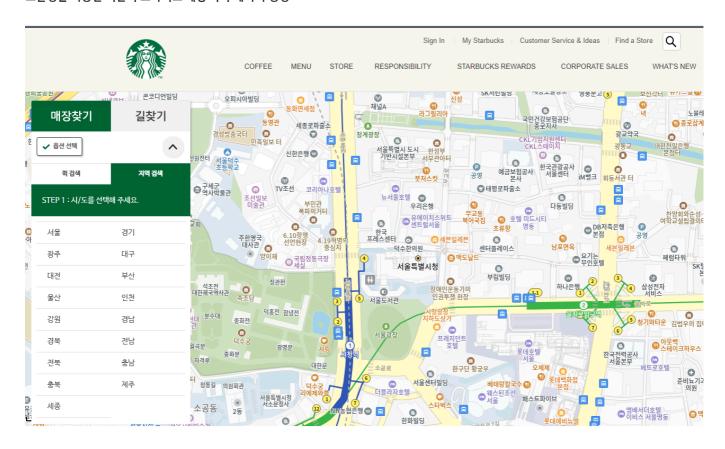# 5. 왜 우리 동네에는 스타벅스가 없을까?

데이터 수집

```
# 스타벅스 입지 가설
# 1) 거주 인구가 많은 지역에 스타벅스 매장이 많이 입지해 있을 것이다.
# 2) 직장인이 많은 지역에 스타벅스 많은 입지해 있을 것이다.

# 두 가지 가설을 검증하기 위해서 인구통계 데이터 수집도 필요합니다.
# 크롤링을 통해, 스타벅스 홈페이지에서 스타벅스 매장들의 정보를 수집하고,
# 서울시 열린 데이터 광장 Open API를 이용해 인구통계 데이터를 수집해 보겠습니다.
```

크롤링을 이용한 서울시 스타벅스 매장 목록 데이터 생성


```
# 스타벅스 매장 목록: https://www.starbucks.co.kr/index.do
# 페이지에서 왼쪽 메뉴의 [매장찾기]탭 하단의 [지역검색] 탭에서 지역별 매장 목록을 확인할 수 있습니다.
```

라이브러리 임포트

```
!pip install selenium
!apt-get update
!apt install chromium-chromedriver

from selenium import webdriver
from urllib.request import urlopen
from bs4 import BeautifulSoup as bs
from urllib.parse import quote_plus
from selenium.webdriver.common.keys import Keys
import time
```

```
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 ... 124027 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/libtbbbind_2_5.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_0.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc_proxy.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbb.so.12 is not a symbolic link

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

Processing triggers for man-db (2.10.2-1) ...
```

```python
from selenium import webdriver
# 크롤링을 실행하기 위해 selenium 라이브러리의 web driver를 불러왔습니다.

from bs4 import BeautifulSoup
# 크롤링으로 가져온 HTML에서 정보를 추출하기 위해 "BeautifulSoup" 라이브러리를 추가했습니다.

import pandas as pd
```

Selenium 설치 & 내 구글 드라이브에 chromedriver 설치

```python
!pip install selenium
!apt-get update
```

```
Requirement already satisfied: selenium in /usr/local/lib/python3.10/dist-packages (4.23.1)
Requirement already satisfied: urllib3<3,>=1.26 in /usr/local/lib/python3.10/dist-packages (from urllib3[socks]<3,>=1.26->selenium) (2.0.7)
Requirement already satisfied: trio~=0.17 in /usr/local/lib/python3.10/dist-packages (from selenium) (0.26.0)
Requirement already satisfied: trio-websocket~=0.9 in /usr/local/lib/python3.10/dist-packages (from selenium) (0.11.1)
Requirement already satisfied: certifi>=2021.10.8 in /usr/local/lib/python3.10/dist-packages (from selenium) (2024.7.4)
Requirement already satisfied: typing_extensions~=4.9 in /usr/local/lib/python3.10/dist-packages (from selenium) (4.12.2)
Requirement already satisfied: websocket-client~=1.8 in /usr/local/lib/python3.10/dist-packages (from selenium) (1.8.0)
Requirement already satisfied: attrs>=23.2.0 in /usr/local/lib/python3.10/dist-packages (from trio~=0.17->selenium) (23.2.0)
Requirement already satisfied: sortedcontainers in /usr/local/lib/python3.10/dist-packages (from trio~=0.17->selenium) (2.4.0)
Requirement already satisfied: idna in /usr/local/lib/python3.10/dist-packages (from trio~=0.17->selenium) (3.7)
Requirement already satisfied: outcome in /usr/local/lib/python3.10/dist-packages (from trio~=0.17->selenium) (1.3.0.post0)
Requirement already satisfied: sniffio>=1.3.0 in /usr/local/lib/python3.10/dist-packages (from trio~=0.17->selenium) (1.3.1)
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from trio~=0.17->selenium) (1.2.2)
Requirement already satisfied: wsproto>=0.14 in /usr/local/lib/python3.10/dist-packages (from trio-websocket~=0.9->selenium) (1.2.0)
Requirement already satisfied: pysocks!=1.5.7,<2.0,>=1.5.6 in /usr/local/lib/python3.10/dist-packages (from urllib3[socks]<3,>=1.26->selenium) (
Requirement already satisfied: h11<1,>=0.9.0 in /usr/local/lib/python3.10/dist-packages (from wsproto>=0.14->trio-websocket~=0.9->selenium) (0.14
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ InRelease
Hit:3 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64  InRelease
Hit:4 http://archive.ubuntu.com/ubuntu jammy InRelease
Ign:5 https://r2u.stat.illinois.edu/ubuntu jammy InRelease
Hit:6 https://r2u.stat.illinois.edu/ubuntu jammy Release
Hit:7 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:8 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:9 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy InRelease
Hit:10 https://ppa.launchpadcontent.net/graphics-drivers/ppa/ubuntu jammy InRelease
Hit:11 https://ppa.launchpadcontent.net/ubuntugis/ppa/ubuntu jammy InRelease
Reading package lists... Done
```
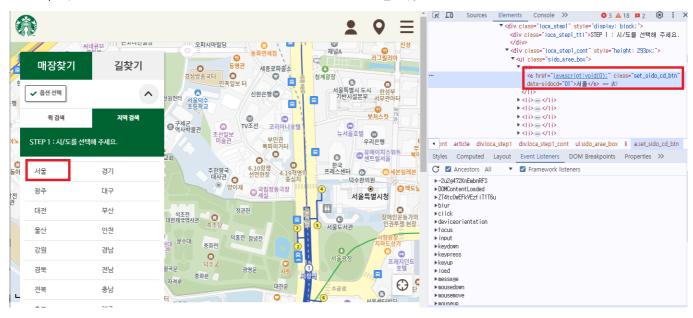
W: Skipping acquire of configured file 'main/source/Sources' as repository '<u>https://r2u.stat.illinois.edu/ubuntu</u> jammy InRelease' does not seem t

```
!python --version
```

```
import selenium
print(selenium.__version__)
```

> Python 3.10.12
> 4.31.1

라이브러리 임포트

```
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
from bs4 import BeautifulSoup

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

```
pip install chromedriver-autoinstaller
# 참고사이트: https://velog.io/@kite_day/colab-%EC%97%90%EC%84%9C-%EC%9B%B9-%ED%81%AC%EB%A1%A4%EB%A7%81%ED%95%98%EA%B8%B0-selenium
```

> Collecting chromedriver-autoinstaller
>   Downloading chromedriver_autoinstaller-0.6.4-py3-none-any.whl.metadata (2.1 kB)
> Requirement already satisfied: packaging>=23.1 in /usr/local/lib/python3.10/dist-packages (from chromedriver-autoinstaller) (24.1)
> Downloading chromedriver_autoinstaller-0.6.4-py3-none-any.whl (7.6 kB)
> Installing collected packages: chromedriver-autoinstaller
> Successfully installed chromedriver-autoinstaller-0.6.4

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

```
driver = webdriver.Chrome(options=chrome_options)
# webdriver의 Chrome()함수를 이용해 크롬 브라우저를 실행하고 browser변수에 할당합니다.
```

```
url='https://www.starbucks.co.kr/store/store_map.do?disp=locale'
driver.get(url)
# driver.get(url): 해당 url로 이동.
```

```
driver.implicitly_wait(3) # element가 로드될 때까지 지정한 시간만큼 대기할 수 있도록 설정
print(driver)
```

> <selenium.webdriver.chrome.webdriver.WebDriver (session="db42f00d4d7b33596f925b4e28b10975")>

```
# 크롬 브라우저에서 '서울'버튼의 위치를 얻기 위해 개발자 도구를 실행합니다.
# [메뉴]->[도구 더보기]->[개발자 도구]

# 개발자 도구 화면에 버튼의 HTML 요소인 <a> 태그가 선택되어 하이라이트 된 것을 확인할 수 있습니다.
# 하이라이트 표시된 태그 영역에 마우스 커서를 이동한 후, 마우스 오른쪽 버튼을 클릭하면 [선택] 메뉴가 나타나고, [copy] 메뉴에서
# [copy selector]를 클릭하면 전체 HTML 문서 내에서 '서울' 버튼에 해당하는 태그의 구조 정보가 복사됩니다.(CSS selector)
```

```python
# webdriver로 '서울' 버튼 요소를 찾아 클릭
# '서울' 클릭
seoul_btn='#container > div > form > fieldset > div > section > article.find_store_cont > article > article:nth-child(4) > div.loca_step1 > div.loca_
driver.find_element_by_css_selector(seoul_btn).click()
```

```
---------------------------------------------------------------------------
AttributeError                          Traceback (most recent call last)
<ipython-input-39-f47c6f0f9ef5> in <cell line: 4>()
      2 # '서울' 클릭
      3 seoul_btn='#container > div > form > fieldset > div > section >
article.find_store_cont > article > article:nth-child(4) > div.loca_step1 >
div.loca_step1_cont > ul > li:nth-child(1) > a'
----> 4 driver.find_element_by_css_selector(seoul_btn).click()

AttributeError: 'WebDriver' object has no attribute 'find_element_by_css_selector'
```

```python
from selenium.webdriver.common.by import By
seoul_btn='#container > div > form > fieldset > div > section > article.find_store_cont > article > article:nth-child(4) > div.loca_step1 > div.loca_

counts=driver.find_element(By.CSS_SELECTOR,seoul_btn)
counts.click() # 웹브라우저에서 특정 요소를 클릭한다.

# driver.find_element(By.CSS_SELECTOR) = seoul_btn에서 저장된 '서울'버튼의 위치를 선택하고, click() 명령으로 해당 버튼 클릭.
```

구/군 선택 화면

```python
# 전체'버튼'을 클릭해 서울시 전체 매장을 조회합니다.
# '서울' 버튼의 CSS 셀렉터를 복사한 것처럼 '전체' 버튼의 CSS 셀렉터를 복사합니다.

all_btn='#mCSB_2_container > ul > li:nth-child(1) > a'
counts=driver.find_element(By.CSS_SELECTOR,all_btn)
counts.click() # 웹브라우저에서 특정 요소를 클릭한다.


# beautifulsoup으로 html 파서 만들기
html=driver.page_source
# driver.page_source를 통해 크롬 브라우저의 현재 화면에 나타난 웹페이지의 HTML을 가져올 수 있습니다.

soup=BeautifulSoupu=(html, 'html.parser')
# html.parser는 HTML 문법을 이해하고, 웹 페이지의 정보를 분류하는 역할을 합니다.


html
```

```
'<html lang="ko"><head>\n\t\t\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-s
cale=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" c
ontent="https://www.starbucks.co.kr/">\n<meta property="og:image" content="https://image.ista
```

```python
soup
```

```
('<html lang="ko"><head>\n\t\t\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 type="text/javascript" async=""
```

```
src="https://www.googletagmanager.com/gtag/js?id=G-WC8Q3C59QP&amp;cx=c&amp;_slc=1"></script><script
src="https://connect.facebook.net/ko_KR/sdk.js?hash=43230cbf1d03f7107ae2216a90f016e9" async="" crossorigin="anonymous"></script><script
id="facebook-jssdk" src="//connect.facebook.net/ko_KR/sdk.js"></script><script async="" src="//www.google-analytics.com/analytics.js">
</script><script type="text/javascript" src="/common/js/esabsbuxkr.js?single"></script><script async="" type="text/javascript"
src="/common/js/esabsbuxkr.js?async" id="6d5004df9831ad748389f1f7b856e95c"></script> <!-- 220117 수정 -->\n<link rel="shortcut icon"
href="https://image.istarbucks.co.kr/common/img/common/favicon.ico?v=200828" type="image/ico"> <!-- 20200827 파비콘 교체 및 CDN 변수처리 --
>\n<link href="/common/css/reset.css" rel="stylesheet">\n<link href="/common/css/style.css?v=230809" rel="stylesheet">\n<link
href="/common/css/jquery.bxslider.css" rel="stylesheet">\n<link href="/common/css/idangerous.swiper.css" rel="stylesheet">\n<link
href="/common/css/idangerous.swiper.scrollbar.css" rel="stylesheet">\n<link href="/common/css/jquery.mCustomScrollbar.css"
rel="stylesheet">\n<link href="/common/css/jquery.scrollbar.css" rel="stylesheet">\n<link href="/common/css/jquery-ui.css"
rel="stylesheet">\n<link href="/common/css/ezmark.css" rel="stylesheet">\n<link href="/common/css/style_dt.css?v=240306" rel="stylesheet">
\n<!--[if lt IE 9]>\n<script src="//html5shiv.googlecode.com/svn/trunk/html5.js"></script>\n<link href="/common/css/style_ie.css"
rel="stylesheet">\n\t<script>\n\t\tdocument.createElement('header');\n\t\tdocument.createElement('nav');\n\t\tdocument.createElement('sec
[endif]-->\n\n<script>\n   (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){\n   (i[r].q=i[r].q||
[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),\n   m=s.getElementsByTagName(o)
[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)\n   })(window,document,'script','//www.google-
analytics.com/analytics.js','ga');\n\n   ga('create', 'UA-66158138-1', 'auto');\n   ga('send',
'pageview');\n</script>\n\n\n<script type="text/javascript">\nvar eFrequency = 'N';\nvar eFrequencySeq = '240';\nvar
eFrequencyPlannerYn = 'N';\n</script>\n\n\t<link href="/common/css/style_main.css?r=1abii2" rel="stylesheet">\n\t<link
href="/common/css/style_whatsnew.css?r=1abii2" rel="stylesheet">\n\t<link href="/common/css/style_find_road.css?v=240125"
rel="stylesheet">\n\t<link href="/common/css/style_util.css?v=240319" rel="stylesheet">\n\t<link href="/common/css/skdslider.css?
r=1abii2" rel="stylesheet">\n\t<link href="/common/css/ezmark.css?r=1abii2" rel="stylesheet">\n\t<style type="text/css" data-
fbcssmodules="css:fb.css.base css:fb.css.dialog css:fb.css.iframewidget
css:fb.css.customer_chat_plugin_iframe">.fb_hidden{position:absolute;top:-10000px;z-
index:10001}.fb_reposition{overflow:hidden;position:relative}.fb_invisible{display:none}.fb_reset{background:none;border:0;border-
spacing:0;color:#000;cursor:auto;direction:ltr;font-family:'lucida grande', tahoma, verdana, arial, sans-serif;font-size:11px;font-
style:normal;font-variant:normal;font-weight:normal;letter-spacing:normal;line-height:1;margin:0;overflow:visible;padding:0;text-
align:left;text-decoration:none;text-indent:0;text-shadow:none;text-transform:none;visibility:visible;white-space:normal;word-
spacing:normal}.fb_reset>div{overflow:hidden}@keyframes
fb_transform{from{opacity:0;transform:scale(.95)}to{opacity:1;transform:scale(1)}}.fb_animate{animation:fb_transform .3s
forwards}\n.fb_hidden{position:absolute;top:-10000px;z-
index:10001}.fb_reposition{overflow:hidden;position:relative}.fb_invisible{display:none}.fb_reset{background:none;border:0;border-
spacing:0;color:#000;cursor:auto;direction:ltr;font-family:'lucida grande', tahoma, verdana, arial, sans-serif;font-size:11px;font-
style:normal;font-variant:normal;font-weight:normal;letter-spacing:normal;line-height:1;margin:0;overflow:visible;padding:0;text-
align:left;text-decoration:none;text-indent:0;text-shadow:none;text-transform:none;visibility:visible;white-space:normal;word-
spacing:normal}.fb_reset>div{overflow:hidden}@keyframes
fb_transform{from{opacity:0;transform:scale(.95)}to{opacity:1;transform:scale(1)}}.fb_animate{animation:fb_transform .3s
forwards}\n.fb_dialog{background:rgba(82, 82, 82, .7);position:absolute;top:-10000px;z-index:10001}.fb_dialog_advanced{border-
radius:8px;padding:10px}.fb_dialog_content{background:#fff;color:#373737}.fb_dialog_close_icon{background:url(https://connect.facebook.net/rsr
no-repeat scroll 0 0 transparent;cursor:pointer;display:block;height:15px;position:absolute;right:18px;top:17px;width:15px}.fb_dialog_mobile
.fb_dialog_close_icon{left:5px;right:auto;top:5px}.fb_dialog_padding{background-color:transparent;position:absolute;width:1px;z-
index:-1}.fb_dialog_close_icon:hover{background:url(https://connect.facebook.net/rsrc.php/v3/yg/r/IE9JII6Z1Ys.png) no-repeat scroll 0 -15px
transparent}.fb_dialog_close_icon:active{background:url(https://connect.facebook.net/rsrc.php/v3/yg/r/IE9JII6Z1Ys.png) no-repeat scroll 0
-30px transparent}.fb_dialog_iframe{line-height:0}.fb_dialog_content .dialog_title{background:#6d84b4;border:1px solid
#365899;color:#fff;font-size:14px;font-weight:bold;margin:0}.fb_dialog_content
.dialog_title>span{background:url(https://connect.facebook.net/rsrc.php/v3/yd/r/Cou7n-nqK52.gif) no-repeat 5px 50%;float:left;padding:5px 0
7px
26px}body.fb_hidden{height:100%;left:0;margin:0;overflow:visible;position:absolute;top:-10000px;transform:none;width:100%}.fb_dialog.fb_dialog
white no-repeat 50% 50%;min-height:100%;min-width:100%;overflow:hidden;position:absolute;top:0;z-
index:10001}.fb_dialog.fb_dialog_mobile.loading.centered{background:none;height:auto;min-height:initial;min-
width:initial;width:auto}.fb_dialog.fb_dialog_mobile.loading.centered
#fb_dialog_loader_spinner{width:100%}.fb_dialog.fb_dialog_mobile.loading.centered .fb_dialog_content{background:none}.loading.centered
```



```python
# 매장 리스트에서 첫 번째 '역삼역 아레나 빌딩'에 해당하는 태그 확인
# 스타벅스 매장 목록에서 개별 매장 정보(지점명, 주소 등)는 각각 다르지만, HTML 상에서 동일한 구조의 태그로 표시된다는 사실을 알 수 있다.
# HTML에서 <li>태그: 여러 항목을 리스트(목록)로 화면에 나타낼 때 사용하는 태그.


# select()를 이용해 원하는 HTML 태그를 모두 찾아오기
soup=BeautifulSoup(html, 'html.parser')

starbucks_soup_list=soup.select('li.quickResultLstCon')
# soup안에서 태그명이 li이면서 class명이 quickResultLstCon인 태그를 모두 찾아 리스트 형태로 저장합니다.

print(len(starbucks_soup_list))
```

```
615
```

```
# 태그 문자열 살펴보기
starbucks_soup_list[0]
```

아래를 보면 태그 li값이면서, class명이 quickResultLstCon값을 도출했음을 알 수 있다.

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

\<strong\> 태그: '역삼아레나빌딩'이라는 텍스트와 이미지를 표시하는 \<img\>태그 포함
\<p\> 태그 사이에는 주소와 전화번호가 표시돼 있습니다.
\<i\>: class의 속성값으로 pin_general을 가지고 있습니다. 이 아이콘은 스타벅스 매장 타입에 따라 클래스명이 다르게 표시되고, 아이콘 이미지도 달라집니다.
이 값을 통해, 스타벅스 매장 타입을 분류할 수 있습니다.