Spring Day 4

▼ Spring

Spring Security

인증 VS 인가

인증(authentication) : 회원 로그인, 회원/비회원, 멤버쉽

인가(authorization) : 1급 기밀, 접근 권한, 무료회원/유료회원

환경설정

pom.xml 설정

spring-security-web, spring-security-config 의존성 라이브러리 주입.
(Maven Repository 사용하기)

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.carshop</groupId>
  <artifactId>controller</artifactId>
  <name>CarShop</name>
  <packaging>war</packaging>
  <version>1.0.0-BUILD-SNAPSHOT</version>
  <properties>
    <java-version>1.6</java-version>
    <org.springframework-version>5.3.19</org.springframework-version>
    <org.aspectj-version>1.6.10</org.aspectj-version>
    <org.slf4j-version>1.6.6</org.slf4j-version>
  </properties>
  <dependencies>
  <!-- Spring -->
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>${org.springframework-version}</version>
    <exclusions>
      <!-- Exclude Commons Logging in favor of SLF4j -->
      <exclusion>
        <groupId>commons-logging</groupId>
        <artifactId>commons-logging</artifactId>
      </exclusion>
    </exclusions>
  </dependency>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>${org.springframework-version}</version>
  </dependency>

  <!-- AspectJ -->
  <dependency>
    <groupId>org.aspectj</groupId>
    <artifactId>aspectjrt</artifactId>
    <version>${org.aspectj-version}</version>
  </dependency>

  <!-- Logging -->
```

```
  <dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-api</artifactId>
    <version>${org.slf4j-version}</version>
  </dependency>
  <dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>jcl-over-slf4j</artifactId>
    <version>${org.slf4j-version}</version>
    <scope>runtime</scope>
  </dependency>
  <dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-log4j12</artifactId>
    <version>${org.slf4j-version}</version>
    <scope>runtime</scope>
  </dependency>
  <dependency>
    <groupId>log4j</groupId>
    <artifactId>log4j</artifactId>
    <version>1.2.15</version>
    <exclusions>
      <exclusion>
        <groupId>javax.mail</groupId>
        <artifactId>mail</artifactId>
      </exclusion>
      <exclusion>
        <groupId>javax.jms</groupId>
        <artifactId>jms</artifactId>
      </exclusion>
      <exclusion>
        <groupId>com.sun.jdmk</groupId>
        <artifactId>jmxtools</artifactId>
      </exclusion>
      <exclusion>
        <groupId>com.sun.jmx</groupId>
        <artifactId>jmxri</artifactId>
      </exclusion>
    </exclusions>
    <scope>runtime</scope>
  </dependency>

  <!-- @Inject -->
  <dependency>
    <groupId>javax.inject</groupId>
    <artifactId>javax.inject</artifactId>
    <version>1</version>
  </dependency>

  <!-- Servlet -->
  <dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>servlet-api</artifactId>
    <version>2.5</version>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>javax.servlet.jsp</groupId>
    <artifactId>jsp-api</artifactId>
    <version>2.1</version>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jstl</artifactId>
    <version>1.2</version>
  </dependency>

  <!-- Test -->
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.7</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-web</artifactId>
    <version>5.4.6</version>
  </dependency>
  <!-- https://mvnrepository.com/artifact/org.springframework.security/spring-security-config -->
  <dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-config</artifactId>
    <version>5.4.6</version>
  </dependency>
```

```
</dependencies>
<!-- https://mvnrepository.com/artifact/org.springframework.security/spring-security-web -->



  <build>
      <plugins>
          <plugin>
              <artifactId>maven-eclipse-plugin</artifactId>
              <version>2.9</version>
              <configuration>
                  <additionalProjectnatures>
                      <projectnature>org.springframework.ide.eclipse.core.springnature</projectnature>
                  </additionalProjectnatures>
                  <additionalBuildcommands>
                      <buildcommand>org.springframework.ide.eclipse.core.springbuilder</buildcommand>
                  </additionalBuildcommands>
                  <downloadSources>true</downloadSources>
                  <downloadJavadocs>true</downloadJavadocs>
              </configuration>
          </plugin>
          <plugin>
              <groupId>org.apache.maven.plugins</groupId>
              <artifactId>maven-compiler-plugin</artifactId>
              <version>2.5.1</version>
              <configuration>
                  <source>1.6</source>
                  <target>1.6</target>
                  <compilerArgument>-Xlint:all</compilerArgument>
                  <showWarnings>true</showWarnings>
                  <showDeprecation>true</showDeprecation>
              </configuration>
          </plugin>
          <plugin>
              <groupId>org.codehaus.mojo</groupId>
              <artifactId>exec-maven-plugin</artifactId>
              <version>1.2.1</version>
              <configuration>
                  <mainClass>org.test.int1.Main</mainClass>
              </configuration>
          </plugin>
      </plugins>
  </build>
</project>
```

web.xml 설정

filter 추가 : jsp 페이지를 유저에게 보내기 전에 필터링.

param 추가 : **security-context.xml** 설정(권한/인증 설정)

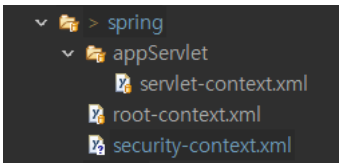
```
<filter>
<filter-name>springSecurityFilterChain</filter-name>
<filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
</filter>
    <filter-mapping>
    <filter-name>springSecurityFilterChain</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

```
<!-- The definition of the Root Spring Container shared by all Servlets and Filters -->
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/spring/root-context.xml
                 /WEB-INF/spring/security-context.xml
    </param-value>
</context-param>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee https://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">

  <!-- The definition of the Root Spring Container shared by all Servlets and Filters -->
  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/spring/root-context.xml
          /WEB-INF/spring/security-context.xml
    </param-value>
  </context-param>

  <!-- Creates the Spring Container shared by all Servlets and Filters -->
  <listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
  </listener>

  <!-- Processes application requests -->
  <servlet>
    <servlet-name>appServlet</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <init-param>
      <param-name>contextConfigLocation</param-name>
      <param-value>/WEB-INF/spring/appServlet/servlet-context.xml</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>

  <servlet-mapping>
    <servlet-name>appServlet</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>

  <filter>
    <filter-name>encodingFilter</filter-name>
    <filter-class>org.springframework.web.filter.CharacterEncodingFilter
    </filter-class>
    <init-param>
      <param-name>encoding</param-name>
      <param-value>UTF-8</param-value>
    </init-param>
  </filter>
  <filter-mapping>
    <filter-name>encodingFilter</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>

  <filter>
  <filter-name>springSecurityFilterChain</filter-name>
  <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
  </filter>
    <filter-mapping>
    <filter-name>springSecurityFilterChain</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>




</web-app>
```

security-context.xml 생성 & 설정

💡 **파일 위치를 잘 확인하자.**




```
<?xml version="1.0" encoding="UTF-8"?>
<beans:beans
  xmlns="http://www.springframework.org/schema/security"
  xmlns:beans="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
  http://www.springframework.org/schema/beans/spring-beans.xsd
  http://www.springframework.org/schema/security
  http://www.springframework.org/schema/security/spring-security.xsd">


  <http auto-config="true" use-expressions="true">

  <intercept-url pattern="/admin/add" access="hasAuthority('ROLE_ADMIN')" /> <!-- ADMIN만 허용 -->
  <intercept-url pattern="/manager" access="hasRole('ROLE_MANAGER')" /> <!-- MANAGER만 허용 -->
  <intercept-url pattern="/member" access="isAuthenticated()" /> <!-- 로그인한 사람만 허용 -->
  <intercept-url pattern="/**" access="permitAll" /> <!-- 전체 허용 -->



  </http>

  <authentication-manager>
    <authentication-provider>
    <user-service>

  <user name="admin" password="{noop}admin" authorities="ROLE_ADMIN, ROLE_USER" />
  <user name="manager" password="{noop}manager" authorities="ROLE_MANAGER, ROLE_USER" />
  <user name="guest" password="{noop}guest" authorities="ROLE_USER" />



    </user-service>
    </authentication-provider>
  </authentication-manager>
</beans:beans>
```

Custom Login

💡 **스프링에서는 기본적인 로그인과 로그아웃 기능을 제공한다.**
(Custom Login, Custom Logout)

CarController.java

```
    @GetMapping("/login")
    public String loginMethod() {
        return "login";
    }

    @GetMapping("//loginfailed")
    public String loginfailedMethod() {
        return "login";
    }

    @GetMapping("/logout")
    public String logoutMethod() {
        return "login";
    }
}
```

security-context.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans:beans
  xmlns="http://www.springframework.org/schema/security"
  xmlns:beans="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
  http://www.springframework.org/schema/beans/spring-beans.xsd
  http://www.springframework.org/schema/security
  http://www.springframework.org/schema/security/spring-security.xsd">


<!--  <http auto-config="true" use-expressions="true"> 기본 로그인 창 사용시 -->
  <http use-expressions="true">

  <intercept-url pattern="/admin/add" access="hasAuthority('ROLE_ADMIN')" /> <!-- ADMIN만 허용 -->
  <intercept-url pattern="/manager" access="hasRole('ROLE_MANAGER')" /> <!-- MANAGER만 허용 -->
  <intercept-url pattern="/member" access="isAuthenticated()" /> <!-- 로그인한 사람만 허용 -->
  <intercept-url pattern="/**" access="permitAll" /> <!-- 전체 허용 -->

  <form-login login-page="/login"
      default-target-url="/"
      authentication-failure-url="/loginfailed"
      username-parameter="username"
      password-parameter="password"/>

  <csrf/>

  <logout logout-success-url="/logout"/>

  </http>

  <authentication-manager>
    <authentication-provider>
    <user-service>

  <user name="admin" password="{noop}admin" authorities="ROLE_ADMIN, ROLE_USER" />
  <user name="manager" password="{noop}manager" authorities="ROLE_MANAGER, ROLE_USER" />
  <user name="guest" password="{noop}guest" authorities="ROLE_USER" />



    </user-service>
    </authentication-provider>
  </authentication-manager>
</beans:beans>
```

login.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ page session="false" %>
<html>
<head>
  <title>main</title>
  <%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
      <meta charset="utf-8">
```

```
</head>
<body class="text-center">
<%@ include file="header.jsp" %>
<!-- <div class="alert alert-dark" role="alert"> -->
<!--   <h1>로그인</h1> -->
<!-- </div> -->

<div class="container" style="width: 30%; padding-top: 200px;">
<main class="form-signin w-100 m-auto">
  <form action="/login" method="post">
    <img class="mb-4" src="
    <h1 class="h3 mb-3 fw-normal">로그인</h1>

    <div class="form-floating">
      <input type="text" class="form-control" id="username" name="username" placeholder="ID">
      <label for="floatingInput">ID</label>
    </div>
    <div class="form-floating">
      <input type="password" class="form-control" id="password" name="password" placeholder="Password">
      <label for="floatingPassword">Password</label>
    </div>

    <div class="checkbox mb-3">
      <label>
        <input type="checkbox" value="remember-me"> Remember me
      </label>
    </div>
    <button class="w-100 btn btn-lg btn-primary" type="submit">로그인</button>

    <input type="hidden" name="${_csrf.parameterName}" value="${_csrf.token}"/>
    <p class="mt-5 mb-3 text-muted">&copy; 2017–2022</p>

  </form>
</main>
</div>




<%@ include file="footer.jsp" %>
</body>
</html>
```

로그인 & 로그아웃 표시 설정

pom.xml

```
  <!-- https://mvnrepository.com/artifact/org.springframework.security/spring-security-taglibs -->
  <dependency>
      <groupId>org.springframework.security</groupId>
      <artifactId>spring-security-taglibs</artifactId>
      <version>5.4.6</version>
  </dependency>
```

header.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://www.springframework.org/security/tags" prefix="sec" %>
<html>
<head>
  <%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
      <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-rbsA2V
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KFdBIe4zVF0s
```

```
  <style>
    .bd-placeholder-img {
      font-size: 1.125rem;
      text-anchor: middle;
      -webkit-user-select: none;
      -moz-user-select: none;
      user-select: none;
    }

    @media ( min-width : 768px) {
      .bd-placeholder-img-lg {
        font-size: 3.5rem;
      }
    }

    .b-example-divider {
      height: 3rem;
      background-color: rgba(0, 0, 0, .1);
      border: solid rgba(0, 0, 0, .15);
      border-width: 1px 0;
      box-shadow: inset 0 .5em 1.5em rgba(0, 0, 0, .1), inset 0 .125em .5em
        rgba(0, 0, 0, .15);
    }

    .b-example-vr {
      flex-shrink: 0;
      width: 1.5rem;
      height: 100vh;
    }

    .bi {
      vertical-align: -.125em;
      fill: currentColor;
    }

    .nav-scroller {
      position: relative;
      z-index: 2;
      height: 2.75rem;
      overflow-y: hidden;
    }

    .nav-scroller .nav {
      display: flex;
      flex-wrap: nowrap;
      padding-bottom: 1rem;
      margin-top: -1px;
      overflow-x: auto;
      text-align: center;
      white-space: nowrap;
      -webkit-overflow-scrolling: touch;
    }
  </style>


</head>
<body class="text-center">
<nav class="navbar navbar-expand navbar-dark bg-dark fixed-top" aria-label="Second navbar example">
    <div class="container-fluid">
      <a class="navbar-brand" href="#">CarShop</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarsExample02" aria-controls="navb
        <span class="navbar-toggler-icon"></span>
      </button>

      <div class="collapse navbar-collapse" id="navbarsExample02">
        <ul class="navbar-nav me-auto">
         <li class="nav-item">
            <a class="nav-link active" aria-current="page" href="/">홈</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="/cars">차량보기</a>
          </li>
                   <li class="nav-item">
            <a class="nav-link" href="/board">게시판</a>
          </li>
                  <li class="nav-item dropdown">
          <button class="btn btn-dark dropdown-toggle" data-bs-toggle="dropdown" aria-expanded="false">
            회원관리
          </button>
          <ul class="dropdown-menu dropdown-menu-dark">
            <li><a class="dropdown-item" href="#">로그인</a></li>
            <li><a class="dropdown-item" href="#">로그아웃</a></li>
          </ul>
        </li>

        </ul>
        <sec:authorize access="isAnonymous()">
```

```
                <form action="/login" method="post">
         <input type="submit" class="btn btn-primary" value="login">
            <input type="hidden" name="${_csrf.parameterName}" value="${_csrf.token}"/>
        </form>
        </sec:authorize>

        <sec:authorize access="isAuthenticated()">
                <form action="/logout" method="post">
         <input type="submit" class="btn btn-success" value="logout">
            <input type="hidden" name="${_csrf.parameterName}" value="${_csrf.token}"/>
        </form>
        </sec:authorize>




      </div>
    </div>
  </nav>
```

▼ Python

영화 순위 크롤링

```
import requests
from selenium import webdriver      # 웹 브라우저 컨트롤(크롬)
from bs4 import BeautifulSoup as bs # 데이터 분석을 용이하게 정제
import pandas as pd
import time
from selenium.webdriver.common.by import By
from pprint import pprint

driver = webdriver.Chrome('chromedriver.exe')   # 버전 주의
url = 'https://movie.naver.com/movie/sdb/rank/rmovie.naver'
driver.get(url)

time.sleep(3)   # 3초간 대기

txt = driver.page_source    # 이 때 읽어온 데이터는 그냥 글자
html = bs(txt)



area_btn = '#old_content > div.tab_type_6 > ul > li:nth-child(2) > a > img'
driver.find_element(By.CSS_SELECTOR , area_btn).click()

time.sleep(2)

txt = driver.page_source    # 이 때 읽어온 데이터는 그냥 글자
html = bs(txt)

time.sleep(2)

titles = html.select('tbody > tr > td.title')
points = html.select('tbody > tr > td.point')

print(titles[0])

title_data = []

for i in range(len(titles)):
    subject = titles[i].text.strip()
    score = points[i].text.strip()
    title_data.append([subject, score])

df = pd.DataFrame(title_data, columns= ['제목', '평점'])
df

# df.to_excel('Genie(김도영).xlsx', index=False)
```

데이터 분석(스타벅스)

```
import requests
from selenium import webdriver      # 웹 브라우저 컨트롤(크롬)
from bs4 import BeautifulSoup as bs # 데이터 분석을 용이하게 정제
import pandas as pd
import time
from selenium.webdriver.common.by import By
from pprint import pprint

driver = webdriver.Chrome('chromedriver.exe')   # 버전 주의
url = 'https://movie.naver.com/movie/sdb/rank/rmovie.naver'
driver.get(url)

time.sleep(3)   # 3초간 대기

txt = driver.page_source    # 이 때 읽어온 데이터는 그냥 글자
html = bs(txt)



area_btn = '#old_content > div.tab_type_6 > ul > li:nth-child(2) > a > img'
driver.find_element(By.CSS_SELECTOR , area_btn).click()

time.sleep(2)

txt = driver.page_source    # 이 때 읽어온 데이터는 그냥 글자
html = bs(txt)

time.sleep(2)

titles = html.select('tbody > tr > td.title')
points = html.select('tbody > tr > td.point')

print(titles[0])

title_data = []

for i in range(len(titles)):
    subject = titles[i].text.strip()
    score = points[i].text.strip()
    title_data.append([subject, score])

df = pd.DataFrame(title_data, columns= ['제목', '평점'])
df

# df.to_excel('Genie(김도영).xlsx', index=False)
```

```
##########################################
#              스타벅스 위치               #
##########################################

import requests
from selenium import webdriver      # 웹 브라우저 컨트롤(크롬)
from bs4 import BeautifulSoup as bs # 데이터 분석을 용이하게 정제
import pandas as pd
import time
from selenium.webdriver.common.by import By

driver = webdriver.Chrome('chromedriver.exe')   # 버전 주의
url = 'https://www.starbucks.co.kr/store/store_map.do'
driver.get(url)

time.sleep(3)   # 3초간 대기

txt = driver.page_source    # 이 때 읽어온 데이터는 그냥 글자
html = bs(txt)

# 파이썬으로 브라우저에서 단추 클릭하기


area_btn = '#container > div > form > fieldset > div > section > article.find_store_cont > article > header.loca_search > h3 > a'
driver.find_element(By.CSS_SELECTOR , area_btn).click()

time.sleep(2)
```

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

time.sleep(2)

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

time.sleep(3)

txt = driver.page_source    # 이 때 읽어온 데이터는 그냥 글자
html = bs(txt)

time.sleep(3)

from pprint import pprint

shops = html.select('ul.quickSearchResultBoxSidoGugun > li')

time.sleep(3)

shop = shops[0]


# 위도 : data-lat(딕셔너리 형태로 추출. 즉, 키값 : 밸류값)
lat = shop['data-lat']
# 경도 : data-long(딕셔너리 형태로 추출. 즉, 키값 : 밸류값)
long = shop['data-long']
# 지점명 : data-name(딕셔너리 형태로 추출. 즉, 키값 : 밸류값)
name = shop['data-name']
# 주소 : 딕셔너리 형태가 아니기 때문에 select으로 추출.
address = str(shop.select('p.result_details')[0]).split('<br/>')[0].split('>')[1]
# 전화번호 : 주소와 전화번호는 묶여 있으므로 split으로 추출.
tel = str(shop.select('p.result_details')[0]).split('<br/>')[1].split('<')[0]
# 매장종류 : 딕셔너리 형태가 아니기 때문에 select으로 추출.
stype = shop.select('i.pin_general')[0].text

title_data = []

for shop in shops:
    lat = shop['data-lat']
    long = shop['data-long']
    name = shop['data-name']
    address = str(shop.select('p.result_details')[0]).split('<br/>')[0].split('>')[1]
    tel = str(shop.select('p.result_details')[0]).split('<br/>')[1].split('<')[0]
    stype = shop.select('i')[0].text
    title_data.append([name, lat, long, address, tel, stype])

df = pd.DataFrame(title_data, columns= ['매장명', '위도', '경도', '주소', '전화번호', '매장종류'])
df

df.to_excel('starbucks_seoul.xlsx', index=False)

# df = pd.read_excel('starbucks_seoul.xlsx', header=0)
```

```
##########################################
#               데이터 분석               #
##########################################

!pip install folium

import folium

starmap = folium.Map(
    location=[37.5666805, 126.9784147],     # 구글에서 검색한 서울 시청 좌표
    zoom_start=11,
    tiles = "Stamen Terrain"
)

folium.CircleMarker(
    location=[37.5666805, 126.9784147],
    fill= True
).add_to(starmap)


# 타일 종류
# OpenStreetMap (기본값)
# Stamen Terrain, Stamen Toner, Stamen Watercolor
# CartoDB positron, CartoDB dark_matter
```

```
starmap
```

```
starmap = folium.Map(
location=[37.5666805, 126.9784147],     # 구글에서 검색한 서울 시청 좌표
zoom_start=11,
tiles = "Stamen Terrain"
)

for idx in df.index:
    lat = df.loc[idx,'위도']
    lng = df.loc[idx,'경도']

    folium.Marker(
    location=[lat, lng],
    fill= True
    ).add_to(starmap)



starmap
```

```
# 각 지점에서 '구' 이름 추출

gu_names = []

for addr in df['주소']:
    gu = addr.split()[1]
    gu_names.append(gu)

print(gu_names)
print(len(gu_names))

# 기존의 데이터프레임에 시군구명 추가

df['시군구명'] = gu_names
df

# 구별 스타벅스 매장 수 계산

starbucks_count = df.pivot_table(index='시군구명',
                                 values="매장명",
                                 aggfunc='count').rename(columns={'매장명':'매장수'})


starbucks_count

# 거주자 수, 업체 수 등의 엑셀 파일과 결합하기

seoul_sgg = pd.read_excel('seoul_sgg_stat.xlsx', thousands=',')
seoul_sgg.head()

seoul_sgg = pd.merge(seoul_sgg, starbucks_count, how='left', on='시군구명')
seoul_sgg

# 주민등록인구 십만명당 매장수

seoul_sgg['거주자_십만명당_매장수'] = seoul_sgg['매장수'] / (seoul_sgg['주민등록인구'] / 100000)

seoul_sgg

# 종사자수 십만명당 매장수

seoul_sgg['종사자수_십만명당_매장수'] = seoul_sgg['매장수'] / (seoul_sgg['종사자수_y'] / 100000)

seoul_sgg

# 사업체수 천개당 매장수

seoul_sgg['사업체수_천개당_매장수'] = seoul_sgg['매장수'] / (seoul_sgg['사업체수_y'] / 10000)
# seoul_sgg.drop(['사업체수_천명당_매장수'], axis=1, inplace=True)
seoul_sgg
```