

Final Project

웹페이지 크롤러 만들기

인공지능융합학과 2020310967 황새연

선정한 웹사이트:

잡플래닛 데이터 분석 공고

https://www.jobplanet.co.kr/job_postings/search?rs_act=browse&rs_con=job&rs_element=job_postings&occupation_level2_ids%5B%5D=11613

분류 정의:

기업명 / 제목 / 별점 / 평균연봉 / 직원 추천

잡플래닛 데이터 분석 공고 사이트에 있는 모든 데이터들을 기업명, 제목, 별점, 평균 연봉, 직원 추천을 기준으로 크롤링 하였음.

	(재)한국의류시험연구원 2021년 정규직전환형 직원 채용공고 D-5 @Sponsored 저장
(재)한국의류시험연구원	★ 2.6 평균 연봉 5,422만원 직원 추천 36.0%
데이터분석 · 시스템엔지니어 · 환경/수질/대기/폐기물 · 국내 비영리단체/협회/교육재단 · 신입 · 경력 · 경력무관	더보기
	GIS 엔지니어 D-36 별점 / 평균 연봉 / 직원 추천 저장
(주)카카오모빌리티	★ 3.9 평균 연봉 5,630만원 직원 추천 86.0%
데이터분석 · 시스템엔지니어 · 대기업 계열사/자회사 · 경력	더보기
	AI - SW D-3 저장
(주)한화/방산	★ 3.3 평균 연봉 5,377만원 직원 추천 56.0%
경기 · 데이터분석 · 소프트웨어엔지니어 · 대기업 · 경력	더보기
	[잡플래닛 매칭] 데이터 사이언티스트(분석가) D-67 (채용시 마감) 저장
헤드헌팅 (주)자란다	★ 3.4 직원 추천 66.0%
서울 · 데이터분석 · 소프트웨어아키텍트 · 중소기업 · 경력	더보기
	[잡플래닛 매칭] AI Application Engineer D-8 (채용시 마감) 저장
헤드헌팅 네이버웹툰	★ 4.3 평균 연봉 4,590만원 직원 추천 86.0%
경기 · 데이터분석 · 소프트웨어아키텍트 · 소프트웨어엔지니어 · 중견기업 · 경력	더보기

실행 예시:

잡플래닛 웹사이트를 입력하면
현재 시간을 파일 이름으로 한
Csv 파일이 생성되고,
소요시간, 시작시간, 완료시간을
알려줌

CSV 예시:

총 7페이지로, 7페이지 모두 크롤링 하였고
115개의 분석 결과가 도출됨.

첫번째 데이터를 분석해보면,

기업명: (재)한국의류시험연구원

제목: (재)한국의류시험연구원

2021년 정규직전환형 직원 채용공고

별점: 2.6

평균연봉: 평균 연봉 5,422만원

직원추천: 36.0% 로, 잡플래닛 공고와 동일함.

```
웹사이트 주소:https://www.jobplanet.co.kr/job_postings/search?_rs_act=browse&_rs_con=job&_rs_element=job_postings&occupation_level2_ids%5B%5D=11613
202105251745.csv 파일에 크롤링 결과가 저장되었습니다.
소요시간: 8초
시작시간: 2021년 05월 25일 17시 45분 33초
완료시간: 2021년 05월 25일 17시 45분 42초
계속하시겠습니까? 네
웹사이트 주소:https://www.jobplanet.co.kr/job_postings/search?_rs_act=browse&_rs_con=job&_rs_element=job_postings&occupation_level2_ids%5B%5D=11613
202105251745.csv 파일에 크롤링 결과가 저장되었습니다.
소요시간: 7초
시작시간: 2021년 05월 25일 17시 45분 44초
완료시간: 2021년 05월 25일 17시 45분 51초
계속하시겠습니까? 아니오
>>> |
```

	A	B	C	D	E	F	G
1		기업명	제목	별점	평균연봉	직원추천	
2	0	(재)한국의	(재)한국의	2.6	평균 연봉	직원 추천	36.0%
3	1	(주)카카오	GIS 엔지니	3.9	평균 연봉	직원 추천	80.0%
4	2	(주)한화/병	AI - SW	3.3	평균 연봉	직원 추천	56.0%
5	3	(주)자란다	[잡플래닛	3.4	제공X	직원 추천	66.0%
6	4	네이버웹툰	[잡플래닛	4.3	평균 연봉	직원 추천	86.0%
7	5	네이버웹툰	[잡플래닛	4.3	평균 연봉	직원 추천	86.0%
8	6	네이버웹툰	[잡플래닛	4.3	평균 연봉	직원 추천	86.0%
9	7	잡플래닛	[국내 대표	제공X	제공X	제공X	
10	8	(주)카카오	[클라우드	4.3	평균 연봉	직원 추천	83.0%
11	9	잡플래닛	[유통 대기	제공X	제공X	제공X	
12	10	잡플래닛	[유통 대기	제공X	제공X	제공X	
13	11	(주)카카오	[데이터서	4.3	평균 연봉	직원 추천	83.0%



(재)한국의류시험연구원 2021년 정규직전환형 직원 채용공고 D-5 @Sponsored

저장

(재)한국의류시험연구원 ★ 2.6 | 평균 연봉 5,422만원 | 직원 추천 36.0%

데이터분석 · 시스템엔지니어 · 환경/수질/대기/폐기물 · 국내 비영리단체/협회/교육재단 · 신입 · 경력 · 경력무관

더보기 ▼

코드 설명

Library:

```
from bs4 import BeautifulSoup #bs4에서 BeautifulSoup 함수를 import
from urllib.request import Request, urlopen #urllib에서 urlopen, Request 함수를 import
import pandas as pd #pandas를 import해서 pd로 사용
import time #소요시간, 시작시간, 완료시간 print하기 위함
```

Def:

```
def Crawling(url): #크롤링 함수

    company=[] #기업명 리스트
    title=[] #제목 리스트
    score=[] #별점 리스트
    salary=[] #평균연봉 리스트
    recommend=[] #직원추천 리스트
```

기업명	제목	별점	평균연봉	F	G
(재)한국의	(재)한국의	2.6	평균 연봉	직원추천	
(주)카카오	2GIS 엔지니어	3.9	평균 연봉	직원 추천	36.0%
(주)한화/병	영AI - SW	3.3	평균 연봉	직원 추천	80.0%
(주)자란다	잡플래닛	3.4	제공X	직원 추천	56.0%
네이버웹툰	잡플래닛	4.3	평균 연봉	직원 추천	66.0%
네이버웹툰	잡플래닛	4.3	평균 연봉	직원 추천	86.0%
네이버웹툰	잡플래닛	4.3	평균 연봉	직원 추천	86.0%
잡플래닛	(국내 대표기	제공X	제공X	직원 추천	86.0%
(주)카카오	클라우드	4.3	평균 연봉	제공X	
잡플래닛	유통 대기	제공X	제공X	직원 추천	83.0%
잡플래닛	유통 대기	제공X	제공X	제공X	
(주)카카오	데이터서터	4.3	평균 연봉	제공X	
(주)카카오	데이터서터	4.3	평균 연봉	직원 추천	83.0%
잡플래닛	(국내 대기	제공X	제공X	직원 추천	83.0%
잡플래닛	(국내 대기	제공X	제공X	제공X	
(주)카카오	카카오엔터	3.2	평균 연봉	제공X	
(주)카카오	카카오엔터	3.2	평균 연봉	직원 추천	49.0%
(주)카카오	카카오엔터	3.2	평균 연봉	직원 추천	49.0%

Def:

```
for n in range (1,8): #page 1~7
```

```
URL=url+"&page="+str(n) #페이지에 따라 url 변경  
req = Request(URL,headers={'User-Agent': 'Mozilla/5.0'}) #크롤링할 수 있도록 request  
openUrl = urlopen(req).read() #req 호출해 url 연다  
soup = BeautifulSoup(openUrl, 'html.parser') #BeautifulSoup 이용해 구문 분석
```

1. 잡플래닛 데이터 분석 공고가 7페이지까지 있으므로 for문으로 url 주소를 바꾸도록 했음.

Ex) 잡플래닛의 2번째 페이지 주소는

https://www.jobplanet.co.kr/job_postings/search?_rs_act=browse&_rs_con=job&_rs_element=job_postings&occupation_level2_ids%5B%5D=11613&page=2

2. urlopen을 그냥 하게 되면 크롤링 접속 차단이 되므로 header에 로봇이 아니라는 정보를 넣어줌.

3. Urlopen을 통해 req 변수 안에 담긴 url을 열도록 함.

4. BeautifulSoup을 통해 구문 분석

Def:

```
... <button class="btn_open">...</button> == $0
```

#기업명

```
company_text = soup.find_all('button',{'class':{'btn_open'}})
```

#('태그명',{'속성명1':{'값1'}}) 찾기

#soup에서 <button class="btn_open"></button> 출력

```
for i in company_text: #company_text 요소
```

```
    company.append(i.get_text()) #company 리스트에 기업명만 append
```

#제목

```
title_text = soup.find_all('a',{'class':{'posting_name'}})
```

```
for i in title_text: #title_text 요소
```

```
    title.append(i.get_text()) #title 리스트에 제목만 append
```

a.posting_name 405.48 x 22

(재)한국의류시험연구원 2021년 정규직전환형 직원 채용공고 D-5

```
<a class="posting_name" data-no-turbolink="true" href="/job/search?
=%28%EC%9E%AC%29%ED%95%9C%EA%B5%AD%EC%9D%98%EB%A5%98%EC%8B%9...B%5D=1:
93612&rs_con=job_postings&rs_act=search&rs_element=search_result'
target="_blank">(재)한국의류시험연구원 2021년 정규직전환형 직원 채용공고
</a> == $0
```

1. html 태그, 속성, 값을 이용하여 원하는 데이터(기업명, 제목)만 추출함.
2. For문 활용하여 company_text, title_text에 있는 값 추출. Company, title 리스트에 데이터 append (cf. get_text()는 값만 추출하게 해줌)

Def:

#별점, 평균연봉, 직원추천

info=[]

#별점, 평균연봉, 직원추천을 append할 info 리스트

#리뷰가 10개 미만인 기업은 별점, 평균연봉, 직원추천을 open하지 않으므로, 혹은 평균연봉을 제공하지 않는 기업이 있으므로 따로 리스트 생성

info_text = soup.find_all('div',{'class':{'jp_data_set'}}) #별점, 평균연봉, 직원추천을 모두 크롤링

for i in range(len(info_text)): #info_text 길이만큼 반복

data=info_text[i].get_text().strip().split('\n|\n') #info_text에서 별점, 평균연봉, 직원추천을 하나의 리스트로 data에 저장. 공백 제거 및 \n|\n을 기준으로 split

info.append(data) #info 리스트에 data append

```
[<div class="jp_data_set">
<a class="rate" href="/companies/64418/reviews" target="_blank"><em>3.6</em></a>
<span class="bar_txt">|</span>
<a class="salary" href="/companies/64418/salaries" target="_blank">평균 연봉 <em>6,713</em>만원</a>
<span class="bar_txt">|</span>
<a class="recommend" href="/companies/64418/reviews" target="_blank">직원 추천 <em>72.0</em>%</a>
</div>, <div class="jp_data_set">
<a class="rate" href="/companies/75788/reviews" target="_blank"><em>4.4</em></a>
<span class="bar_txt">|</span>
<a class="salary" href="/companies/75788/salaries" target="_blank">평균 연봉 <em>5,097</em>만원</a>
<span class="bar_txt">|</span>
<a class="recommend" href="/companies/75788/reviews" target="_blank">직원 추천 <em>75.0</em>%</a>
</div>, <div class="jp_data_set">
리뷰/연봉정보 <em>10</em>개 이하는 평점 미리보기를 제공하지 않습니다.
</div>, <div class="jp_data_set">
...
```



국내 대표 핀테크 스타트업 기업 - 데이터 사이언티스트 D-67 (채용시 마감)

헤드헌팅 잡플래닛 매칭서비스 리뷰/연봉정보 10개 이하는 평점 미리보기를 제공하지 않습니다.

...

><div class="jp_data_set">...</div> == \$0

1. 위 사진 처럼 리뷰/연봉정보 10개 이하는 평점 미리보기를 제공하지 않음.
따라서 <div class="jp_data_set"> 을 크롤링 => 별점, 평균연봉, 직원추천 한꺼번에 크롤링 하게 됨.
2. For문 활용하여 info 리스트에 데이터 append
(strip()은 공백을 제거하고, split을 활용하여 '\n|\n'도 제거해줌)

Def:

```
[[ '3.6', '평균 연봉 6,713만원', '직원 추천 72.0%' ], [ '4.4', '평균 연봉 5,097만원', '직원 추천 75.0%' ], [ '리뷰/연봉정보 10개 이하의 평점 미리보기를 제공하지 않습니다.' ], [ '3.9', '평균 연봉 7,606만원', '직원 추천 84.0%' ], [ '리뷰/연봉정보 10개 이하의 평점 미리보기를 제공하지 않습니다.' ], [ '리뷰/연봉정보 10개 이하의 평점 미리보기를 제공하지 않습니다.' ], [ '3.3', '평균 연봉 5,165만원', '직원 추천 59.0%' ], [ '3.3', '평균 연봉 5,165만원', '직원 추천 59.0%' ], [ '3.1', '평균 연봉 7,035만원', '직원 추천 48.0%' ], [ '2.9', '평균 연봉 7,178만원', '직원 추천 36.0%' ], [ '3.7', '평균 연봉 3,261만원', '직원 추천 62.0%' ], [ '3.7', '평균 연봉 3,261만원', '직원 추천 62.0%' ], [ '2.6', '평균 연봉 3,987만원', '직원 추천 26.0%' ], [ '2.6', '평균 연봉 3,987만원', '직원 추천 26.0%' ], [ '3.2', '평균 연봉 3,259만원', '직원 추천 48.0%' ], [ '3.2', '평균 연봉 3,259만원', '직원 추천 48.0%' ], [ '리뷰/연봉정보 10개 이하의 평점 미리보기를 제공하지 않습니다.' ] ]
```

```
for i in range(len(info)): #info 리스트 길이만큼 반복
    if len(info[i]) == 3: #별점, 평균연봉, 직원추천이 모두 존재
        score.append(info[i][0]) #score 리스트에 별점 append
        salary.append(info[i][1]) #salary 리스트에 평균연봉 append
        recommend.append(info[i][2]) #recommend 리스트에 직원추천 append
    elif len(info[i]) == 2: #별점, 직원추천만 존재
        score.append(info[i][0])
        salary.append('제공X')
        recommend.append(info[i][1])
    elif len(info[i]) == 1: #별점, 평균연봉, 직원추천 모두 제공X
        score.append('제공X')
        salary.append('제공X')
        recommend.append('제공X')
```

1. for문을 활용해서 info 리스트 길이만큼 반복
2. Info[i]의 길이가 3이면, 별점, 평균연봉, 직원추천이 모두 존재하는 것임 → score, salary, recommend 리스트에 모두 append
길이가 2이면, 별점, 직원추천만 존재하는 것 → score, recommend 리스트에 값 append, salary 에는 '제공X' append
길이가 1이면, 모두 존재 X → score, salary, recommen에 '제공X' append

Def:

#pandas, 딕셔너리 사용해서 위 리스트들을 보기 쉽게 정리

```
result = pd.DataFrame({'기업명': company,  
                        '제목': title,  
                        '별점': score,  
                        '평균연봉': salary,  
                        '직원추천': recommend  
                        })
```

```
return result
```

1. Pandas 와 딕셔너리 사용해서 위 리스트를
 보기 쉽게 정리함.
(202105252232 기준으로 데이터 117개)

```
>>> print(rst)
```

	기업명	...	직원 추천
0	(재)한국의류시험연구원	...	직원 추천 36.0%
1	이지케어텍(주)	...	직원 추천 53.0%
2	(주)루티너리	...	제공X
3	(주)카카오모빌리티	...	직원 추천 80.0%
4	(주)한화/방산	...	직원 추천 56.0%
...
113	한국전자인종(주)	...	직원 추천 26.0%
114	한국전자인종(주)	...	직원 추천 26.0%
115	리디(주)	...	직원 추천 48.0%
116	리디(주)	...	직원 추천 48.0%
117	위프로스퍼(주)	...	제공X

[118 rows x 5 columns]

Def:

#pandas, 딕셔너리 사용해서 위 리스트들을 보기 쉽게 정리

```
result = pd.DataFrame({'기업명': company,  
                        '제목': title,  
                        '별점': score,  
                        '평균연봉': salary,  
                        '직원추천': recommend  
                        })
```

```
return result
```

1. Pandas 와 딕셔너리 사용해서 위 리스트를
 보기 쉽게 정리함.
(202105252232 기준으로 데이터 117개)

```
>>> print(rst)
```

	기업명	직원 추천
0	(재)한국의류시험연구원	직원 추천 36.0%
1	이지케어텍(주)	직원 추천 53.0%
2	(주)루티너리	제공X
3	(주)카카오모빌리티	직원 추천 80.0%
4	(주)한화/방산	직원 추천 56.0%
...
113	한국전자인종(주)	직원 추천 26.0%
114	한국전자인종(주)	직원 추천 26.0%
115	리디(주)	직원 추천 48.0%
116	리디(주)	직원 추천 48.0%
117	위프로스퍼(주)	제공X

[118 rows x 5 columns]

Main:

```
#main
while True:
    jobplanet=input('웹사이트 주소: ') #https://www.jobplanet.co.kr/job_postings/search?_rs_act=browse&_rs_con=job&_rs_element=job_postings&occupation_level2_ids%5B%5D=11613 입력

    start=time.localtime(time.time()) #시작시간
    st_check=time.time() #소요시간 계산 위한 시작시간 체크
    save=time.strftime('%Y%m%d%H%M.csv',start) #출력파일 이름은 현재 시간으로
    #time.strftime('포맷',시간객체)

    rst=Crawling(jobplanet) #함수에 url 입력

    #csv 파일로 저장
    rst.to_csv(save)
    print('%s'%(save), '파일에 크롤링 결과가 저장되었습니다.')

    end=time.localtime(time.time()) #완료시간
    end_check=time.time() #소요시간 계산 위한 완료시간 체크

    print('소요시간: %d초'%(end_check-st_check)) #소요시간 계산
    print(time.strftime('시작시간: %Y년 %m월 %d일 %H시 %M분 %S초', start))
    print(time.strftime('완료시간: %Y년 %m월 %d일 %H시 %M분 %S초', end))

    con=input('계속하시겠습니까? ') #'네' 입력시 처음으로 되돌아감. 이외 입력시 종료
    if con == '네':
        continue
    else:
        break
```

1. While문 활용해서 con이 '네' 일시, 계속 반복하도록 함.
2. Time library 활용해서 시작시간, 소요시간, 완료시간 print. 현재시간을 파일 이름으로 지정.
3. Crawling 함수의 return 값을 rst에 저장해서 csv 파일로 저장.

감사합니다.