4. 가장 뜨는 제주도 핫플레이스는 어디일까?

필요한 라이브러리 불러오기

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
# 코랩을 시작할 때 아래코드를 한 번 돌려줍니다.
!pip install selenium
!apt-get update
!apt install chromium-chromedriver
!cp /usr/lib/chromium-browser/chromedriver /usr/bin

from selenium import webdriver
from selenium.webdriver.common.by import By

import sys
sys.path.insert(0,'/usr/lib/chromium-browser/chromedriver')

!pip install selenium
!apt-get update

# (최초 1회)
!apt install chromium-chromedriver
!cp /usr/lib/chromium-browser/chromedriver '/content/drive/MyDrive/Colab Notebooks' #
!pip install chromedriver-autoinstaller

!pip install selenium
!apt-get update

# (최초 1회)
!apt install chromium-chromedriver
!cp /usr/lib/chromium-browser/chromedriver '/content/drive/MyDrive/Colab Notebooks' #
!pip install chromedriver-autoinstaller

from selenium import webdriver
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By
import sys
from selenium.webdriver.common.keys import Keys
import urllib.request
import os
from urllib.request import urlretrieve

import time
import pandas as pd
import chromedriver_autoinstaller  # setup chrome options

chrome_path = "/content/drive/MyDrive/Colab Notebooks/chromedriver"

sys.path.insert(0,chrome_path)
chrome_options = webdriver.ChromeOptions()
chrome_options.add_argument('--headless') # ensure GUI is off
chrome_options.add_argument('--no-sandbox')
chrome_options.add_argument('--disable-dev-shm-usage')  # set path to chromedriver as per your configuration
chrome_options.add_argument('lang=ko_KR') # 한국어

chromedriver_autoinstaller.install()  # set the target URL
```

```
Setting up libfuse3-3:amd64 (3.10.5-1build1) ...
Setting up snapd (2.63+22.04) ...
Created symlink /etc/systemd/system/multi-user.target.wants/snapd.apparmor.service → /lib/systemd/system/snapd.apparmor.service.
Created symlink /etc/systemd/system/multi-user.target.wants/snapd.autoimport.service → /lib/systemd/system/snapd.autoimport.service.
Created symlink /etc/systemd/system/multi-user.target.wants/snapd.core-fixup.service → /lib/systemd/system/snapd.core-fixup.service.
Created symlink /etc/systemd/system/multi-user.target.wants/snapd.recovery-chooser-trigger.service → /lib/systemd/system/snapd.recovery-choos
Created symlink /etc/systemd/system/multi-user.target.wants/snapd.seeded.service → /lib/systemd/system/snapd.seeded.service.
Created symlink /etc/systemd/system/cloud-final.service.wants/snapd.seeded.service → /lib/systemd/system/snapd.seeded.service.
Unit /lib/systemd/system/snapd.seeded.service is added as a dependency to a non-existent unit cloud-final.service.
Created symlink /etc/systemd/system/multi-user.target.wants/snapd.service → /lib/systemd/system/snapd.service.
Created symlink /etc/systemd/system/timers.target.wants/snapd.snap-repair.timer → /lib/systemd/system/snapd.snap-repair.timer.
Created symlink /etc/systemd/system/sockets.target.wants/snapd.socket → /lib/systemd/system/snapd.socket.
Created symlink /etc/systemd/system/final.target.wants/snapd.system-shutdown.service → /lib/systemd/system/snapd.system-shutdown.service.
Selecting previously unselected package chromium-browser.
(Reading database ... 124026 files and directories currently installed.)
Preparing to unpack .../chromium-browser_1%3a85.0.4183.83-0ubuntu2.22.04.1_amd64.deb ...
=> Installing the chromium snap
==> Checking connectivity with the snap store
===> System doesn't have a working snapd, skipping
Unpacking chromium-browser (1:85.0.4183.83-0ubuntu2.22.04.1) ...
Selecting previously unselected package chromium-chromedriver.
Preparing to unpack .../chromium-chromedriver_1%3a85.0.4183.83-0ubuntu2.22.04.1_amd64.deb ...
Unpacking chromium-chromedriver (1:85.0.4183.83-0ubuntu2.22.04.1) ...
Selecting previously unselected package systemd-hwe-hwdb.
Preparing to unpack .../systemd-hwe-hwdb_249.11.5_all.deb ...
Unpacking systemd-hwe-hwdb (249.11.5) ...
Setting up systemd-hwe-hwdb (249.11.5) ...
Setting up chromium-browser (1:85.0.4183.83-0ubuntu2.22.04.1) ...
update-alternatives: using /usr/bin/chromium-browser to provide /usr/bin/x-www-browser (x-www-browser) in auto mode
update-alternatives: using /usr/bin/chromium-browser to provide /usr/bin/gnome-www-browser (gnome-www-browser) in auto mode
Setting up chromium-chromedriver (1:85.0.4183.83-0ubuntu2.22.04.1) ...
Processing triggers for udev (249.11-0ubuntu3.12) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for libc-bin (2.35-0ubuntu3.4) ...
/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_adapter_opencl.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind.so.3 is not a symbolic link
```

스타벅스의 지역별 매장 검색 화면에 접속

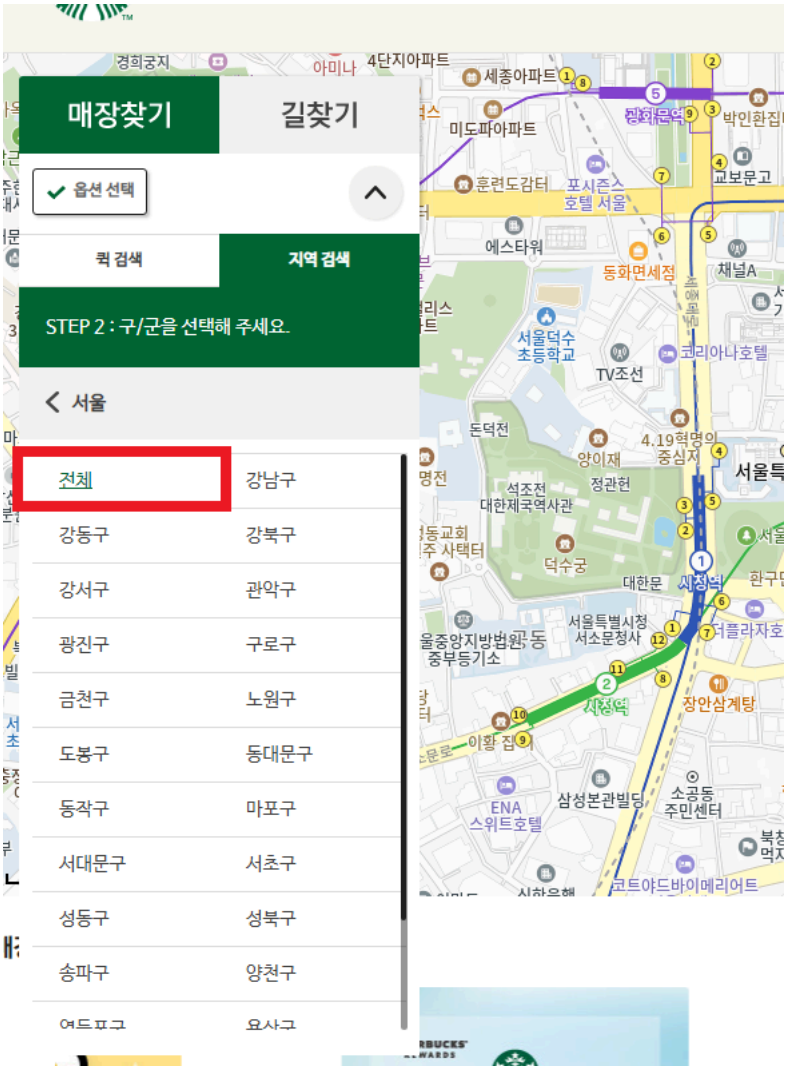
```
driver = webdriver.Chrome(options=chrome_options)
url='https://www.starbucks.co.kr/store/store_map.do?disp=locale'
driver.get(url)

# webdriver의 Chrome()함수를 이용해 크롬 브라우저를 실행하고, browser 변수에 할당.
# driver.get(url)명령어를 통해 해당 url로 이동.
```

webdriver로 '서울'버튼 요소를 찾아 클릭

```
seoul_btn='#container > div > form > fieldset > div > section > article.find_store_cont > article > article:nth-child(4) > div.loca_step1 > div.loca_st
driver.find_element(By.CSS_SELECTOR,seoul_btn).click()
```

```
# '서울' 버튼을 클릭하면, 서울시의 구/군 목록이 나타나고 구/군별 스타벅스 매장을 조회할 수 있는 화면이 나타납니다.
# 이번에는 '전체'버튼을 클릭해 서울시 전체 매장을 조회하겠습니다.
```



webdriver '전체'버튼 요소를 찾아 클릭

```
all_btn='#mCSB_2_container > ul > li:nth-child(1) > a'
driver.find_element(By.CSS_SELECTOR,all_btn).click()
```

BeautifulSoup으로 HTML 파서 만들기

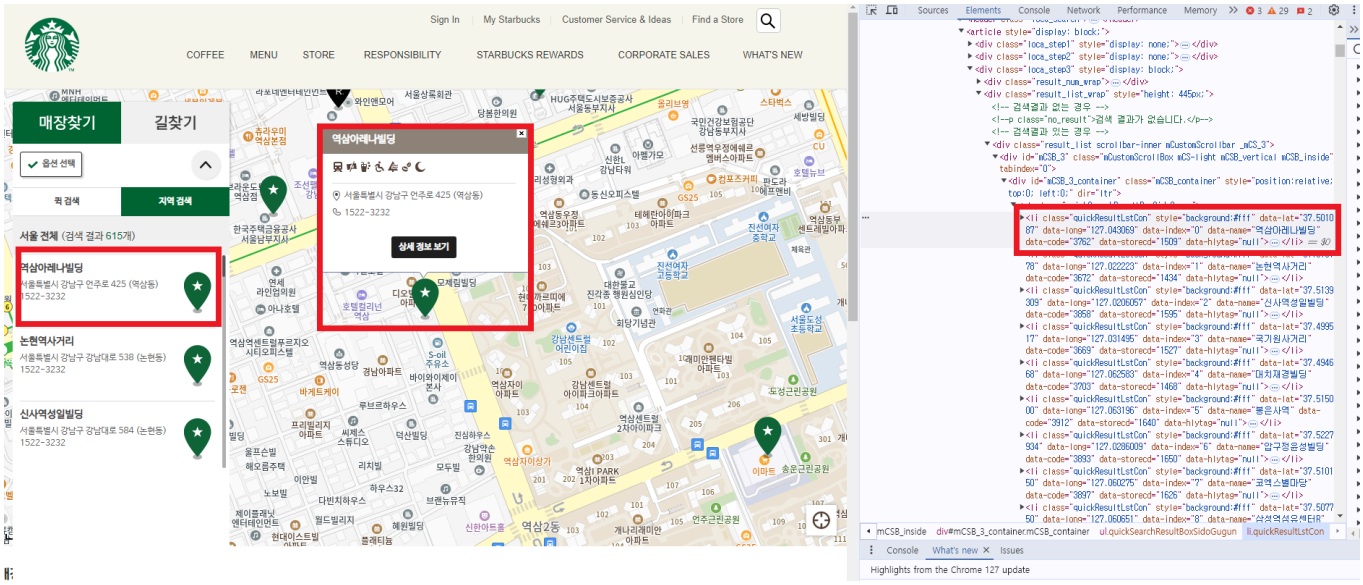
```
html=driver.page_source
# driver.page_source: 크롬 브라우저의 현재 화면에 나타난 웹페이지의 HTML을 가져올 수 있습니다.

soup=BeautifulSoup(html, 'html.parser')
# html.parser: HTML 문법을 이해하고, 웹 페이지의 정보를 분류하는 역할.
# HTML 문서를 구조적으로 분석하여 프로그래밍적으로 접근하고 조작할 수 있도록 돕는 것.

# driver.page_source: 웹페이지의 HTML 소스를 가져온다.
# BeautifulSoup 라이브러리: 그 HTML 소스를 라이브러리와 'html.parser'를 통해 분석하여 프로그래밍적으로 접근 가능한 구조로 변경.
# 이 과정을 통해, 웹 페이지의 내용을 더 쉽게 추출하고 조작할 수 있게 됩니다.
```

```
html
```

```
'<html lang="ko"><head>₩n₩t₩t₩n₩n₩n₩n₩n₩n₩n₩n₩n<meta http-equiv="X-UA-Compatible" content="IE=edge">₩n<meta charset="utf-8">₩n<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0, user-scalable=no">₩n<meta property="og:type" content="website">₩n<meta property="og:title" content="Starbucks">₩n<meta property="og:url" content="https://www.starbucks.co.kr/">₩n<meta property="og:image" content="https://image.istarbucks.co.kr/common/img/kakaotalk.png">₩n<meta property="og:description" content="Starbucks">₩n₩n<title id="titleJoin">Starbucks Korea</title><script src="https://connect.facebook.net/ko_KR/sdk.js?hash=a01ff6e51bced808ecfdbdc574e2bcfa" async="" crossorigin="anonymous"></script><script id="facebook-jssdk" src="//connect.facebook.net/ko_KR/sdk.js"></script><script type="text/javascript" async="" src="https://www.googletagmanager.com/gtag/js?id=G-WC8Q3C59QP&amp;cx=c&amp;_slc=1"></script><script async="" src="//www.google-analytics.c
```



매장 리스트에서 1번째 '역삼역 아레나 빌딩'에 해당하는 태그를 확인해보면,
<li class="quickResultLstCon"... data-name="역삼아레나빌딩"..>
인 것을 확인할 수 있습니다.

스타벅스 매장 목록에서 개별 매장 정보(지점명, 주소 등)는 각각 다르지만, HTML 상에서 동일한 구조의 태그로 표시된다는 사실을 알 수 있습니다.

<li class="quickResultLstCon" style="background:#fff" data-lat="37.501087" data-long="127.043069" data-index="0" data-name="역삼아레나빌딩" data-code

select()를 이용해 원하는 HTML 태그를 모두 찾아오기

```
starbucks_soup_list=soup.select('li.quickResultLstCon')
print(len(starbucks_soup_list))
```

soup.select('li.quickResultLstCon): soup안에서 태그명이 li이면서, class명이 quickResultLstCon인 태그를 모두 찾아 리스트 형태로 저장.

```
615
```

태그 문자열 살펴보기

```
starbucks_soup_list[0]
```

태그는 class를 비롯한 여러 속성값을 가지고 있는데, 대표적으로 data-lat(37.501087), data-long(127.043069) 속성은 각각 매장의 위도, 경도 데이터를

```
<li class="quickResultLstCon" data-code="3762" data-hlytag="null" data-index="0" data-lat="37.501087" data-long="127.043069" data-name="역삼아레
나빌딩" data-storecd="1509" style="background:#fff"> <strong data-my_siren_order_store_yn="N" data-name="역삼아레나빌딩" data-store="1509" data-
yn="N">역삼아레나빌딩  </strong> <p class="result_details">서울특별시 강남구 언주로 425 (역삼동)<br/>1522-3232</p> <i class="pin_general">리저브
매장 2번</i></li>
```

역삼아레나빌딩
태그는 '역삼아레나빌딩'이라는 텍스트와 이미지를 표시하는 <img>태그를 포함하고 있다.

해당 내용에는 img값이 없다. 책의 경우, 역삼아레나 빌딩 옆에 별 그림이 있지만, 현재는 없기 때문에.

서울특별시 강남구 언주로 425 (역삼동)

1522-3232

태그 사이에는 주소와 전화번호가 표시돼 있습니다.

리저브 매장 2번

태그는 class의 속성값으로 pin_general을 가지고 있습니다.
아이콘은 스타벅스 매장 타입에 따라 클래스명이 다르게 표시되고, 아이콘 이미지도 달라집니다. 이 값을 통해, 스타벅스 매장 타입을 분류할 수 있습니다.

스타벅스 매장 정보 샘플 확인

```
starbucks_store=starbucks_soup_list[0]
name=starbucks_store.select('strong')[0].text.strip()
print('매장명:',name)

lat=starbucks_store['data-lat'].strip()
print('위도:',lat)

lng=starbucks_store['data-long'].strip()
print('경도:',lng)

store_type=starbucks_store.select('i')[0]['class'][0][4:]
print('매장타입:',store_type)

store_type1=starbucks_store.select('i')[0]['class'][0]
print('매장타입:',store_type1)

address=str(starbucks_store.select('p.result_details')[0]).split('<br/>')[0].split('>')[1]
print('주소:',address)

tel=str(starbucks_store.select('p.result_details')[0]).split('<br/>')[1].split('<')[0]
print('전화번호:',tel)
```

```
매장명: 역삼아레나빌딩
위도: 37.501087
경도: 127.043069
매장타입: general
매장타입: pin_general
주소: 서울특별시 강남구 언주로 425 (역삼동)
전화번호: 1522-3232
```

```
str(starbucks_store.select('p.result_details')[0]).split('<br/>')[0]
```

```
'<p class="result_details">서울특별시 강남구 언주로 425 (역삼동)'
```

```
str(starbucks_store.select('p.result_details')[0]).split('<br/>')[0].split('>')
```

```
['<p class="result_details"', '서울특별시 강남구 언주로 425 (역삼동)']
```

```
str(starbucks_store.select('p.result_details')[0]).split('<br/>')[1]
```

```
'1522-3232</p>'
```

서울시 스타벅스 매장 목록 데이터 만들기

```
starbucks_list=[]
for item in starbucks_soup_list:
  print('item값:',item)
  name=item.select('strong')[0].text.strip();
  lat=item['data-lat'].strip();
  lng=item['data-long'].strip();
  store_type=item.select('i')[0]['class'][0][4:]
  address=str(item.select('p.result_details')[0]).split('<br/>')[0].split('>')[1]
  tel=str(item.select('p.result_details')[0]).split('<br/>')[1].split('<')[0]

  starbucks_list.append([name, lat, lng, store_type, address, tel])
```

```
item값: <li class="quickResultLstCon" data-code="4237" data-hlytag="null" data-index="580" data-lat="37.544462083045" data-long="127.05541331
item값: <li class="quickResultLstCon" data-code="4136" data-hlytag="null" data-index="581" data-lat="37.5430298" data-long="127.0133661" data-
item값: <li class="quickResultLstCon" data-code="3959" data-hlytag="null" data-index="582" data-lat="37.547326" data-long="127.066367" data-na
item값: <li class="quickResultLstCon" data-code="3837" data-hlytag="null" data-index="583" data-lat="37.539305" data-long="126.963060" data-na
item값: <li class="quickResultLstCon" data-code="3918" data-hlytag="null" data-index="584" data-lat="37.555165" data-long="126.968446" data-na
item값: <li class="quickResultLstCon" data-code="3584" data-hlytag="null" data-index="585" data-lat="37.534273" data-long="126.994789" data-na
item값: <li class="quickResultLstCon" data-code="3556" data-hlytag="null" data-index="586" data-lat="37.537131" data-long="127.00098" data-nam
item값: <li class="quickResultLstCon" data-code="3550" data-hlytag="null" data-index="587" data-lat="37.527586" data-long="126.965401" data-na
item값: <li class="quickResultLstCon" data-code="3518" data-hlytag="null" data-index="588" data-lat="37.518605" data-long="126.977271" data-na
item값: <li class="quickResultLstCon" data-code="3487" data-hlytag="null" data-index="589" data-lat="37.528702" data-long="126.966757" data-na
item값: <li class="quickResultLstCon" data-code="3386" data-hlytag="null" data-index="590" data-lat="37.523818" data-long="126.970389" data-na
item값: <li class="quickResultLstCon" data-code="3321" data-hlytag="null" data-index="591" data-lat="37.542831" data-long="126.972242" data-na
item값: <li class="quickResultLstCon" data-code="3288" data-hlytag="null" data-index="592" data-lat="37.53485" data-long="127.010919" data-nam
item값: <li class="quickResultLstCon" data-code="3246" data-hlytag="null" data-index="593" data-lat="37.551102" data-long="126.988317" data-na
item값: <li class="quickResultLstCon" data-code="9934" data-hlytag="null" data-index="594" data-lat="37.551752" data-long="126.972694" data-na
item값: <li class="quickResultLstCon" data-code="9800" data-hlytag="null" data-index="595" data-lat="37.533394" data-long="127.00563" data-nam
item값: <li class="quickResultLstCon" data-code="9391" data-hlytag="null" data-index="596" data-lat="37.54460479" data-long="126.9672252" data
item값: <li class="quickResultLstCon" data-code="9462" data-hlytag="null" data-index="597" data-lat="37.5289857" data-long="126.991782" data-n
item값: <li class="quickResultLstCon" data-code="9577" data-hlytag="null" data-index="598" data-lat="37.544225782" data-long="126.971957017" d
item값: <li class="quickResultLstCon" data-code="9307" data-hlytag="null" data-index="599" data-lat="37.5289068079414" data-long="126.96463454
item값: <li class="quickResultLstCon" data-code="9568" data-hlytag="null" data-index="600" data-lat="37.52111" data-long="126.96905" data-name
item값: <li class="quickResultLstCon" data-code="4341" data-hlytag="null" data-index="601" data-lat="37.54474797" data-long="126.96484196" dat
item값: <li class="quickResultLstCon" data-code="4002" data-hlytag="null" data-index="602" data-lat="37.5265379583476" data-long="126.96736209
item값: <li class="quickResultLstCon" data-code="4232" data-hlytag="null" data-index="603" data-lat="37.53854" data-long="126.98737" data-name
item값: <li class="quickResultLstCon" data-code="4080" data-hlytag="null" data-index="604" data-lat="37.55415699928705" data-long="126.9717936
item값: <li class="quickResultLstCon" data-code="3891" data-hlytag="null" data-index="605" data-lat="37.538571" data-long="126.967409" data-na
item값: <li class="quickResultLstCon" data-code="4185" data-hlytag="null" data-index="606" data-lat="37.53286105987034" data-long="126.9610009
item값: <li class="quickResultLstCon" data-code="3223" data-hlytag="null" data-index="607" data-lat="37.597842" data-long="127.092509" data-na
item값: <li class="quickResultLstCon" data-code="3135" data-hlytag="null" data-index="608" data-lat="37.5930326" data-long="127.07473579999998
item값: <li class="quickResultLstCon" data-code="3087" data-hlytag="null" data-index="609" data-lat="37.60538908" data-long="127.0957558" data
item값: <li class="quickResultLstCon" data-code="3023" data-hlytag="null" data-index="610" data-lat="37.579594" data-long="127.087966" data-na
item값: <li class="quickResultLstCon" data-code="9686" data-hlytag="null" data-index="611" data-lat="37.59689" data-long="127.08647" data-name
item값: <li class="quickResultLstCon" data-code="3936" data-hlytag="null" data-index="612" data-lat="37.615368" data-long="127.076633" data-na
item값: <li class="quickResultLstCon" data-code="4307" data-hlytag="null" data-index="613" data-lat="37.6066536267232" data-long="127.10635979
item값: <li class="quickResultLstCon" data-code="3801" data-hlytag="null" data-index="614" data-lat="37.60170912407773" data-long="127.0784113
```

```
starbucks_list
```

```
 '1522-3232'],
['중화역',
 '37.60170912407773',
 '127.07841136432036',
 'general',
 '서울특별시 중랑구 봉화산로 35 1층',
 '1522-3232']]
```

pandas의 데이터 프레임 생성

```
columns=['매장명','위도','경도','매장타입','주소','전화번호']
seoul_starbucks_df=pd.DataFrame(starbucks_list, columns=columns)
# columns 변수에 칼럼명을 순서대로 입력.

seoul_starbucks_df.head()
```

	매장명	위도	경도	매장타입	주소	전화번호
0	역삼아레나빌딩	37.501087	127.043069	general	서울특별시 강남구 언주로 425 (역삼동)	1522-3232
1	논현역사거리	37.510178	127.022223	general	서울특별시 강남구 강남대로 538 (논현동)	1522-3232
2	신사역성일빌딩	37.5139309	127.0206057	general	서울특별시 강남구 강남대로 584 (논현동)	1522-3232
3	국기원사거리	37.499517	127.031495	general	서울특별시 강남구 테헤란로 125 (역삼동)	1522-3232
4	대치재경빌딩	37.494668	127.062583	general	서울특별시 강남구 남부순환로 2947 (대치동)	1522-3232

다음 단계: seoul_starbucks_df변수로 코드 생성 | 추천 차트 보기 | New interactive sheet

데이터프레임의 요약 정보 확인

```
seoul_starbucks_df.info()
# 데이터프레임에 615개에 달하는 매장 정보가 저장돼 있고 모든 칼럼에 null인 값이 없습니다.
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 615 entries, 0 to 614
Data columns (total 6 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   매장명     615 non-null    object
 1   위도      615 non-null    object
 2   경도      615 non-null    object
 3   매장타입    615 non-null    object
 4   주소      615 non-null    object
 5   전화번호    615 non-null    object
dtypes: object(6)
memory usage: 29.0+ KB
```

엑셀로 저장

```
seoul_starbucks_df.to_excel('seoul_starbucks_list.xlsx',index=False)
```

서울열린데이터 광장의 OPEN API를 활용한 공공데이터 수집

```
# 서울 열린데이터 광장: https://data.seoul.go.kr/

# 시민들에게 서울 행정 구역 내의 다양한 공공 데이터를 제공하고 있습니다.
# 사용자들은 해당 사이트에서 데이터를 내려받거나 Open API를 통해 획득할 수 있습니다.


# 서울열린데이터광장의 OPEN API를 사용하려면 사이트에 가입하고 인증키를 발급받아야 합니다.
# 인증키가 없으면 OPEN API를 사용할 수 없습니다.
```


```
# 위의 사진처럼 신청이 완료되면, [마이페이지]에서 [인증키 관리]를 클릭하고 인증키를 확인할 수 있습니다.
# 이 인증키는 서울열린데이터광장의 OPEN API를 호출할 때 항상 함께 보내야 하는데, 서울시열린데이터광장에서는 약간의
# 제한을 두고 있습니다.
```

서울열린데이터광장 OPEN API 호출하기 실습(서울시 시군구 목록 데이터)

```
# API의 URL형식: http://openapi.seoul.go.kr:8088/{KEY}/{TYPE}/{SERVICE}/{START_INDEX}/{END_INDEX}/{OBJECTID}

# 요청인자는 필수 값과 선택 값으로 구분됩니다. 만약 API를 호출할 때 필수 값을 전달하지 않으면 에러가 발생하기 때문에
# 필수 값은 반드시 전달해야 합니다.

# {KEY}: 인증키 - 서울열린데이터광장에서 발급된 인증키
# {SERVICE}: 해당 API의 서비스명(SdeTlSccoSigw: 서울시 행정구역 시군구 정보에 해당하는 서비스명.)

# ex) 서울시 시군구 데이터의 1번데이터부터 25번 데이터까지 json파일 형식으로 얻고 싶다면
# http://openapi.seoul.go.kr:8088/{인증키}/json/SdeTlSccoSigw/1/25/
```

라이브러리 임포트

```
import requests
import pandas as pd
# OPEN API를 호출하기 위해 requests 라이브러리를 사용합니다.
```