텍스트이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| **처갓집치킨 데이터 크롤링**  **코드 분석 레포트** |

|  |  |
| --- | --- |
| 과 목 명 : | 데이터크롤링 |
| 담당 교수님 : | 이지영 교수님 |
| 전 공 : | 컴퓨터과학과 |
| 학 번 : | 2017301023 |
| 이 름 : | 김태환 |
| 제 출 일 : | 2022.10.12 |

**목 차**

1. 소스코드 분석 3 ~ 7
2. 결과 확인 8

**1. 소스코드 분석**

* **소스 코드**

1. import urllib.request
2. from bs4 import BeautifulSoup
3. import pandas as pd
4. import datetime
5. from itertools import count

* **import module**

- urllib.request: 쿠키, 리디렉션 등의 URL을 여는 데에 도움이 되는 함수와 클래스를 정의한 것으로, 간단하게 웹 페이지 요청 및 데이터를 가져오는 것이 가능하다.

- BeautifulSoup: HTML 문서를 분석(parse)하여 원하는 부분만 쉽게 뽑아낼 수 있도록 하는 라이브러리이다.

- pandas: 데이터를 원하는 형태로 가공하기 위해 지우기, 재편성, 재구조화, 합치기 등의 기능을 제공하는 데이터 분석 라이브러리이다.

- datetime: 날짜와 시간을 조작하는 클래스를 제공하는 모듈이다.

- count: itetools는 효율적인 루핑을 위한 이테레이터(반복자)를 만드는 모듈이다. 그 안의 count() 함수는 사용자가 원하는 시작지점(start)부터 지정한 간격(step)만큼 무한히 반복을 돌리는 함수이다.

* **메소드**

-

* **변수**

-

* **코드 흐름**

1. 필요한 모듈들을 임포트한다.

**1. 소스코드 분석**

* **소스 코드**

#[CODE 1]

1. def CheogajipAddress(result):

2. for page\_idx in count(0, 1):

3. Cheogajip\_URL='https://www.cheogajip.co.kr/bbs/board.php?bo\_table=store&page=%s' %str(page\_idx + 1)

4. print(Cheogajip\_URL)

5. response = urllib.request.urlopen(Cheogajip\_URL)

6. soupData = BeautifulSoup(response, 'html.parser')

7. tbody\_tag = soupData.find('tbody')

8. for store\_tr in tbody\_tag.findAll('tr'):

9. if(len(store\_tr) <= 1):

10. #마지막 페이지 이상을 넘어가지 않도록 크롤링을 끝낸다.

11. print("LAST PAGE = ", page\_idx)

12. return

13. tr\_tag = list(store\_tr.strings)

14. store\_name = tr\_tag[1]

15. store\_address = tr\_tag[3]

16. store\_sido\_gu = store\_address.split()[:2]

17. store\_phone = tr\_tag[5]

18. result.append([store\_name] + store\_sido\_gu + [store\_address] + [store\_phone])

19. #print(tr\_tag)

* **메소드**

- CheogajipAddress(result): 처갓집 매장 찾기 페이지에서 1페이지부터 마지막 페이지까지 탐색을 하며 각 매장 정보(매장명, 매장 주소, 시(구), 전화번호를 합쳐)를 배열에 저장하여 반환한다.

- count(0, 1): 처갓집 매장 찾기 페이지에서 1페이지부터 하나씩, 모든 페이지를 범위로 반복문을 진행한다.

- str(page\_idx + 1): 반복문 인덱스는 0부터 시작하므로 +1인 1페이지부터 탐색한다는 의미로 반복문은 URL 내에 페이지 값을 1부터 진행한다.

- urllib.request.urlopen(URL): URL 요청을 인스턴트화하기 위해 사용한다.

- BeautifulSoup(response, ‘html.parser’): BeautifulSoup 객체를 생성하면서 HTML 문서를 저장한 response를 HTML 구조(‘html.parser’)로 분석한다.

- soupData.find(‘tbody’): Beautiful 객체에서 tbody 태그에 해당하는 HTML 태그들을 파싱한다.

- tbody\_tag.findAll(‘tr’): 위에서 찾은 tbody 태그 내에 있는 tr 태그들을 파싱한다.

- list(store\_tr.strings): 위에서 찾은 tr 태그들을 문자열 배열로 치환한다.

- result.append(~): store\_tr에서 찾은 매장 정보들을 result 배열에 저장한다.

* **변수**

- Cheogajip\_URL: 데이터 크롤링을 할 처갓집 매장 찾기 URL을 저장한 변수이다.

- response: Cheogajip\_URL 요청 인스턴트이다.

- soupData: HTML을 분석한 객체이다.

- tbody\_tag: soupData에서 tbody에 해당하는 태그들을 파싱하여 저장한 변수이다.

- store\_tr: tbody\_tag 내에 있는 각 tr 태그를 각 반복문마다 저장한 변수이다.

- tr\_tag: store\_tr를 문자열 배열로 저장한 변수이다.

- store\_name: tr\_tag에서 매장명에 해당하는 문자열을 저장한 변수이다.

- store\_address: tr\_tag에서 매장 주소에 해당하는 문자열을 저장한 변수이다.

- store\_sido\_gu: tr\_tag에서 매장 위치(시/도/구)에 해당하는 문자열을 저장한 변수이다.

- store\_phone: tr\_tag에서 매장 전화번호에 해당하는 문자열을 저장한 변수이다.

- result: 위에서 매장 정보를 저장한 문자열을 합쳐 배열에 저장한 변수이다.

* **코드 흐름**

1. (2~3) 첫페이지부터 무한히 반복하며 Cheogajip\_URL(파싱할 웹 페이지)를 설정한다.

2. (5) URL을 요청하여 응답 받은 웹 페이지를 저장한다.

3. (6) Beautiful 객체(html 분석) 생성하여 soupData에 저장한다.

4. (7) soupData에서 tbody에 해당하는 태그들을 가져와 tbody\_tag에 저장한다.

5. (8) tbody\_tag 내에 있는 tr 마다 반복문을 돌리며 해당 tr을 store\_tr로 지정한다.

6. (9~12) store\_tr 내에 정보가 없다면 데이터 크롤링을 종료한다.

7. (13) store\_tr를 문자열 배열로 바꾸고 tr\_tag에 저장한다.

8. (14~17) tr\_tag 내에서 필요한 매장 정보를 각 변수에 저장한다.

9. (18) 8.에서 찾아낸 매장 정보 변수들을 result 배열에 하나로 합쳐 저장한다.

**1. 소스코드 분석**

* **소스 코드**

#[CODE 0]

1. def cswin\_Cheogajip():

2. result = []

3. print("CHEOGAJIP ADDRESS CRAWLING START")

4. CheogajipAddress(result) #[CODE 1] 호출

5. cheogajip\_table = pd.DataFrame(result, columns = ('store', 'sido', 'gungu', 'store\_address', 'store\_phone'))

6. cheogajip\_table.to\_csv("./cheogajip.csv", encoding = "cp949", mode = 'w', index = True)

7. del result[:]

8. print('FINISHED')

9. if \_\_name\_\_ == '\_\_main\_\_':

10. cswin\_Cheogajip()

* **메소드**

- cswin\_Cheogajip(): main 프로세스가 실행될 때 실행할 함수(main 함수 역할)이다.

- CheogajipAddress(result): result 배열(빈 배열)를 인자로 넘겨 찾고자 하는 처갓집치킨 매장 찾기페이지에서 데이터들을 저장한다.

- pd.DataFrame(~): pandas를 활용하여 result 데이터를 저장할 테이블 형태의 데이터 프레임을 생성한다.

- cheogajip\_table.to\_csv(~): 위에서 생성한 데이터 프레임(cheogajip\_table)을 csv 형태의 파일로 변환하는 메소드이다.

- del result[:]: 종료하지 않고 프로그램을 재실행한 경우 result 배열에 데이터가 중복적으로 저장되는 것을 방지하기 위해 데이터 크롤링 작업이 완료되면 result 배열을 초기화한다.

* **변수**

- result: 찾고자 하는 데이터들을 저장할 배열 변수이다.

- cheogajip\_table: result 데이터를 테이블로 정리한 데이터 프레임 객체이다.

* **코드 흐름**

1. (2) result 배열 변수를 생성한다.

2. (4) CheogajipAddress(result)를 통해 #[CODE 1]을 호출하여 찾는 데이터를 result에 저장한다.

3. (5) 찾은 데이터(result)를 기반으로 데이터 프레임 테이블을 생성하여 cheogajip\_table에 저장한다.

4. (6) cheogajip\_table을 csv 형태의 파일로, “cheogajip.csv” 이름으로 저장한다.

5. (7) result 배열을 초기화한다(중복 저장 방지).

6. (9~10) 현재 스크립트 파일이 메인 프로그램으로 사용되는지, 모듈로 사용되는지 구분하기 위함으로, 메인 프로그램인 경우 실행한다.

**2. 결과 확인**

테이블이(가) 표시된 사진

자동 생성된 설명

“cheogajip.csv” 파일을 열어보면 지정한 데이터 프레임 형식에 맞게 테이블이 만들어졌다.

해당 파일에서는 처갓집 치킨 매장 찾기 페이지에서 매장명, 매장 위치(시/도/군/구), 매장 주소, 매장 전화번호 모두 저장됐음을 확인 할 수 있다.