WEEK05_STAGE1,2_장하은

Stage1. 순서를 활용해서 데이터 선택하기

아이디나 클래스로 선택자를 구분할 수 없고,
같은 태그 안의 같은 위치에 있기 때문에 상하관계를 통해서도 구분할 수 없는상황

방법1. 리스트 활용하기

```
info = m.select("dl.info_txt1 dd")
```

info[0]	info[1]	info[2]
장르	감독	배우

`select`함수를 활용하면 데이터를 리스트 형태로 저장
> 리스트에 저장된 데이터를 인덱싱 방법을 활용하여 특정 순서의 데이터를 수집
>>누락 데이터가 오류로 뜬다

```
for m in moive
    info = m.select("dl.info_txt1 dd")
    genre = info[0].select("a")
    directors = info[1].select("a")
    actors = info[2].select.("a")
```

- 각 `dd`태그 안의 `a`태그에 장르, 감독, 배우에 대한 데이터가 들어있음
- 한 영화에 여러개의 장르, 감독, 배우에 대한 데이터가 있을 수 있으므로 `select`함수를 활용

방법 2. 선택자 응용하기(nth-of-type)

태그이름:`nth-of-type(N)` 방법을 활용하면 N번째 태그 데이터를 선택할 수 있다.
>누락 데이터가 오류로 뜨지 않는다

```
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")
```

주의

- 프로그래밍 언어는 대부분 숫자를 0부터 세지만 `nth-of-type()`방법의 경우는 숫자를 1부터 셈
- `nth-of-type()`의 앞에는 클래스 또는 아이디가 올 수 없기 때문에 해당하는 태그가 몇번째 태그인지도 꼭 확인해야 함.

Stage2.조건에 따라서 데이터 선택하기

`try/except`문과 비슷한 구조를 띄고 있으며, `if` 뒤의 조건이 참인 경우에는 `if` 안의 실행문이 실행되고, 거짓인 경우에는 `else` 안의 실행문이 실행됩니다.

▼ 참고

`try/except`에서는 except가 필수였지만, `if`문에서는 else를 꼭 써줄 필요는 없다.

연산자

비교연산자	설명	결과1 (A=3, B=1)	결과2 (A=2, B=2)
`A > B`	A가 B보다 크다.	**True**	**False**
`A >= B`	A가 B보다 크거나 같다.	**False**	**True**
`A == B`	A와 B가 같다.	**False**	**True**
`A != B`	A와 B가 같지않다.	**True**	**False**

조건 연산자

비교연산자	설명	결과1 (A=True, B=True)	결과2 (A=True, B=False)
`A and B`	A와 B가 둘다 참이다.	**True**	**False**
`A or B`	A 또는 B가 참이다.	**True**	**True**
`not B`	B가 거짓이다.	**False**	**True**

- `A or B` 는 둘 중 한 조건만 충족해도 됨
- `A and B` 는 둘 다 충족해야 함

`i % 2 == 0` < 2의 배수인 경우

```
articles = ["손흥민은 손으로 상대의 얼굴을 밀며 맞받아쳤다.",
            "AS로마의 니콜로 자니올로",
            "이강인의 팀 동료 페란 토레스"]

for a in articles:
    if "손흥민" in a:
        print("손흥민이 나오는 기사")
    elif "손흥민" not in a:
        print("손흥민이 안나오는 기사")
    else "이강인" in a:
        print("이강인이 나오는 기사")
```

- `elif`는 `if`이외에 새로운 조건을 추가하고 싶을때, 무한정 추가 가능

리스트와 문자열의 구분

비교연산자	설명
`A in 문자열`	A가 문자열 안에 포함되어있다.
`A not in 문자열`	A가 문자열 안에 포함되어있지않다.

비교연산자	설명
`A in 리스트`	A가 리스트에 있다.
`A not in 리스트`	A가 리스트에 없다.

- 문자열: 문자열 안에 일부가 포함되어 있을 때
- 리스트: 리스트의 값 중에 해당하는 값이 있을 때

예시

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

name = input("선수 이름을 입력해주세요: ")
if name in players:
    print("출전하는 선수 입니다.")
elif name not in players:
    print("출전하지 않는 선수입니다.")
```

실행결과(이강을 입력한 경우)

선수 이름을 입력해주세요:

이강

출전하지 않는 선수입니다.

```
if float(score.text) < 8.5:
    continue
```

- `score.text`에는 평점이 문자형식으로 저장되어 있기 때문에 실수형(`float`)으로 형변환
- `continue`는 반복문 안에서 현재의 실행문을 종료하고 다음 순서의 반복문이 실행

장르 조건 확인하기

- 장르의 경우 한 영화가 여러가지 장르를 가지고 있을 수 있기 때문에 `genre` 리스트가 가지고 있는 모든 값들을 확인해야함. > `a`가 아닌 `span.link_text` 검색

```
genre = m.select("dl.info_txt1 dd:nth-of-type(1) a")

# genre_all을 수집
# "액션"장르가 아니면 continue
    genre_all = m.select_one("dl.info_txt1 dd:nth-of-type(1) span.link_txt")
    if "액션" not in genre_all.text:
        continue
```

주의

`gere_all`은 태그데이터 안에서 선택했기때문에 `gere_all.text`로 바꿔야함