. 정형 파일

엑셀이나 관계형 데이터베이스와 같이 표 형식의 텍스트 파일. CSV가 대표적이며, 텍스트 마이닝을 할 때는 콤마(,)를 구분하기 위하여 탭(tab)을 구분자로 사용한다.

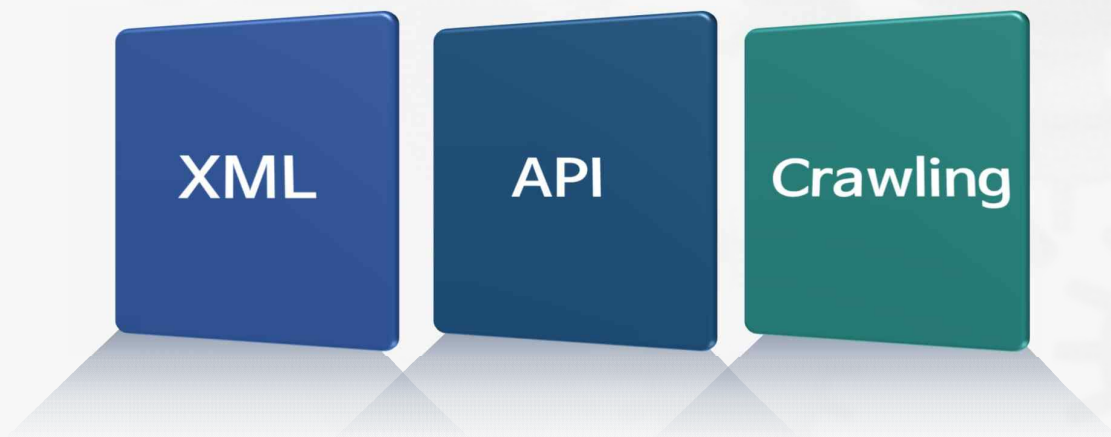
04 텍스트 마이닝 구조



05 텍스트 데이터 수집

▶ 텍스트 분석을 위한 텍스트 데이터 수집 방법

대표적인 방법 3가지



06 웹크롤링

웹크롤링 Web scraping

- ① 컴퓨터 소프트웨어 기술로 웹 사이트들에서 원하는 정보를 추출하는 것을 의미한다.
- ② 일반적으로 사람이 일일이 웹 페이지 데이터들을 모으는것이 힘들기 때문에 웹 크롤링이 필요하다.

06 웹크롤링

웹크롤링 Web scraping

예시

음악 사이트에서 차트 순위가 필요하다고 가정하면 일일이 찾아 저장한 후 다른 사람에게 알리거나 저장을 할 수도 있지만 차트 순위 데이터를 다른 프로그램이나 소프트웨어를 만들 때 저장하고 사용하려면 DB 형태로 저장할 필요가 있다.
또한 순위 뿐만 아니라 가수명, 제목, 카테고리 등등 여러 정보도 있을 수 있다.

07 파이썬에서 웹크롤링 진행하기

- ◆ 파이썬에서 웹크롤링을 진행하려면 **Beautifulsoup** 모듈이 필요
- ◆ **Beautifulsoup**를 사용하면 웹문서의 구조를 찾아낼 수 있고 html 파일을 가져오거나 직접 웹에서 소스를 가져올 수 있다.

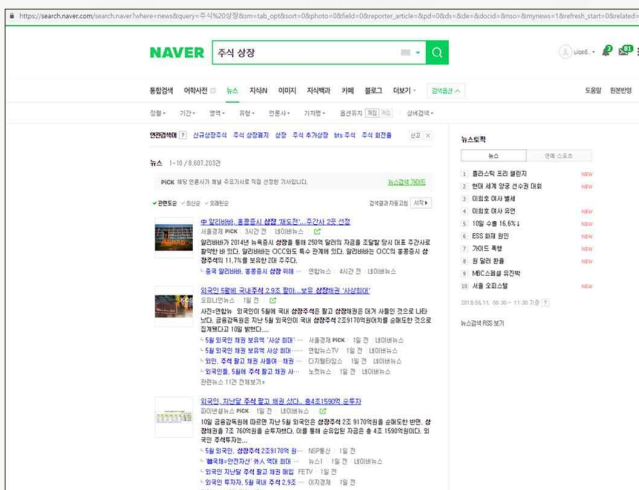
07 파이썬에서 웹크롤링 진행하기

- 1 크롤링을 위해 필요한 파이썬 모듈들을 불러온다.
- 2 BeautifulSoup를 이용하여 html 파일을 가져오고 다중 페이지 크롤링을 위하여 datetime 모듈을 사용하고 request를 통하여 웹소스를 가져오고 re를 통하여 전처리를 시작한다.

```
from bs4 import BeautifulSoup
from datetime import datetime
import requests
import pandas as pd
import re
```

07 파이썬에서 웹크롤링 진행하기

- 3 네이버뉴스의 기사 제목과 기사링크 그리고 언론사 목록을 가져오기위해 먼저 네이버 뉴스검색 주소를 가져온다.



07 파이썬에서 웹크롤링 진행하기

- 3 네이버뉴스의 기사 제목과 기사링크 그리고 언론사 목록을 가져오기위해 먼저 네이버 뉴스검색 주소를 가져온다.

request를 이용해 크롤링해서 가져온html 파일

```
html
<!doctype html> <html lang="ko"> <head> <meta charset="utf-8"> <meta name="referrer" content="always"> <meta name="format-detection" c
ontent="telephone=no,address=no,email=no"> <meta name="viewport" content="width=device-width,initial-scale=1.0,maximum-scale=2.0"> <meta
property="og:title" content="주식 상장 : 네이버 뉴스검색"/> <meta property="og:image" content="https://ssl.pstatic.net/sstatic/search/co
mmon/og_v3.png"> <meta property="og:description" content="주식 상장"의 네이버 뉴스검색 결과입니다."/> <meta name="description" lang="k
o" content="주식 상장"의 네이버 뉴스검색 결과입니다."/> <title>주식 상장 : 네이버 뉴스검색</title> <link rel="shortcut icon" href="htt
ps://ssl.pstatic.net/sstatic/search/favicon/favicon_140327.ico"> <link rel="search" type="application/opensearchdescription+xml" href
="https://ssl.pstatic.net/sstatic/search/opensearch-description.https.xml" title="Naver" /> <link rel="stylesheet" type="text/css" href
="https://ssl.pstatic.net/sstatic/search/pc/css/search1_190523.css"> <link rel="stylesheet" type="text/css" href="https://ssl.pstatic.ne
t/sstatic/search/pc/css/search2_190523.css"> <link rel="stylesheet" type="text/css" href="https://ssl.pstatic.net/sstatic/search/pc/css/
api_atomp_190523.css"> <script type="text/javascript"> naver = window.naver || {}; naver.search = naver.search || {}; var g_d = 0; if (!
String.prototype.trim) { String.prototype.trim = function () { return this.replace(/^[#s#u#f#x#A#0#]+|[#s#u#f#x#A#0#]+$/g, ''); }; } function urlencode (q) { return escape(q).replace(/#%2B/g, "%2B"); } function urlexpand (url) { var href = document.location.href; if
(url == "") return href; if (url.match(/^[-.A-Za-z]+:/)) return url; if (url.charAt(0) == '#') return href.split("#")[0] + url; if
(url.charAt(0) == '?') return href.split("?")[0] + url; if (url.charAt(0) == '/') return href.replace(/([:\/])#/, ".$1") + ur
l; return href.substring(0, href.lastIndexOf("/") + 1) + url; } naver.search.error = (function () { var errorList = Array(); return { add : function (s) { errorList.push(s); }, clear : function () { delete errorList; }, get : function (s) { return errorList; }, getStr
ing : function (d) { if (typeof d == '#undefined') d = '#'; return errorList.join(d); } })(); naver.search.cookie = (function ()
{ return { set : function (key, value, expire, domain) { var cookie = key + "=" + escape(value); if (typeof expire != '#undefined') {
if (expire instanceof Date) { cookie = cookie + "; expires=" + expire.toUTCString(); } else { var exdate = new Date(new Date()).getTime
+ 86400000; cookie = cookie + "; expires=" + exdate.toUTCString(); } } } })();
```

07 파이썬에서 웹크롤링 진행하기

- 3 네이버뉴스의 기사 제목과 기사링크 그리고 언론사 목록을 가져오기위해 먼저 네이버 뉴스검색 주소를 가져온다.

beautifulsoup을 이용해 받은 웹소스에서 가져온html 파일

```
SOUP
<!DOCTYPE doctype html>
<html lang="ko"> <head> <meta charset="utf-8"> <meta content="always" name="referrer"/> <meta content="telephone=no,address=no,email=n
o" name="format-detection"/> <meta content="width=device-width,initial-scale=1.0,maximum-scale=2.0" name="viewport"/> <meta content="주
식 상장 : 네이버 뉴스검색" property="og:title"> <meta content="https://ssl.pstatic.net/sstatic/search/common/og_v3.png" property="og:ima
ge"/> <meta content="주식 상장"의 네이버 뉴스검색 결과입니다."/> <meta content="주식 상장"의 네이버 뉴스검색
결과입니다."/> <title>주식 상장 : 네이버 뉴스검색</title> <link href="https://ssl.pstatic.net/sstatic/search
h/favicon/favicon_140327.ico" rel="shortcut icon"/> <link href="https://ssl.pstatic.net/sstatic/search/opensearch-description.https.xml"
rel="search" title="Naver" type="application/opensearchdescription+xml"> <link href="https://ssl.pstatic.net/sstatic/search/pc/css/search
1_190523.css" rel="stylesheet" type="text/css"/> <link href="https://ssl.pstatic.net/sstatic/search/pc/css/search2_190523.css" rel="styl
esheet" type="text/css"/> <link href="https://ssl.pstatic.net/sstatic/search/pc/css/api_atomp_190523.css" rel="stylesheet" type="text/cs
s"/> <script type="text/javascript"> naver = window.naver || {}; naver.search = naver.search || {}; var g_d = 0; if (!String.prototype.t
rim) { String.prototype.trim = function () { return this.replace(/^[#s#u#f#x#A#0#]+|[#s#u#f#x#A#0#]+$/g, ''); }; } function urlencode (q)
{ return escape(q).replace(/#%2B/g, "%2B"); } function urlexpand (url) { var href = document.location.href; if (url == "") return href;
if (url.match(/^[-.A-Za-z]+:/)) return url; if (url.charAt(0) == '#') return href.split("#")[0] + url; if (url.charAt(0) == '?') retur
n href.split("?")[0] + url; if (url.charAt(0) == '/') return href.replace(/([:\/])#/, ".$1") + url; return href.substring(0, href.
lastIndexOf("/") + 1) + url; } naver.search.error = (function () { var errorList = Array(); return { add : function (s) { errorList.push
(s); }, clear : function () { delete errorList; }, get : function (s) { return errorList; }, getString : function (d) { if (typeof d
== '#undefined') d = '#'; return errorList.join(d); } })(); naver.search.cookie = (function () { return { set : function (key, valu
e, expire, domain) { var cookie = key + "=" + escape(value); if (typeof expire != 'undefined') { if (expire instanceof Date) { cookie =
cookie + "; expires=" + expire.toUTCString(); } else { var exdate = new Date(new Date()).getTime + 86400000; cookie = cookie + "; expires=" + exdate.toUTCString(); } } } })();
```

07 파이썬에서 웹크롤링 진행하기

4 url로 지정후 request를 사용해 웹소스를 가져오고 BeautifulSoup를 이용하여 html파일을 가져온다.

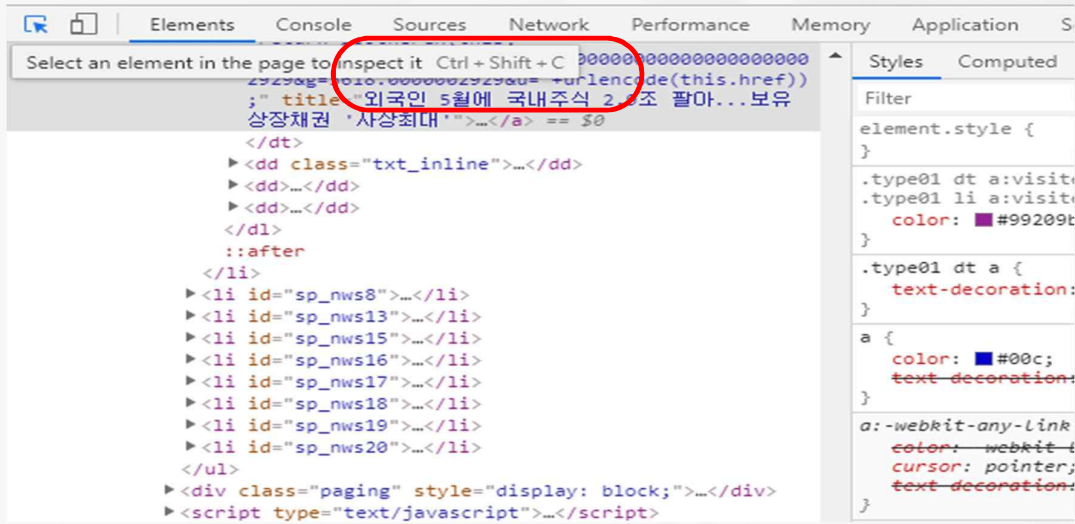
```
url = 'https://search.naver.com/search.naver?where=news&query=%EC%A3%BC%EC%8B%9D%20%EC%83%81%EC%9E%A5&sm=tab_opt&sort=0&photo=0&field=0&re
response = requests.get(url)
html = response.text
soup = BeautifulSoup(html, 'html.parser')
```

07 파이썬에서 웹크롤링 진행하기

5 추가적인 코딩으로 크롤링하고 싶은 페이지에 들어가 f12를 눌러 화면을 불러온다.

07 파이썬에서 웹크롤링 진행하기

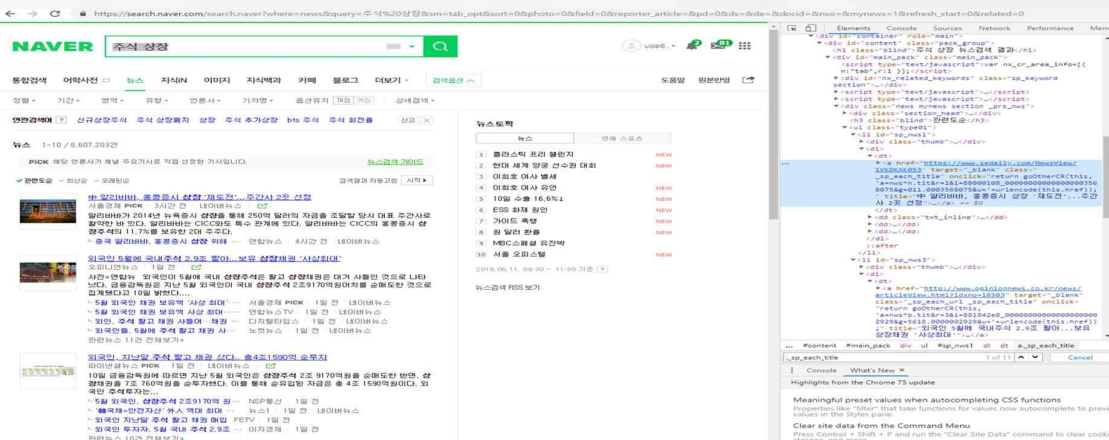
6 기사 제목을 클릭하기 위해 다음의 버튼을 누른다. Ctrl + Shift + C



07 파이썬에서 웹크롤링 진행하기

6 기사 제목을 클릭하기 위해 다음의 버튼을 누른다. Ctrl + Shift + C

❖ 좌측상단의 버튼을 누르고 커서를 이동하여 원하는 제목위에 올린다.



07 파이썬에서 웹크롤링 진행하기

- 6 기사 제목을 클릭하기 위해 다음의 버튼을 누른다.
Ctrl + Shift + C

◆ 제목 위에 마우스를 올려놓았을 때, 아래 부분이 선택되는 것을 확인

```
...
▶ <a href="https://www.sedaily.com/NewsView/
1VKDKAK053" target="_blank" class="
_sp_each_title" onclick="return goOtherCR(this,
'a=nws*h.tit&r=1&i=88000108_000000000000000000356
8075&g=011.0003568075&u='+urlencode(this.href));
" title="中 알리바바, 홍콩증시 상장 '재도전'...주간
사 2곳 선정">...</a> == $0
```

07 파이썬에서 웹크롤링 진행하기

- 6 기사 제목을 클릭하기 위해 다음의 버튼을 누른다.
Ctrl + Shift + C

◆ 확인을 위해 다른 제목도 클릭하여 확인한다.

```
▶ <a href="http://www.opinionnews.co.kr/news/
articleView.html?idxno=18503" target="_blank"
class="_sp_each_url _sp_each_title" onclick=
"return goOtherCR(this,
'a=nws*b.tit&r=3&i=881842e8_00000000000000000000
2929&g=5618.0000002929&u='+urlencode(this.href))
;" title="외국인 5월에 국내주식 2.9조 팔아...보유
상장채권 '사상최대'">...</a> == $0
```

07 파이썬에서 웹크롤링 진행하기

- 6 기사 제목을 클릭하기 위해 다음의 버튼을 누른다.
Ctrl + Shift + C

- ▶ 제목에 해당되는 웹소스부분에서 어떤 부분에 해당하는지 살펴본다.
._sp_each_title에 해당됨을 확인

```

<dt>
  <dd class="txt_inline">...</dd>
... #main_pack div ul #sp_nws3 dl dt a._sp_each_url._sp_each_title

```

07 파이썬에서 웹크롤링 진행하기

- 7 BeautifulSoup를 이용해 원하는 부분을 선택하고
리스트에 저장한다.

```

title_text = [] ## 빈리스트 만들기
atags = soup.select('._sp_each_title') ## html 소스에서 원하는 부분 선택
for atag in atags:
    title_text.append(atag.text) ## 제목만 가져온다

```

07 파이썬에서 웹크롤링 진행하기

- 7 **beautifulsoup**를 이용해 원하는 부분을 선택하고 리스트에 저장한다.

title_text

```
[ "中 알리바바, 홍콩증시 상장 '재도전'...주간사 2곳 선정",
  "외국인 5월에 국내주식 2.9조 팔아...보유 상장채권 '사상최대'",
  '외국인, 지난달 주식 팔고 채권 샀다.. 총4조1590억 순투자',
  '국내 주식형 펀드 자금 13일째 순유입',
  '강남 롯데백화점 건물주 되볼까...리츠 상장 속도내는 까닭은',
  "'정수기 필터 제조 기업' 한독크린텍, 코스닥 상장 시동",
  "한국투자증권, 첫 주식계좌 개설시 '국내 대표회사 주식' 지급",
  '주식 팔고 채권 산 외국인.. 총 4조 1590억 순투자',
  '6월 11일 오렌지라이프 주식현황 0.29% 34000원',
  '(오늘의 주가정보)세하 증권 정보 전망과 주식평가 11일 현재 1395원']
```

07 파이썬에서 웹크롤링 진행하기

- 8 링크를 가져오기 위하여 **._sp_each_title** 부분에 존재하는 하이퍼 링크 부분을 크롤링한다.

```
> <a href="http://www.opinionnews.co.kr/news/articleView.html?idxno=18503" target="_blank"
class="_sp_each_url _sp_each_title" onclick=
"return goOtherCR(this,
'a=nws*b.tit&r=3&i=881842e8_00000000000000000000
2929&g=5618.0000002929&u='+urlencode(this.href))
;" title="외국인 5월에 국내주식 2.9조 팔아...보유
상장채권 '사상최대'">...</a> == $0
```

07 파이썬에서 웹크롤링 진행하기

- 9 BeautifulSoup을 이용해 원하는 부분을 선택하고 하이퍼링크 부분을 가져와 리스트에 저장한다.

```
link_text = [] ## 빈 리스트 만들기
atags = soup.select('._sp_each_title') ## 원하는 부분 선택
for atag in atags:
    link_text.append(atag['href']) ## 선택된 부분에서 하이퍼링크 추출
|
```

07 파이썬에서 웹크롤링 진행하기

- 9 BeautifulSoup을 이용해 원하는 부분을 선택하고 하이퍼링크 부분을 가져와 리스트에 저장한다.

```
link_text
['https://www.sedaily.com/NewsView/1VKDKAK053',
'http://www.opinionnews.co.kr/news/articleView.html?idxno=18503',
'http://www.fnnews.com/news/201906081130165414',
'http://www.edaily.co.kr/news/newspath.asp?newsid=01351366622520736',
'http://biz.chosun.com/site/data/html_dir/2019/06/10/2019061001980.html',
'https://view.asiae.co.kr/article/2019061111180150499',
'http://www.babytimes.co.kr/news/articleView.html?idxno=28174',
'http://ilyo.co.kr/?ac=article_view&entry_id=337554',
'http://www.nbntv.co.kr/news/articleView.html?idxno=396847',
'http://www.seoulwire.com/news/articleView.html?idxno=150079']
```

07 파이썬에서 웹크롤링 진행하기

10 각 기사당 언론사를 알아보기 위해 다음과 같은 과정을 거친다. 언론사를 클릭한 후 해당 부분을 살펴본다.

11 해당 부분에 속하는 `_sp_each_source` 부분을 가져와야 한다.

```
*** ▶ <span class="_sp_each_source">...</span> == $0
    <span class="bar"></span>
    " 3시간 전 "
    <span class="bar"></span>
    <a href="https://news.naver.com/main/read.nhn?
mode=LSD&mid=sec&sid1=101&oid=011&aid=000356807
5" target="_blank" class="_sp_each_url"
onclick="return goOtherCR(this,
'a=nws*h.nav&r=1&i=88000108_0000000000000000035
68075&u='+urlencode(this.href));">네이버뉴스</a>
▶ <div class="scial" id="sp_nws1_base">...</div>
```

07 파이썬에서 웹크롤링 진행하기

12 앞서와 비슷한 방법으로 언론사를 가져온다.

```
source_text = []
source_lists = soup.select('._sp_each_source')
for source_list in source_lists:
    source_text.append(source_list.text)
```

source_text

```
['서울경제언론사 선정',
'오피니언뉴스',
'파이낸셜뉴스언론사 선정',
'이데일리',
'조선비즈언론사 선정',
'아시아경제',
'베이비타임즈',
'일요신문',
'내외경제tv',
'서울와이어']
```

08 동적 크롤링이란?

동적 크롤링

- 1 브라우저 자동화 툴
- 2 동적 크롤링: 메모리에 로딩하여 크롤링
정적 크롤링: 메모리에 로딩하지 않고 데이터를 크롤링

09 파이썬에서의 동적 크롤링

Selenium

대부분 파이썬에서의 동적 크롤링에서 본 모듈을 많이 사용한다.
Selenium은 주로 웹앱을 테스트하는데 이용하는 프레임워크다.

09 파이썬에서의 동적 크롤링

Selenium

- webdriver라는 API를 통해 운영체제에 설치된 Chrome등의 브라우저를 제어하게 된다.
- 브라우저를 직접 동작시킨다는 것은 JavaScript를 이용해 비동기적으로 혹은 뒤늦게 불러와지는 컨텐츠들을 가져올 수 있다는 것이다. 즉, '눈에 보이는' 컨텐츠라면 모두 가져올 수 있다는 뜻이다.
- 쉽게 말하면 정적 크롤링 같은 경우에는 url에 존재하는 것만 가져올수 있지만 동적 크롤링은 사람이 실행하듯이 검색을 하거나 링크를 누르거나와 같은 행위를 코드로 진행할수도 있다는 것이다.

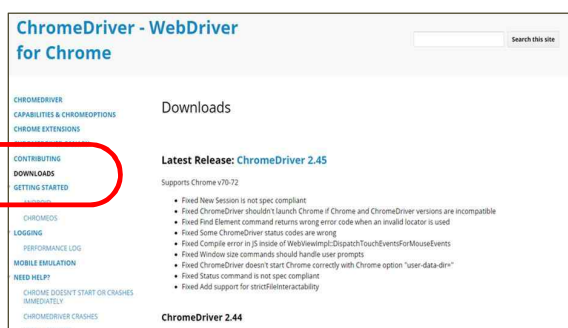
09 파이썬에서의 동적 크롤링

Selenium 설치

Selenium 웹드라이버 다운로드

```
>pip install selenium
```

<https://sites.google.com/a/chromium.org/chromedriver/downloads>



이미지 출처 : <https://sites.google.com/a/chromium.org/chromedriver/downloads>

09 파이썬에서의 동적 크롤링

Selenium 설치

Selenium을 다운로드 받았으면 다음과 같이 코드를 실행한다.

```
from selenium import webdriver  
  
driver = webdriver.Chrome('자기자신의 크롬드라이버 경로')  
driver.implicitly_wait(3)
```

기본적인 실행을 위해 다음과 같은 코드를 작성한다.
제대로 설치가 되었다면 구글이 자동적으로 열린다.

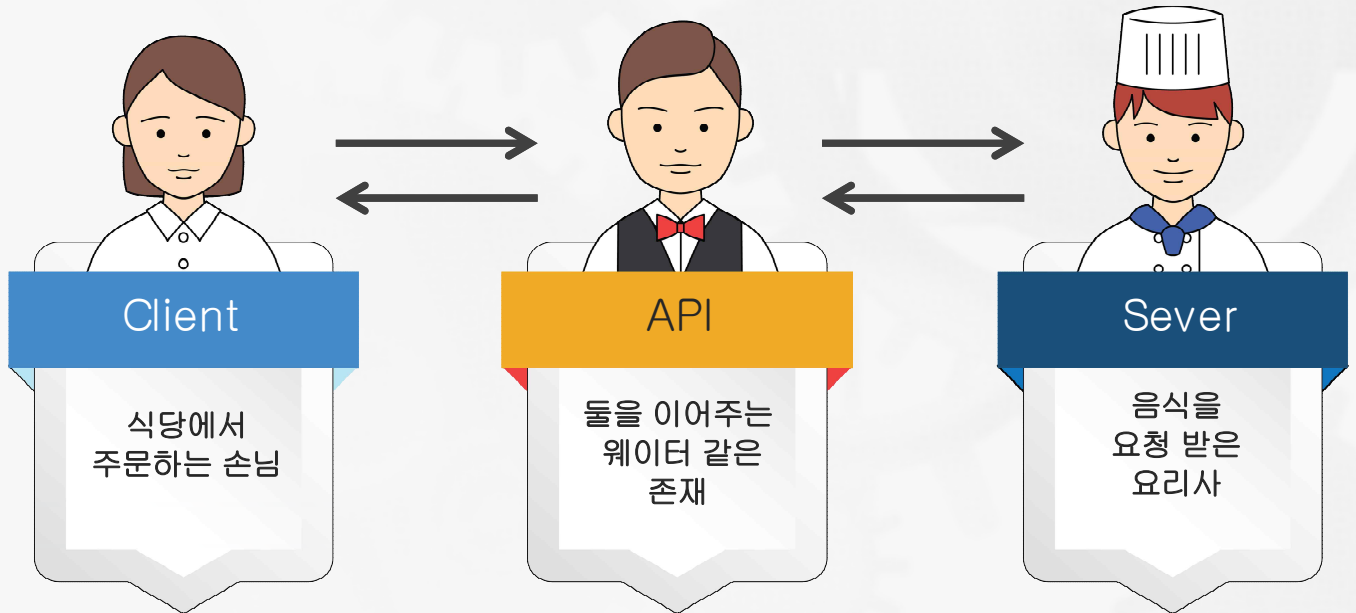
```
from selenium import webdriver  
  
driver = webdriver.Chrome('자기자신의 크롬드라이버 경로')  
driver.implicitly_wait(3)  
# url에 접근한다.  
driver.get('https://google.com')
```

10 API

• API (Application Programming Interface) •

- 응용 프로그램에서 사용할 수 있도록, 운영체제나 프로그래밍 언어가 제공하는 기능을 제어할 수 있게 만든 인터페이스
- 주로 파일제어, 창 제어, 화상처리, 문자 제어 등을 위한 인터페이스를 제공함

10 API



API 대표 서비스 5곳

- ▶ 네이버 개발자 센터(<https://developers.naver.com/main/>)
- ▶ Kakao Developers(<https://developers.kakao.com/>)
- ▶ Google Developers(<https://developers.google.com/?hl=ko>)
- ▶ 공공데이터포털(<https://www.data.go.kr/>)
- ▶ OpenWeatherMap(<https://openweathermap.org/>)

03

데이터 수집 실습



(1) 편의점 통계 데이터 분석 – 파이썬

▶ pandas 패키지 – 자동 데이터 포맷 처리

- CSV
- Excel
- HDF5
- SQL
- JSON
- Msgpack
- HTML (Table 태그 → 데이터 프레임 변환)
- Google BigQuery
- Stata
- SAS
- Clipboard

참고 데이터 convenient_store.csv

(1) 편의점 통계 데이터 분석 – 파이썬

▶ csv read 함수 통한 데이터 변수 입력

In [2]: `import pandas as pd`In [3]: `data=pd.read_csv('convenient_store.csv')`In [4]: `data.head(8)`

Out [4]:

	area	company	hourly_wage	area1	area2	outlier	name
0	강남구 삼성동	gs25 오크우드점	5600	강남구	삼성동	0	gs25
1	강남구 삼성동	gs25 코엑스점	5700	강남구	삼성동	0	gs25
2	강서구 화곡동	gs25편의점	5600	강서구	화곡동	0	gs25
3	광진구 군자동	gs25	5580	광진구	군자동	0	gs25
4	광진구 중곡동	gs25중곡대원점	5580	광진구	중곡동	0	gs25
5	구로구 구로동	gs25구로동양점	6000	구로구	구로동	0	gs25
6	구로구 구로동	gs25구로동양점	5580	구로구	구로동	0	gs25
7	동대문구 장안동	gs25장안중앙점	5600	동대문구	장안동	0	gs25

(1) 편의점 통계 데이터 분석 – 파이썬

▶ 데이터 기본 정보 확인

- dataframe 형태 저장 및 레코드수(177)/컬럼(7) 타입 확인

```
In [5]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 177 entries, 0 to 176
Data columns (total 7 columns):
area          177 non-null object
company       177 non-null object
hourly_wage   177 non-null int64
area1         177 non-null object
area2         177 non-null object
outlier       177 non-null int64
name          177 non-null object
dtypes: int64(2), object(5)
memory usage: 9.8+ KB
```

(1) 편의점 통계 데이터 분석 – 파이썬

▶ 데이터 특성 확인

- 전체 int 형태 구성 컬럼 데이터들의 빈도, 평균, 편차, 최소값, 25/50/70%, 최대값 출력

```
In [6]: data.describe()
```

```
Out[6]:
```

	hourly_wage	outlier
count	177.000000	177.0
mean	5787.627119	0.0
std	352.318646	0.0
min	5580.000000	0.0
25%	5580.000000	0.0
50%	5600.000000	0.0
75%	6000.000000	0.0
max	7500.000000	0.0

(1) 편의점 통계 데이터 분석 – 파이썬

◆ 특정 컬럼 명시시 Object에 대한 통계 확인 가능

In [7]: data.area.describe()

```
Out[7]: count      177
unique     117
top      강남구 논현동
freq         7
Name: area, dtype: object
```

(1) 편의점 통계 데이터 분석 – 파이썬

◆ 시급 6천원 이상 항목 추출

```
In [8]: highwage= data[data.hourly_wage >=6000]
highwage.head()
```

```
Out[8]:
```

	area	company	hourly_wage	area1	area2	outlier	name
5	구로구 구로동	gs25구로동양점	6000	구로구	구로동	0	gs25
9	성동구 금호동4	gs25 서울숲푸르지오점	6000	성동구	금호동4	0	gs25
13	영등포구 영등포동	gs25	6300	영등포구	영등포동	0	gs25
18	강남구 논현동	세븐일레븐 논현11호점	6200	강남구	논현동	0	7/11
27	강남구 대치동	세븐일레븐 대치3호점	6000	강남구	대치동	0	7/11

(1) 편의점 통계 데이터 분석 – 파이썬

◆ 시급 6천원 이상 & 마포구 지역 공통 항목 추출

```
In [9]: highwage= data[(data.hourly_wage >=6000) & (data.area1=='마포구')]
        highwage.head()
```

```
Out[9]:
```

	area	company	hourly_wage	area1	area2	outlier	name
66	마포구 성산2동	세븐일레븐 마포 성산대로점	6000	마포구	성산2동	0	7/11
68	마포구 서교동	세븐일레븐 홍대클럽점	6000	마포구	서교동	0	7/11
72	마포구 동교동	세븐일레븐 마포홍익점	6500	마포구	동교동	0	7/11

(1) 편의점 통계 데이터 분석 – 파이썬

◆ 시급 6천원 이상 & 마포구 지역 & '세븐일레븐' 공통 항목 추출

```
In [11]: highwage= data[(data.hourly_wage >=6000) & (data.area1=='마포구') & (data.company.str.contains('세븐일레븐'))]
        highwage.head()
```

```
Out[11]:
```

	area	company	hourly_wage	area1	area2	outlier	name
66	마포구 성산2동	세븐일레븐 마포 성산대로점	6000	마포구	성산2동	0	7/11
68	마포구 서교동	세븐일레븐 홍대클럽점	6000	마포구	서교동	0	7/11
72	마포구 동교동	세븐일레븐 마포홍익점	6500	마포구	동교동	0	7/11

(1) 편의점 통계 데이터 분석 - 파이썬

◆ 시급 6천원 초과인 경우 신규 컬럼 생성(True / False) 분류

```
In [14]: data['higher_wage'] = data.hourly_wage>6000
data.head()
```

```
Out[14]:
```

	area	company	hourly_wage	area1	area2	outlier	name	higher_wage
0	강남구 삼성동	gs25 오크우드점	5600	강남구	삼성동	0	gs25	False
1	강남구 삼성동	gs25 코엑스점	5700	강남구	삼성동	0	gs25	False
2	강서구 화곡동	gs25편의점	5600	강서구	화곡동	0	gs25	False
3	광진구 군자동	gs25	5580	광진구	군자동	0	gs25	False
4	광진구 중곡동	gs25중곡대원점	5580	광진구	중곡동	0	gs25	False

(1) 편의점 통계 데이터 분석 - 파이썬

◆ 시급 6천원 초과인 경우 True 인 경우 데이터 특성 확인

```
In [17]: data[data.higher_wage==True].describe()
```

```
Out[17]:
```

	hourly_wage	outlier
count	25.000000	25.0
mean	6518.000000	0.0
std	343.923441	0.0
min	6100.000000	0.0
25%	6300.000000	0.0
50%	6500.000000	0.0
75%	6500.000000	0.0
max	7500.000000	0.0

(1) 편의점 통계 데이터 분석 – 파이썬

◆ 시급 6천원 초과인 경우 구 정보, 시급 기준 내림차순으로 정렬 처리

```
In [19]: data_re = data[data.higher_wage == True][['area1', 'hourly_wage']]
data_re = data_re.sort_values(by='hourly_wage', ascending=False)
data_re
```

```
Out[19]:
```

	area1	hourly_wage
173	용산구	7500
29	강남구	7500
160	서대문구	6900
135	중구	6690
37	강서구	6600
172	용산구	6600
30	강남구	6500
53	구로구	6500
60	도봉구	6500
161	서대문구	6500
72	마포구	6500

(1) 편의점 통계 데이터 분석 – 파이썬

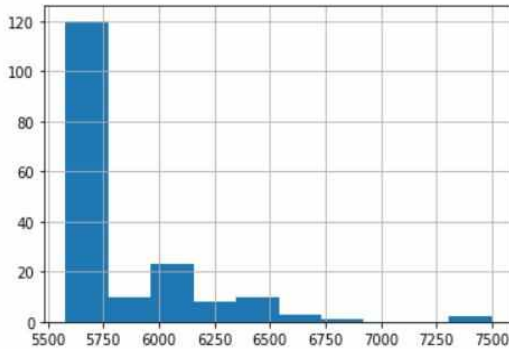
◆ csv로 결과저장 처리: data_re.to_csv('data.csv', index=False')

A	B	C	D	E
	area1	hourly_wage		
173	용산구	7500		
29	강남구	7500		
160	서대문구	6900		
135	중구	6690		
37	강서구	6600		
172	용산구	6600		
30	강남구	6500		
53	구로구	6500		
60	도봉구	6500		
161	서대문구	6500		
72	마포구	6500		
89	성동구	6500		
145	강남구	6500		
137	중구	6500		
121	종로구	6480		
128	종로구	6480		
133	중구	6300		
151	금천구	6300		
152	금천구	6300		
61	동대문구	6300		
13	영등포구	6300		
18	강남구	6200		
107	영등포구	6200		
56	구로구	6200		
143	강남구	6100		

(1) 편의점 통계 데이터 분석 - 파이썬

◆ 시급데이터 기준 10구간으로 나눈 히스토그램 시각화

```
In [27]: data.hourly_wage.hist(bins=10)
         mp.pyplot.show()
```



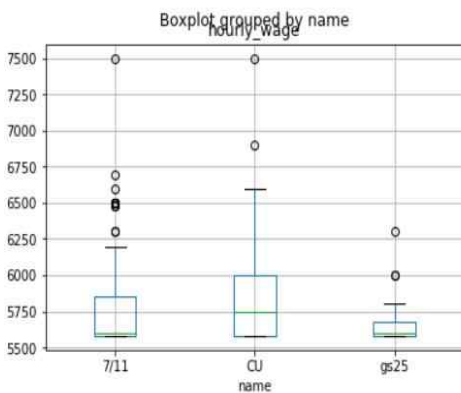
(1) 편의점 통계 데이터 분석 - 파이썬

◆ 시급과 회사별 매핑 기준 그룹핑

- 한글인 경우 matplotlib.rc로 폰트 지정 가능

```
In [28]: data.boxplot(column='hourly_wage', by='name')
```

```
Out[28]: <matplotlib.axes._subplots.AxesSubplot at 0x12e0fb35d68>
```



```
import matplotlib
font_name = matplotlib.font_manager.FontProp
erties(fname='C:/windows/Fonts/malgun.ttf').
get_name()
matplotlib.rc('font', family=font_name)
print(font_name)
```

(1) RDBMS 데이터 수집 사전 요건

◆ SQLAlchemy: 파이썬용 ORM(Object-Relational Mapping) 구현 라이브러리

◆ Pandas: 데이터 처리와 분석을 위한 파이썬 라이브러리

```
pip install sqlalchemy
pip install pandas
```

◆ 샘플 CSV

1	Eldon Bas	Muhamm	3	-213.25	38.94	35	Nunavut	Storage &	0.8
2	1.7 Cubic I	Barry Fren	293	457.81	208.16	68.02	Nunavut	Appliance	0.58
3	Cardinal S	Barry Fren	293	46.71	8.69	2.99	Nunavut	Binders ar	0.39
4	R380	Clay Rozer	483	1198.97	195.99	3.99	Nunavut	Telephone	0.58
5	Holmes HI	Carlos Soli	515	30.94	21.78	5.94	Nunavut	Appliance	0.5
6	G.E. Longe	Carlos Soli	515	4.43	6.64	4.95	Nunavut	Office Furi	0.37
7	Angle-D Bi	Carl Jackso	613	-54.04	7.3	7.72	Nunavut	Binders ar	0.38
8	SAFCO Mc	Carl Jackso	613	127.7	42.76	6.22	Nunavut	Storage & Organization	
9	SAFCO Coi	Monica Fe	643	-695.26	138.14	35	Nunavut	Storage & Organization	
10	Xerox 198	Dorothy B	678	-226.36	4.98	8.33	Nunavut	Paper	0.38

(2) RDBMS 데이터 수집 테이블 읽기

◆ SQLAlchemy: 파이썬용 ORM(Object-Relational Mapping) 구현 라이브러리

◆ Pandas: 데이터 처리와 분석을 위한 파이썬 라이브러리

```
from sqlalchemy import create_engine
import pandas
import os
data=pandas.read_csv('samplecsv.csv',encoding="ISO-8859-1")
engine=create_engine('sqlite:///memory:') #DB엔진 생성
Data.to_sql( ' data_table ' ,engine) #데이터프레임 테이블 저장
print(pandas.read_sql_query('SELECT * FROM data_table',engine))
```

(2) RDBMS 데이터 수집 테이블 읽기

#추가

```
sql.execute('INSERT INTO data_table VALUES(?,?,?,?,?,?,?,?,?,?,?)',W
            engine,params=[(9,11,'Xerox 198','Frederick
Jones',752,-321.57,2.2,9.8,'Nunavut','Paper',0.37)])
sqlalchemy.engine.result.ResultProxy object at 0x00FA3530>
print(pandas.read_sql_query('SELECT * FROM data_table',engine))
```

#삭제

```
sql.execute('DELETE FROM data_table where
35=(?)',engine,params=[(9.80)])
sqlalchemy.engine.result.ResultProxy object at 0x00FA3170>
print(sql.read_sql_query('SELECT * FROM data_table',engine))
```

(1) 파이썬에서의 동적 크롤링

◆ Selenium을 이용해 홈페이지 로그인하는 방법

① 로그인 창에 접근한다.

```
from selenium import webdriver

# webdriver 경로

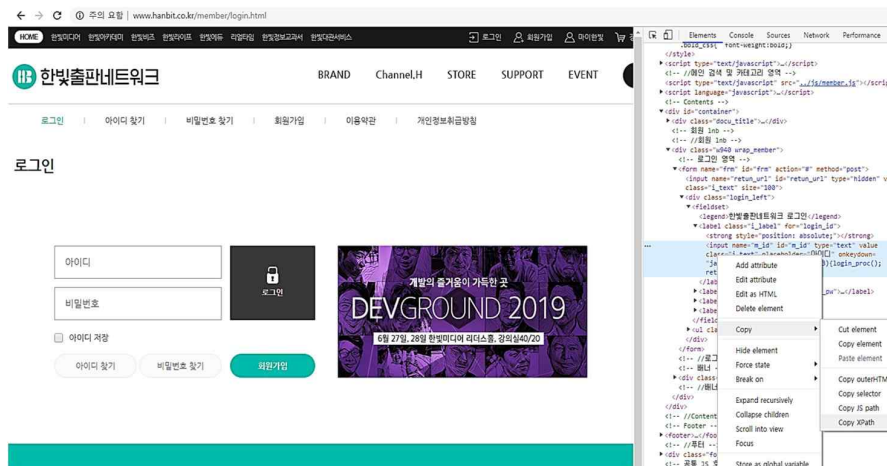
browser = webdriver.Chrome('자기자신의 크롬드라이버 경로')

browser.get('http://www.hanbit.co.kr/member/login.html')
```

(1) 파이썬에서의 동적 크롤링

◆ Selenium을 이용해 홈페이지 로그인하는 방법

② 아이디창에 입력하기 위해 xpath를 가져온다.



(1) 파이썬에서의 동적 크롤링

◆ Selenium을 이용해 홈페이지 로그인하는 방법

③ id 부분의 xpath를 확인하고 send_keys 명령어를 통해 아이디를 입력한다.

```
##찾아낸 xpath: //*[@id="m_id"]
id = browser.find_element_by_xpath('//*[@id="m_id"]')
id.send_keys('아이디')
```

(1) 파이썬에서의 동적 크롤링

◆ Selenium을 이용해 홈페이지 로그인하는 방법

④ 비밀번호를 입력하기위해 비밀번호에 해당하는 부분을 살펴본다.

```
... <input name="m_passwd" id="m_passwd" type="password"
    value class="i_text" placeholder="비밀번호"
    onkeydown="javascript:if(event.keyCode==13)
    {login proc(); return false;}"> == $0
```

(1) 파이썬에서의 동적 크롤링

◆ Selenium을 이용해 홈페이지 로그인하는 방법

⑤ 앞서 방법과 비슷한 방식으로 send_keys 함수를 이용하여 비밀번호를 입력한다.

```
id_ = browser.find_element_by('m_passwd')
id_.send_keys('비밀번호')
```

(1) 파이썬에서의 동적 크롤링

◆ Selenium을 이용해 홈페이지 로그인하는 방법

- ⑥ 로그인 부분을 클릭해야 하므로 동일한 방법으로 로그인 부분의 웹소스를 살펴본다.

```
...
<input type="button" name="login_btn" id="login_btn"
value="로그인" class="btn_login"> == $0
</label>
▶<label class="i_label2">...</label>
</fieldset>
▶<ul class="login_btn">...</ul>
</div>
</form>
<!-- //로그인 영역 -->
<!-- 배너 -->
▶<div class="login_right">...</div>
<!-- //배너 -->
</div>
</div>
<!-- //Contents -->
<!-- Footer -->
▶<footer>...</footer>
<!-- //Footer -->
```

(1) 파이썬에서의 동적 크롤링

◆ Selenium을 이용해 홈페이지 로그인하는 방법

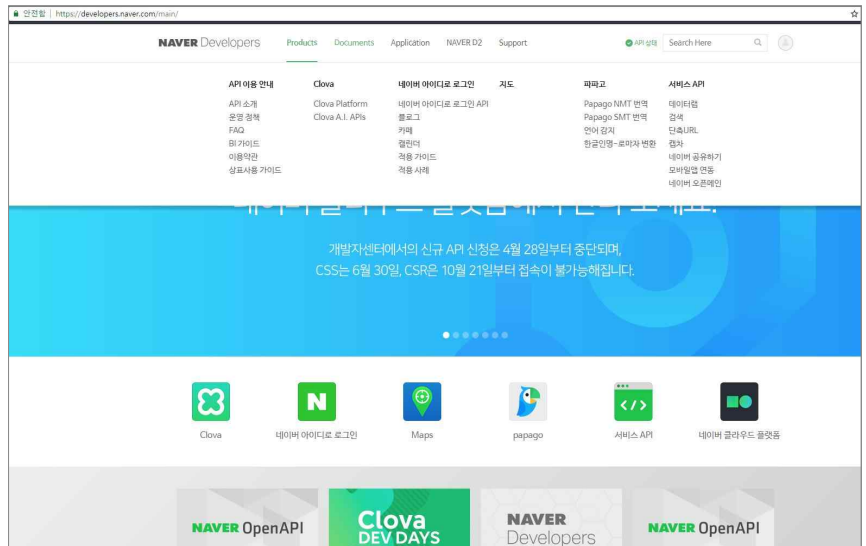
- ⑦ 웹소스의 로그인 부분에 맞추어 부분을 find_element_by_class 함수를 통해서 불러오고 click() 함수를 이용하여 로그인한다.

```
_submit=browser.find_element_by_class_name('btn_login')
_submit.click()
```

(1) 파이썬에서의 동적 크롤링

▶ Naver API 사용하기

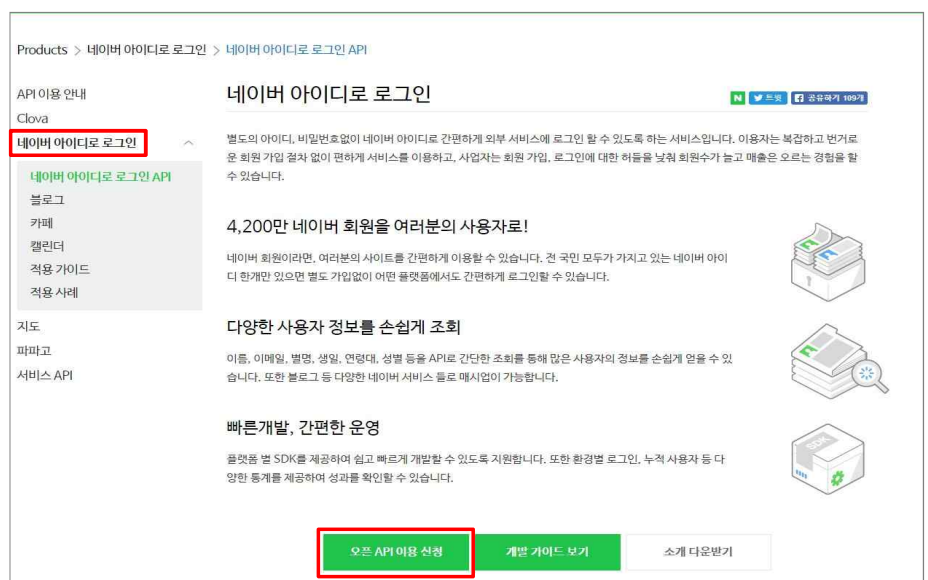
- ① 네이버 개발자 페이지로 이동하여 상단에 Products에 들어가 네이버 아이디로 로그인 API로 들어간다.



(1) 파이썬에서의 동적 크롤링

▶ Naver API 사용하기

- ② 네이버 아이디로 로그인 API로 이동 후 오픈 API 이용을 신청한다.



(1) 파이썬에서의 동적 크롤링

▶ Naver API 사용하기

- ③ 로그인 후 어플리케이션 이름과 자신이 사용할 API를 선택한다.

사용 API

- 검색
- 데이터랩(검색어트렌드)

웹 서비스 URL

- http://localhost

애플리케이션 등록 (API 이용신청)

애플리케이션의 기본 정보를 등록하면, 좌측 내 애플리케이션 메뉴의 서버 메뉴에 등록하신 애플리케이션 이름으로 서버 메뉴가 만들어집니다.

애플리케이션 이름	API_test_Andrew_01 ✓
사용 API	검색 × 데이터랩 (검색어트렌드) ×
비로그인 오픈 API 서비스 환경	웹 서비스 URL (최대 10개) http://localhost + ✓

※ 텍스트 중 우측 끝의 "+" 버튼을 누르면 행이 추가되며, "-" 버튼을 누르면 행이 삭제됩니다.
 ※ http와 https는 구분하지 않습니다.
 ※ www는 제외 입력해 주세요. 예) http://naver.com
 ※ 서버 도메인이 있으면 대표 도메인명만 입력해 주세요. (예: http://naver.com)
 ※ 하이브리드 앱은 location.href 객체 출력 값을 입력하면 됩니다. (예: file://로컬 URL)

(1) 파이썬에서의 동적 크롤링

▶ Naver API 사용하기

- ④ 등록하기를 누르고 신규 애플리케이션 등록을 확인한다.

보기를 누르면 자신의 Client Secret를 확인할 수 있다.

API_test_Andrew_01

개요	API 설정	멤버관리	로그인 통계	API 통계	Playground (Beta)
----	--------	------	--------	--------	-------------------

애플리케이션 정보

Client ID	OeuMIVBoh2yLntneSIOr
Client Secret	 보기

(1) 파이썬에서의 동적 크롤링

◆ 네이버 데이터 수집하기

- 네이버의 서비스는 ‘네이버 아이디로 로그인’이라는 서버 인증을 통해 Access Token을 가지고 온 후 API를 이용할 수 있는 서비스와 HTTP 헤더에 클라이언트 아이디와 시크릿(secret) 값만을 얹어 전송하면 되는 비로그인 서비스로 나뉨

네이버의 서비스의 장점

- 네이버의 뉴스 블로그 및 카페의 글들은 로그인을 하지 않고 비로그인 방식으로 데이터를 조회하여 가지고 올 수 있도록 설계되어 있어 복잡한 인증 과정을 거치지 않아도 됨
- 어플리케이션을 만들 경우 Access Token을 얻기 위한 웹페이지 처리를 하지 않아도 됨

(1) 파이썬에서의 동적 크롤링

◆ 네이버 데이터 수집하기

- 네이버의 서비스의 제약

① 하루 검색 기사수

- ✓ API 호출이 하루당 25000회로 제한되어 있음
- ✓ 최대 한번 호출에 100개의 검색을 가지고 올 수 있으므로 250000건의 데이터 호출이 가능함

② 검색 포인터 문제

- ✓ 실제 네이버에서 이뤄지는 데이터 검색은 최대 1000개만 API를 이용하여 가지고 오는 것이 가능함

(1) 파이썬에서의 동적 크롤링

◆ 네이버 데이터 예시

- 발급받은 아이디와 비밀번호를 지정하고 자신이 검색하고 싶은 키워드를 입력함
- 네이버 책부분에서 JSON 파일로 받을지를 URL을 통하여 결정하고 몇개를 보여줄지 (display) 그리고 어디서부터 보여줄지(start)를 통하여 결정함

```
In [1]: import urllib.request
client_id = 'jZpHFpZAPIR09Dca0jfv' # 애플리케이션 등록시 발급 받은 값 입력
client_secret = "skinm3wul8"
encText = urllib.parse.quote("여행")
url = "https://openapi.naver.com/v1/search/book.json?query=" + encText + "&display=3&start=10&sort=count"
request = urllib.request.Request(url)
request.add_header("X-Naver-Client-Id", client_id)
request.add_header("X-Naver-Client-Secret", client_secret)
response = urllib.request.urlopen(request)
rescode = response.getcode()
if(rescode==200):
    response_body = response.read()
    print(response_body.decode('utf-8'))
else:
    print("Error Code:" + rescode)
```

(1) 파이썬에서의 동적 크롤링

◆ 네이버 데이터 크롤링 예시

- 만들어진 파일을 json 형식으로 저장하는 코드임

```
In [2]: import json
jsonString = json.dumps(response_body.decode('utf-8'))
response_body.decode('utf-8')
jsonString = json.dumps(response_body.decode('utf-8'), indent=4)
tryjson=json.loads(jsonString)
with open('test.json', 'w', encoding="utf-8") as make_file:
    json.dump(tryjson, make_file, ensure_ascii=False, indent="  ")
with open('test.json') as f:
    data = json.load(f)

    print(type(data))

print(data)
```

(1) 파이썬에서의 동적 크롤링

◆ 네이버 데이터 크롤링 예시

• 파이썬에 나오는 크롤링 결과임

```

<class 'str'>
{
  "lastBuildDate": "Fri, 15 Jun 2018 11:55:19 +0900",
  "total": 400807,
  "start": 10,
  "display": 3,
  "items": [
    {
      "title": "일본 소도시 <b>여행</b> (숨은 보석처럼 반짝이는 풍경, 서일본의 소도시를 거닐다)",
      "link": "http://book.naver.com/bookdb/book_detail.php?bid=12841984",
      "image": "http://bookthumb.phinf.naver.net/cover/128/419/12841984.jpg?type=m1&date=20180220",
      "author": "박탄홍",
      "price": "16000",
      "discount": "14400",
      "publisher": "박서스BOOKS",
      "pubdate": "20171210",
      "isbn": "1161651896 9791161651897",
      "description": "지친 <b>여행</b>자를 반겨 주는 소박한 온천탕과 우동 한 그릇, &#x0D;대도시를 떠날 때 &#x0D;숨은 보석처럼 반짝이는 서일본의 33개 소도시, &#x0D;그 소박하고 여유로운 풍경 속으로 잘 알려지지 않은, 그러나 자세히 들여다보면 너무도..."
    },
    {
      "title": "자전거<b>여행</b> 1",
      "link": "http://book.naver.com/bookdb/book_detail.php?bid=6275708",
      "image": "http://bookthumb.phinf.naver.net/cover/082/757/08275708.jpg?type=m1&date=20150718",
      "author": "김훈",
      "price": "15000",
      "discount": "13500",
      "publisher": "문학동네",
      "pubdate": "20141022",
      "isbn": "8954626203 9788954626200",
      "description": "김훈의 두 번째 자전거로 떠난 <b>여행</b>의 정수김훈 산문의 정수가 담긴 『자전거 칼럼』은 언저리에서 태어나는... 외견과 정서의 세계를 멀리하고 물리적 사실을 객관적으로 진술하려는 그의 언어로 표현해낸 자전거 <b>여행</b>길들..."
    }
  ]
}

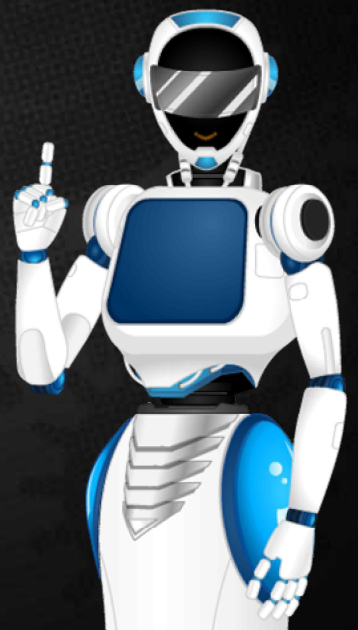
```

Artificial Intelligence (AI) refers to the simulation of human

SUMMARY

학습정리

- ▣ 정형데이터와 비정형데이터의 데이터 수집 개념과 기술
- ▣ 데이터 수집 실습



quiz

정리퀴즈

1. 정형 데이터 분석과 관련이 있는 것은?

- ① 워드 클라우드 ② SQL 분석
- ③ 감성 분석 ④ 군집 분석

정답 및 해설은 강의에서 확인 할 수 있습니다.

quiz

정리퀴즈

2. 텍스트 분석을 위한 텍스트 데이터 수집 방법으로 맞지 않는 것은?

- ① API ✓ ② Log Aggregator
- ③ XML ④ Crawling

정답 및 해설은 강의에서 확인 할 수 있습니다.

참고 문헌

REFERENCE

The book is not only a reference
for any machine that wishes
to be associated with a human
world much as learning and
problem-solving.

- 빅데이터 분석과 활용 :
4차 산업혁명 현장 전문가가 알려주는
by 박인근, 홍지후, 강남규, 김성호, 정구범, 제이펍 ,2019
- 데이터 분석 전문가 가이드 (ADP, ADsP)
by 한국데이터진흥원, 한국데이터진흥원, 2019

☞ 서체 출처 : 에스코어드림체-(주)에스코어, 나눔글꼴체-(주)네이버