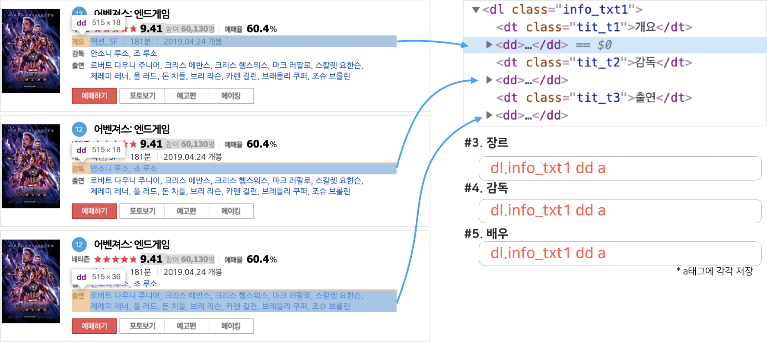
STAGE 1

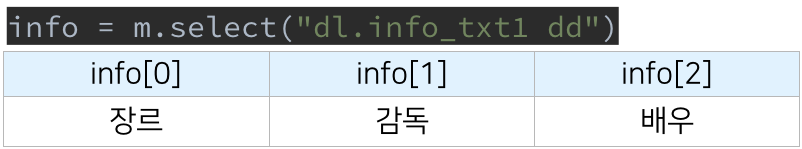
순서를 활용해서 데이터 선택하기

새로 배운 내용: 선택자로 선택되는 데이터



<dd>…</dd>가 3개 존재한다. >> 같은 태그 안의 같은 위치에 존재해서 구분하기 애매함

방법1. 리스트활용하기



select함수를 활용하면 데이터를 리스트 형태로 저장한다. 리스트에 저장된 데이터를 인덱싱 방법을 활용하여 순서를 고려하여 데이터를 수집할 수 있음

    info = m.select("dl.info\_txt1 dd")

    genre = info[0].select("a")

    directors = info[1].select("a")

    actors = info[2].select("a")

방법2. 선택자 응용(nth-of-type)

태그이름 :nth-of-type(N)방법을 활용하면 N번째 태그데이터를 선택할 수 있다.

     genre = m.select("dl.info\_txt1 dd:nth-of-type(1) a")

    directors = m.select("dl.info\_txt1 dd:nth-of-type(2) a")

    actors = m.select("dl.info\_txt1 dd:nth-of-type(3) a")

STAGE 2. 조건에 따라서 데이터 수집하기

<조건문의 기본형태>

if 조건(True/False):

    # 실행문 1

    # 실행문 1

    # ...

else:

    # 실행문 2

    # 실행문 2

    # ...

※문자열과 리스트



#1. in 문자열

articles = ["손흥민은 손으로 상대의 얼굴을 밀며 맞받아쳤다.",

            "AS로마의 니콜로 자니올로",

            "이강인의 팀 동료 페란 토레스"]

for a in articles: # articles : 리스트

    if "손흥민" in a: # a : 문자열

        print("손흥민이 나오는 기사")

    elif "손흥민" not in a:

        print("손흥민이 안나오는 기사")



players = ["손흥민", "이강인", "케빈하르", "백승호", "황의조"]

name = input("선수 이름을 입력해주세요: ")

if name in players: # players : 리스트

    print("출전하는 선수 입니다.")

elif name not in players:

    print("출전하지 않는 선수입니다.")

<네이버 영화에 조건 추가하기>

select함수는 모든 결과를 리스트에 담고 select\_one함수는 하나의 요소만 반환한다

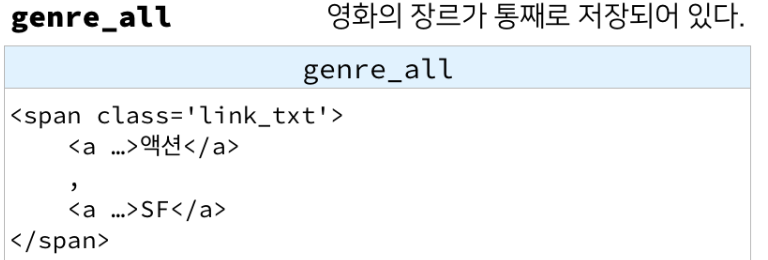
한 영화가 여러 장르를 가지고 있을 수 있으므로 select함수로 생성된 리스트로는 무리가 있다.

모든 장르에 대한 정보를 가지고 있는 span.link\_txt로 데이터를 새롭게 수집해서 조건문으로 사용한다

    genre\_all = m.select\_one("dl.info\_txt1 dd:nth-of-type(1) span.link\_txt")

    if "액션" not in genre\_all.text:

        continue



STAGE 3. 링크안에서 데이터수집하기

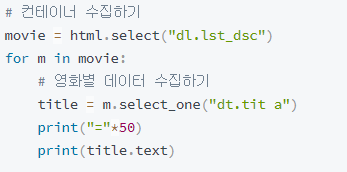
<연결된 페이지로 이동하기>

a태그의 href 속성은 연결되는 페이지의 주소를 저장하고

img 태그의 src속성은 이미지의 주소를 저장하고 있다.

# 속성값(href)가져오기

예를 들어서



에서 title에는 a 태그가 저장되어있다. 이 a 태그의 href속성값에는 링크된 페이지의 주소가 들어있다.

다만 href속성값에 들어있는 페이지의 주소에는 기본홈페이지 주소가 생략되어있다(이 부분은 사이트마다 다르다. 생략되어있는 사이트도 있고, 아닌 사이트도 있다)

title.attrs[“href”] 로 a태그의 href값(링크된 페이지 주소)를 문자열로 가져올 수 있다.

링크된 페이지 주소와 생략되어있던 기본홈페이지 주소를 문자열 더하기를 통해서 연결해준다.

이 새롭게 만들어진 문자열로 다시 requests.get(url, …) BeautifulSoup(…)를 해주면 클릭했을 때 들어가지는 페이지에서 데이터를 수집할 수 있다.

for m in movie:

    # 영화별 데이터 수집하기

    title = m.select\_one("dt.tit a")

    print("="\*50)

    print(title.text)

    url = "https://movie.naver.com" + title.attrs["href"]

    # print(url)

    # url(상세페이지)에 접속, html 파싱

    raw\_each = requests.get(url, headers={"User-Agent":"Mozilla/5.0"})

    html\_each = BeautifiulSoup(raw\_each.text, 'html.parser')

그 이후엔 지금까지의 데이터수집 절차처럼 컨테이너를 수집하고, 상세 데이터를 수집해주면 된다

STAGE4. 이미지 데이터 수집하기

STAGE3에서의 a태그 href속성과 img태그의 src속성은 비슷하다

href속성은 페이지 주소를 저장하고 있고, src속성은 이미지의 실제위치를 저장하고 있음

<이미지 데이터 저장하기>

파이썬 기본 라이브러리인 urllib라이브러리의 urlretreive모듈을 사용해서

urlreteieve(“이미지 주소”,”저장할 이미지의 이름”)을 실행시켜주면 저 이름대로 이미지가 생성된다

여기서 “이미지 주소”를 src속성값에 저장되어 있으므로 attrs[“src”]를 이용해서 문자열로 가져올 수 있다.

※ 꼭 이미지가 아니더라도 데이터의 주소를 알고 있다면 영상이나 프로그램 등을 모두 다운가능

<데이터 수집기를 통해 이미지 데이터 저장하기>

상세페이지(터미네이터, 조커 등)마다 들어가서 포스터이미지를 가져오려면

1. 그 상세페이지에서의 포스터 src속성값을 찾고

2. urlretreieve를 이용해서 이미지로 저장하기

2단계를 거쳐야한다

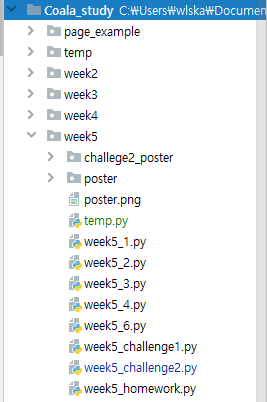
여기서 src속성값을 가져오려면 여태 데이터를 수집했던 것처럼 포스터의 태그값을 받아오고, 거기서 src속성을 가져오면 된다



<파일 경로 설정/ 이름바꾸기>



파일이름(경로)에서 / 는 디렉토리 구조를 의미



week5폴더에 poster라는 폴더가 있고, week5와 같은 계층인 temp, week2 …등이 있다

파일 이름을 “어벤져스.png”라고 했으면 동일 폴더내(week5폴더)에 “어벤져스.png”라는 이미지가 생성되고

“poster/어벤져스.png”라고 파일이름을 정했으면 week5폴더의 하위폴더인 poster폴더에 “어벤져스.png”가 생성된다

만약 “../temp/어벤져스.png”라고 정했다면 상위디렉토리(Coala\_Study폴더)의 temp폴더에 “어벤져스.png”가 생성된다