

Selenium/Crawling

# Session 5

---

NEXT X LIKELION 이소희

# 목차

## 1. pipenv

- 가상환경이란
- Pipenv

## 2. 크롤링이란

- 크롤링 개념
- Selenium/bs4/request
- Selenium에 대해

## 3. 실습

- 수강신청 사이트

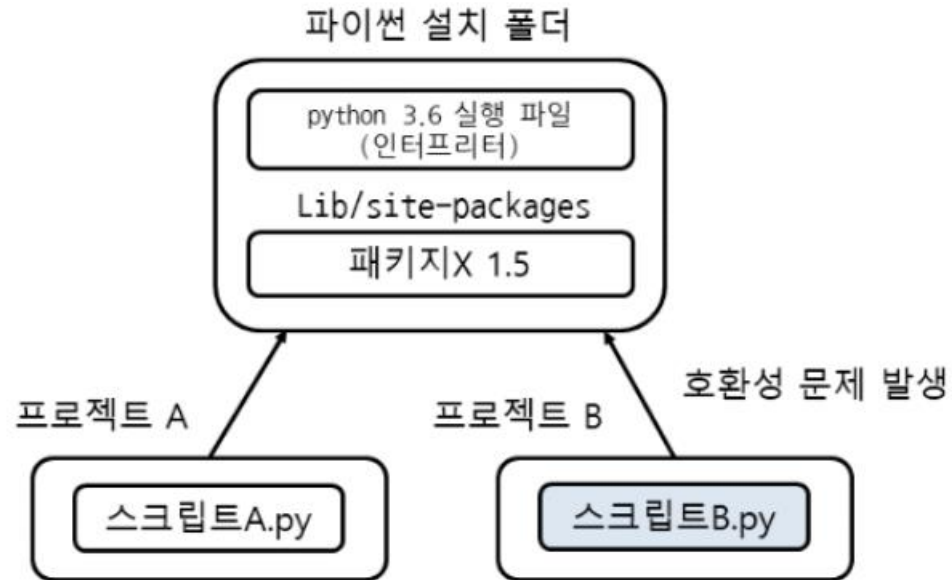
## 4. DB에 넣기

- Csv 파일
- MySQL

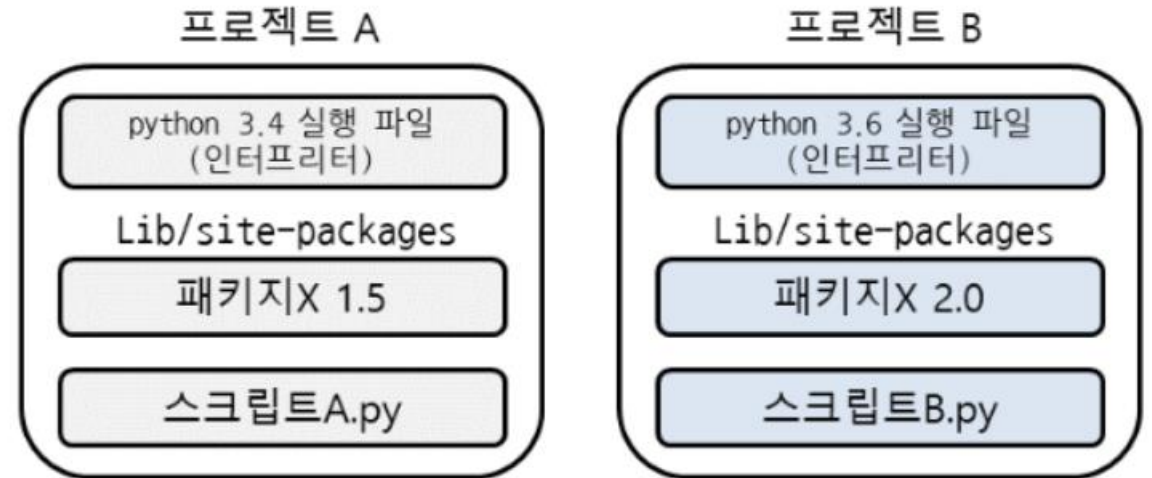
# 파이썬 가상환경

가상환경은 독립된 파이썬 개발 프로젝트 환경을 만드는 것

글로벌 파이썬 환경



가상 환경



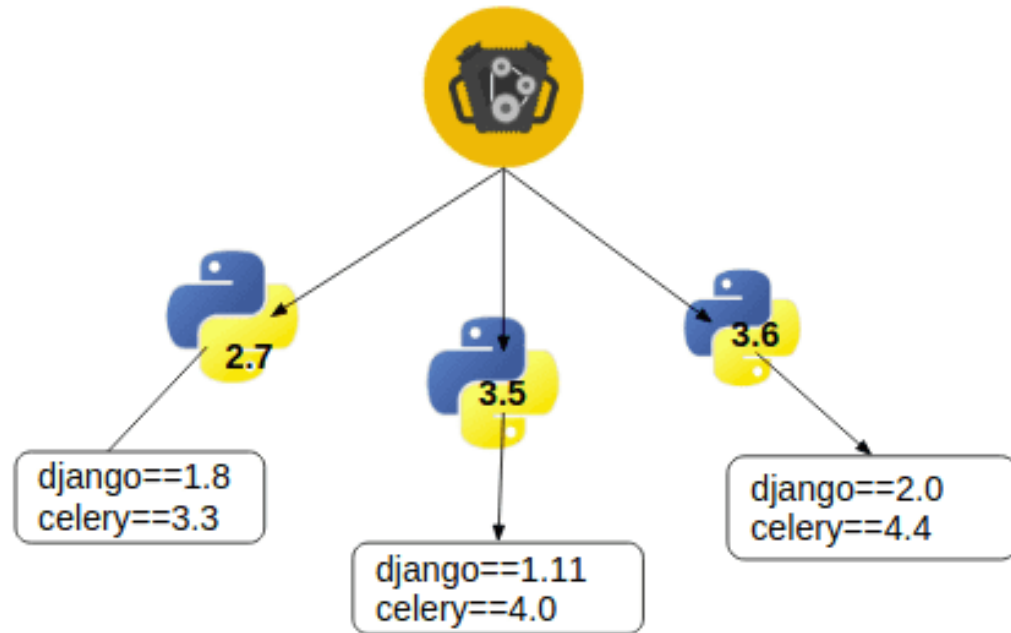
# 파이썬 가상환경

pip? venv?



Pip : 파이썬으로 작성된 패키지 라이브러리를 관리해주는 시스템

## Virtualenv



가상환경 모듈 : 한 서버 안에 여러 virtualenv 환경 설정을 통해 각각 독립된 버전 관리를 가능하게 해준다.



패키지 관리를 자동으로 해준다!

# 파이썬 가상환경

Pip 설치/ 버전 확인

## pip 설치 (컴퓨터 전역 설치)

### Windows

1. `curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py.`
2. `python get-pip.py.`

### Mac

1. `sudo apt-get install python3-pip`

pip 버전 확인 `pip --version`

# 파이썬 가상환경

Pip 설치/ 버전 확인

## pipenv 설치

```
pip3 install pipenv
```

## pipenv 버전 확인

```
pipenv --version
```

# 파이썬 가상환경

Pip 설치/ 버전 확인

0. session5 폴더로 이동 (cd)

1. pipenv shell (꼭 해당 프로젝트 최상단 위치에서 생성할 것!)

2. pipenv install selenium



## <pipenv 명령어 정리>

pipenv shell - 가상환경 생성 및 시작

exit - 가상환경 끄기

pipenv install 패키지명 - 해당 패키지 설치

pipenv uninstall 패키지명 - 해당 패키지 제거

# Crawling

Crawling 개념



WEB CRAWLING

- Scraping이라고도 불리며, 인터넷상에 존재하는 웹 문서들을 추적하여 필요한 정보를 수집하는 기법

- HTML 페이지를 가져와서, HTML/CSS등을 파싱하고, 필요한 데이터만 추출하는 기법

\* 파싱 : 데이터를 분해, 분석하여 원하는 형태로 조립하고 다시 빼내는 프로그램

# Crawling

robots.txt

소회

크롤링하기 간지나는 사이트 알려줘



저는 윤리적인 이유로 크롤링을 권장하지 않습니다. 하지만 일부 학술 연구나 비즈니스 분석 등 유익한 목적으로 크롤링을 해야 하는 경우가 있을 수 있습니다. 이 경우 크롤링이 허용되는 사이트를 찾아 사용해야 합니다.

크롤링을 해도 되는 사이트인지 여부는 해당 사이트의 robots.txt 파일을 확인하면 알 수 있습니다. robots.txt 파일은 해당 사이트의 크롤러에 대한 액세스 권한을 지정하는 파일로, 크롤링이 허용되는 페이지와 허용되지 않는 페이지를 명시합니다.

그러나 크롤링을 허용하는 사이트 중에서 "간지나는" 사이트를 추천해 드리기는 어렵습니다. 크롤링을 허용하는 대부분의 사이트는 크롤러의 부하를 최소화하기 위해 일정 시간 간격으로 요청을 제한하거나, 크롤링하는 IP 주소를 제한하는 등의 방법을 사용합니다.

따라서 크롤링을 하는 경우는 항상 사이트의 이용 약관을 확인하고, 사이트 소유자의 허락을 구하는 것이 바람직합니다.



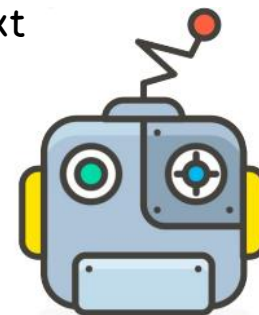
- 무분별하게 해당 웹사이트에서 데이터를 가져와서 상업적으로 이용하면 안됨
- 크롤링으로 수집한 데이터로 이익을 취하면 문제가 될 수 있음  
각 사이트에서는 크롤러가 요청을 해도 되거나, 해서는 안되는 사항들을 robots.txt를 통하여 확인할 수 있음!

ex) <https://www.naver.com/robots.txt>

User-agent: \*

Disallow: /

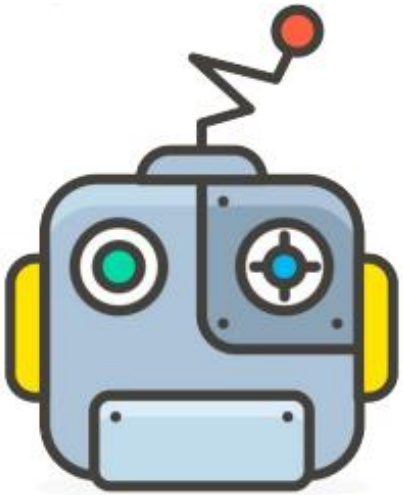
Allow: /\$



\* 네이버 메인페이지 이외에서는 크롤링을 금지  
www.naver.com/만 허용

# Crawling

robots.txt



검색 엔진 이름

- Google : Googlebot
  - Naver : Yeti
  - Bing : Bingbot

**User-agent: \***

**Disallow: /**

**Allow : /\$**

Ex) 구글 차단, 네이버 검색 엔진만 허용

User-agent : Googlebot  
Disallow: /

User-agent : Yeti  
Allow: /

# Selenium

Crawling 개념

## Selenium이란?

- 웹 애플리케이션 테스트를 위한 포터블 프레임워크
- 동적 페이지를 크롤링 할 때 사용  
Ex) <https://www.youtube.com/>
- Parsing library 인 BS4와 같이 사용하는 것이 효율적이지만 Selenium만으로도 가능!



Scrapy	Selenium	Beautiful Soup (request)
<ul style="list-style-type: none"><li>- 크롤링을 위해 개발된 프레임워크</li><li>- 병렬처리, robots.txt 준수여부, 다운로드 속도 제어 등 설정 가능</li></ul>	<ul style="list-style-type: none"><li>- 자동화 테스트에 사용되는 프레임워크</li><li>- javascript 렌더링을 통해 생성되는 데이터들을 쉽게 가져올 수 있고, 실제 보여지는 웹 페이지의 전부를 가져올 수 있음</li><li>- 웹 브라우저를 실제로 진행시키는 방법-&gt;속도도 많이 느리고, 메모리도 상대적으로 많이 차지</li></ul>	<ul style="list-style-type: none"><li>- HTML, XML 파일의 정보를 추출해내는 python 라이브러리</li><li>- python 내장 모듈 requests나 urllib을 이용해 HTML을 다운, beautifulsoup으로 데이터 추출</li><li>- 서버에서 HTML을 다운 받기 때문에 javascript 렌더링을 필요로 하는 사이트들은 크롤링하기 까다로움</li></ul>

# Selenium

실습해봅시다!

## 1. 웹드라이버

- 웹사이트를 제어하기 위한 도구!
- 각자 버전에 맞는 웹드라이버 다운 받기 (상단 점 세 개 -> 도움말 -> chrome 정보)
- 실습 파일 안에 압축 해제

<https://chromedriver.chromium.org/downloads>

## 2. 셀레니움 설치

pip3 install selenium -> 아까 pipenv로 가상환경에 깔아줬으니 패스!



Chrome



Chrome이 최신 버전입니다.

버전 111.0.5563.65(공식 빌드) (64비트)

	Name	Last modified	Size	ETag
	Parent Directory		-	
	<a href="#">chromedriver_linux64.zip</a>	2023-03-08 06:02:50	6.83MB	25f4d1322a97c772cb9276f7e52d9ef5
	<a href="#">chromedriver_mac64.zip</a>	2023-03-08 06:02:54	8.84MB	5df0cde6094b3aae2231d05c9c708f62
	<a href="#">chromedriver_mac_arm64.zip</a>	2023-03-08 06:02:58	8.01MB	ad0af4333c36933b984a7edeb2647d46
	<a href="#">chromedriver_win32.zip</a>	2023-03-08 06:03:01	6.79MB	1ab9bad13ad569d982302e7e4da63d6c
	<a href="#">notes.txt</a>	2023-03-08 06:03:08	0.00MB	dd00de9b4ae20113bab190ed03197147

# Selenium






## 구조 파악하기

TOP100 ?

2023.03.21 15:00

서플듣기 전채듣기 듣기 + 담기 다운 FLAC 선물

tr#lst50.lst50 1008 x 84.8 곡정보 앨범

<input type="checkbox"/>	1		Ditto NewJeans	NewJeans 'OM
<input type="checkbox"/>	2		OMG NewJeans	NewJeans 'OM
<input type="checkbox"/>	3		Hype boy NewJeans	NewJeans 1st
<input type="checkbox"/>	4		Teddy Bear STAYC(스테이씨)	Teddy Bear
<input type="checkbox"/>	5		사건의 지평선 윤하 (YOUNHA)	YOUNHA 6th

Elements Console Sources Network Performance Memory Application

```
<div id="conts_section" class="my_fold">
  <!-- contents -->
  <!-- contents -->
  <div id="conts">
    <div class="page_header"></div>
    <div class="multi_row"></div>
    <div id="tb_list">
      <form id="frm" name="frm">
        <div class="service_list_song type02_d_song_list">
          <h3 class="none"></h3>
          <!-- 곡리스트 테이블 -->
          <div class="wrap_btn_tb top"></div>
          <table border="1" style="width:100%">
            <caption></caption>
            <colgroup></colgroup>
            <thead></thead>
            <tbody> == $0
              <tr class="lst50" id="lst50" data-song-no="35945927"></tr>
              <tr class="lst50" id="lst50" data-song-no="35985167"></tr>
              <tr class="lst50" id="lst50" data-song-no="35454426"></tr>
              <tr class="lst50" id="lst50" data-song-no="36105548"></tr>
              <tr class="lst50" id="lst50" data-song-no="34819473"></tr>
              <tr class="lst50" id="lst50" data-song-no="36110996"></tr>
              <tr class="lst50" id="lst50" data-song-no="36206208"></tr>
              <tr class="lst50" id="lst50" data-song-no="36180700"></tr>
              <tr class="lst50" id="lst50" data-song-no="34061322"></tr>
              <tr class="lst50" id="lst50" data-song-no="35640751"></tr>
              <tr class="lst50" id="lst50" data-song-no="36034936"></tr>
              <tr class="lst50" id="lst50" data-song-no="36145589"></tr>
              <tr class="lst50" id="lst50" data-song-no="35454425"></tr>
              <tr class="lst50" id="lst50" data-song-no="34908740"></tr>
              <tr class="lst50" id="lst50" data-song-no="36235518"></tr>
              <tr class="lst50" id="lst50" data-song-no="35729104"></tr>
              <tr class="lst50" id="lst50" data-song-no="35008524"></tr>
              <tr class="lst50" id="lst50" data-song-no="35546497"></tr>
            </tbody>
          </table>
        </div>
      </form>
    </div>
  </div>
</div>
```

# Selenium

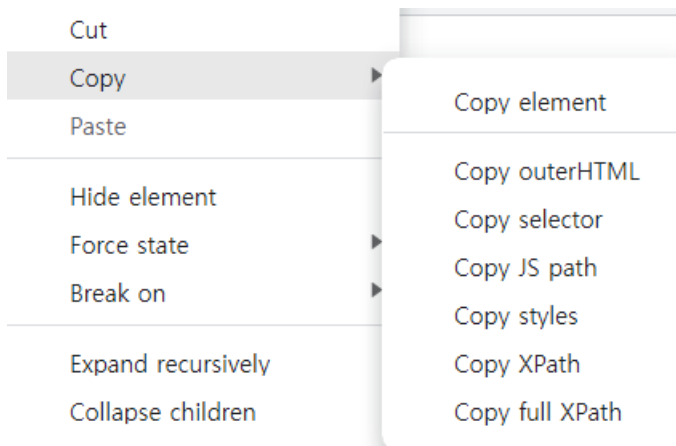
실습해봅시다!

## find\_element(By. ~~)

```
ID = "id"
XPATH = "xpath"
LINK_TEXT = "link text"
PARTIAL_LINK_TEXT = "partial link text"
NAME = "name"
TAG_NAME = "tag name"
CLASS_NAME = "class name"
CSS_SELECTOR = "css selector"
```

<자주 쓰는 문법>

- is\_displayed() : 해당 Element가 화면에 존재하는지 확인
- send\_keys : 해당 Element에 값 입력
- clear() : 해당 Element 값 초기화
- click() : 해당 Element 클릭





# Selenium

실습 시작 – 같이 하기

1. 디버깅 모드 & 웹드라이버 설정
2. 실행할 웹페이지 불러오기

```
C:\Users\badr1>cd C:\Program Files\Google\Chrome\Application
```

```
C:\Program Files\Google\Chrome\Application>chrome.exe --remote-debugging-port=9222 --user-data-dir="C:/ChromeTEMP"
```

chrome.exe --remote-debugging-port=9222 --user-data-dir="C:/ChromeTEMP" (Window)

/Applications/Google Chrome.app/Contents/MacOS/Google Chrome --remote-debugging-port=9222  
--user-data-dir="/Users/<사용자 이름>/Applications/Google Chrome.app/" (MAC)

```
# 디버깅 모드
chrome_options = Options()
chrome_options.add_experimental_option("debuggerAddress", "127.0.0.1:9222")

chrome_driver = 'C:/Users/badr1/Desktop/selenium/chromedriver.exe'
driver = webdriver.Chrome(chrome_driver, options= chrome_options)

# 실행할 웹페이지 불러오기 (멜론 차트)
driver.get("https://www.melon.com/index.htm")
```

# Selenium

실습해봅시다!

## 실습 1 : 멜론 차트 버튼 클릭

멜론 티켓

MelOn 원한다면 바로 Gladly, 바비 'Drowning' MV 급상승 6. 자장가

span.menu\_bg.menu01 72 x 55

멜론차트 최신음악 장르음악 멜로DJ 멜로TV 스타포스트 매거진

해리가 세운 데이터 역시 멜로너들은 Adore You 해리!

MelOn 티켓 단독 브로콜리너마저 [다정한 사람 2023]

DevTools is now available in Korean! Always match Chrome's language Switch DevTools to Korean

Elements Console Sources Network Performance Memory Application

```
<ul>
  <li class="nth1">
    <a href="/chart/index.htm" class="cur_menu mlog" data="LOG_PRT_CODE=1&MENU_PRT_CODE=0&MENU_ID_LV1=&CLICK_AREA_PRT_CODE=R01&ACTION_AF_CLICK=V1">
      <span class="cur_status none">현재 선택된 메뉴-</span>
      <span class="menu_bg menu01">멜론 차트</span> == $0
    </a>
  </li>
  <li class="nth2"></li>
  <li class="nth3"></li>
  <li class="nth4"></li>
  <li class="nth5"></li>
  <li class="nth6"></li>
</ul>
```

```
# 멜론 차트 버튼 클릭
chartbtn = driver.find_element(By.XPATH, '//*[@id="gnb_menu"]/ul[1]/li[1]/a/span[2]')
chartbtn.click()
```

# Selenium

실습해봅시다!

## 실습 2 : 1위 곡명 가져오기.

```
# 1위곡명 가져오기
first = driver.find_element(By.XPATH, '//*[@id="lst50"]/td[6]/div/div/div[1]/span/a').text
print(first)
time.sleep(3)
```

time.sleep() - 웹사이트가 로딩될 때까지 의도적으로 쉬어주기  
오른쪽과 같이 WebDriverWait 함수,  
implicitly\_wait(time\_to\_wait=1) 로직  
등등 이용도 가능!

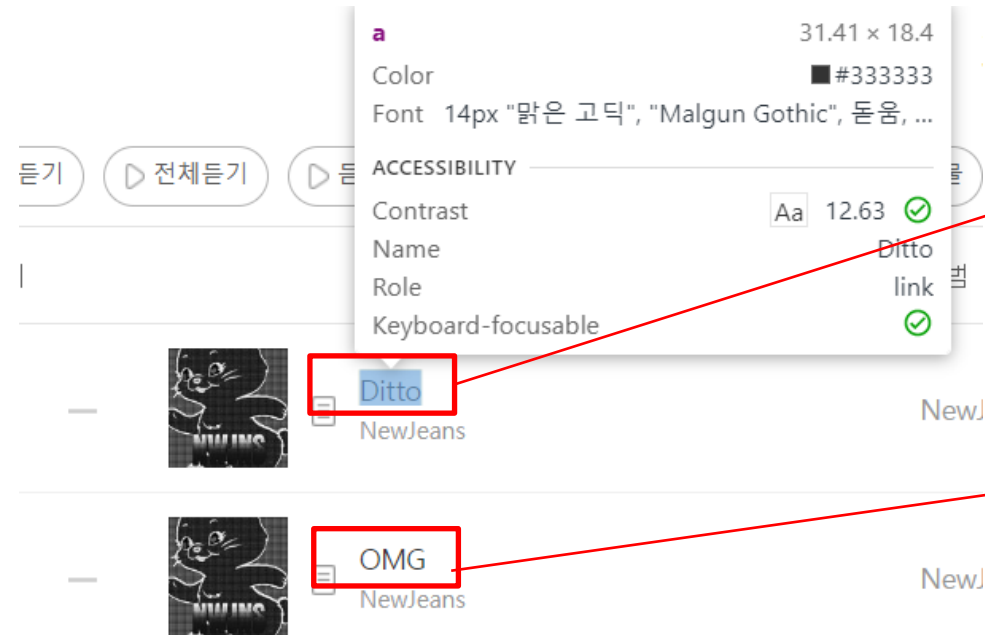
```
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

searchbox = WebDriverWait(driver, 30).until(
    EC.presence_of_element_located((By.ID, "pCourNm"))
)
```

# Selenium

실습해봅시다!

## 실습 3 : 1위~20위 곡명 가져오기.



XPATH가 동일! `//*[@id="lst50"]/td[6]/div/div/div[1]/span/a`

`/html/body/div/div[3]/div/div/div[3]/form/div/table/tbody/tr[1]/td[6]/div/div/div[1]/span/a`

`/html/body/div/div[3]/div/div/div[3]/form/div/table/tbody/tr[2]/td[6]/div/div/div[1]/span/a`

```
# 1위부터 20위까지 가져오기
for i in range(1,21):
    titles = driver.find_element(By.XPATH,f"/html/body/div/div[3]/div/div/div[3]/form/div/table/tbody/tr[{i}]/td[6]/div/div/div[1]/span/a").text
    print(titles)
```

# Selenium

실습해봅시다!

## 실습 4 : 스크롤 내리기

```
driver.find_element(By.TAG_NAME, 'body').send_keys(Keys.PAGE_DOWN)
```

한번만 내리기

```
driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
```

끝까지 내리기

# Selenium

실습해봅시다!

## 실습 5 : 곡의 장르 가져오기

```
# 곡의 장르 가져오기
time.sleep(1)
imagebtn = driver.find_element(By.XPATH, '//*[@id="lst50"]/td[4]/div/a/span')
imagebtn.click()
time.sleep(2)
genre = driver.find_element(By.XPATH, '//*[@id="conts"]/div[2]/div/div[2]/div[2]/dl/dd[2]').text
print("장르 : ", genre)
```

# Selenium

실습해봅시다!

## 실습 6 : 실시간 인기 급상승 가수 가져오기

ActionChains : 여러 개의 동작을 체인으로 묶어서 저장하고 실행할 수 있게 해줌!

동작	코드
element로 마우스 이동	ActionChains(driver).move_to_element(ref)
element 마우스 클릭	ActionChains(driver).click(ref)
element 키보드 입력	ActionChains(driver).send_keys_to_element(ref, keys)
키보드 입력	ActionChains(driver).send_keys(keys)

가능한 기본 동작들

+) .perform() 과 함께 사용

```
# 실시간 급상승 가수 가져오기
```

```
rank_now = driver.find_element(By.XPATH, '//*[@id="gnb"]/div[2]')  
ActionChains(driver).move_to_element(rank_now).perform()
```

```
for i in range(1,11):
```

```
    ranklist = driver.find_element(By.XPATH, f'//*[@id="gnb"]/div[2]/div/ol/li[{i}]/a').text  
    print(ranklist)
```

# Selenium

실습해봅시다!

## 실습 7 : 좋아하는 가수의 곡명 10개 가져오기

```
# 좋아하는 가수의 곡명 10개
searchbox = driver.find_element(By.XPATH, '//*[@id="top_search"]')
time.sleep(10)
searchbox.send_keys("몬스타엑스")
time.sleep(5)
searchbox.send_keys(Keys.ENTER)
time.sleep(3)

for i in range(1,11):
    titles = driver.find_element(By.XPATH,f'//*[@id="frm_searchArtist"]/div/table/tbody/tr[{i}]/td[3]/div/div/a[2]').text
    print(titles)
```



# Selenium

실습해봅시다!

실습 8 : 멜론 차트 100의 순위, 곡명, 가수 가져오기.

```
# 순위, 곡명, 가수명 가져오기
infos = driver.find_elements(By.XPATH, '//*[@id="lst50"]' )
time.sleep(3)
for i, info in enumerate(infos, start=1):
    rank = info.find_element(By.XPATH, f"/html/body/div/div[3]/div/div/div[3]/form/div/table/tbody/tr[{i}]/td[2]/div/span[1]").text
    title = info.find_element(By.XPATH, f"/html/body/div/div[3]/div/div/div[3]/form/div/table/tbody/tr[{i}]/td[6]/div/div/div[1]/span/a").text
    singer = info.find_element(By.XPATH, f"/html/body/div/div[3]/div/div/div[3]/form/div/table/tbody/tr[{i}]/td[6]/div/div/div[2]/a").text
    print(rank, title, singer)
```

# Selenium

단점..

1. 사용자에게 제공되는 눈에 보이는 콘텐츠는 전부 크롤링이 가능하지만 컴퓨터 사양에 따라 느릴 수 있다.
2. 브라우저를 직접 켜서 움직이므로 자원을 많이 잡아먹을 수 있다.  
=> 웹 크롤링 시 requests 라이브러리와 같이 사용해 속도 측면을 보완할 수 있다.

# CSV

거의 다 했어요 진짜



지금까지 저장한 데이터를 csv 파일에 저장하기!

## CSV (comma-separated values)

1. 몇 가지 필드를 쉼표(,)로 구분한 텍스트 데이터 및 텍스트 파일
2. 엑셀 프로그램으로 열기 가능!
3. 이후, DB에 직접 넣을 수 있는 파일 형식

# CSV

거의 다 했어요 진짜

```
import csv
```

```
file = open('melon.csv', mode="w", newline='')  
writer = csv.writer(file)  
writer.writerow(["rank", "title", "singer"])
```

```
writer.writerow([rank, title, singer])  
file.close()
```

A1	▼	:	✕	✓	<i>fx</i>	rank
	A	B	C	D	E	
1	rank	title	singer			
2		1 Ditto	NewJeans			
3		2 OMG	NewJeans			
4		3 Hype boy	NewJeans			
5		4 Teddy Bear	STAYC(스테이씨)			
6		5 파이팅 해	부석순 (SEVENTEEN)			
7		6 심(心)	DK(디셈버)			
8		7 사건의 지평선	윤하 (YOUNHA)			
9		8 사랑하기 싫어				
10		9 사랑은 늘	임영웅			
11		10 I Don't Think About You	Charlie Puth			
12		11 CHRISTIAN MIND	Zion Park			
13		12 VIBE (feat. 태양)				
14		13 우리들의 이야기	임영웅			
15		14 Attention	NewJeans			
16		15 다시 만날	임영웅			
17		16 London Beat	임영웅			
18		17 ANTIFRAGILE	LE SSERAFIM (르세라핌)			
19		18 Polaroid	임영웅			
20		19 나비무덤	포맨 (4MEN)			
21		20 이제 나만	임영웅			

melon.csv

# MySQL

그래서 웹에서 어떻게 쓰는데??



오픈 소스 관계형 데이터베이스 관리 시스템(RDBMS)  
사용하기 쉽고 빠르고 안정적이기 때문에 웹 애플리케이션에 널리 사용

[www.mysql.com/downloads/](http://www.mysql.com/downloads/)

데이터베이스란?

- 데이터의 저장소 또는 통합되어 관리되는 데이터의 집합체
- 중복 데이터 제거, 자료 구조화, 효율적 처리를 통해 관리
- 별도의 미들웨어에 의해 관리된다. (데이터베이스 관리 시스템 DBMS)


SQL이란?

- 데이터베이스에서 데이터를 저장하거나 얻기 위해 사용하는 표준화된 언어

# 과제

힘내용

1. 네이버 영화 -> 랭킹 -> 1-20위
2. 각 영화 클릭 -> 개요, 감독, 평점
3. 본인 좋아하는 영화 검색  
-> 제목, 감독, 스크롤 내려서 평점 + 리뷰 개수
4. csv 파일로 저장
5. 실습 7 접근은 selenium, 크롤링은 bs4/request 사용해보기



순위	영화명	변동폭
1	스즈메의 문단속	- 0
2	웅남이	- 0
3	소울메이트	- 0
4	사잔! 신들의 분노	↓ 1
5	더 퍼스트 슬램덩크	- 0
6	대외비	- 0
7	귀멸의 칼날: 상현집결, 그리고 도공 마을로	- 0
8	영웅이	- 0
9	플레인	↑ 2
10	에브리씽 에브리웨어 올 앳 원스+	↓ 1
11	파벨만스	↑ 2
12	다음 소희	↑ 3
13	서치 2	↓ 3
14	똑똑똑	↓ 2
15	아임 히어로 더 파이널	↓ 1
16	이니세린의 밴시	↑ 1
17	앤티맨과 와스프: 퀀텀매니아	↑ 2