

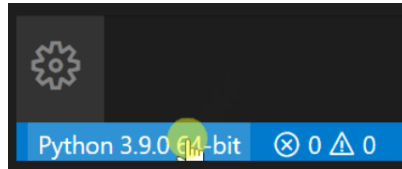
# Python, Selenium을 이용한 웹 크롤링

# 개발환경 구축 1. Python 설치

- Python 설치(!설치경로를 반드시 기억해 둘 것)
  - <https://www.python.org/> > Download > Download for Windows
  - 반드시 Add python 3.x to PATH 체크
- Pip(패키지 관리자) 업그레이드
  - 검색 -> cmd -> 명령프롬프트 열기
  - `python -m pip install --upgrade pip`
- 각종 패키지 설치
  - `pip install selenium`
  - `pip install numpy`
  - `pip install matplotlib`
  - `pip install scipy`

# 개발환경 구축 2. VSCode 설치

- VSCode 설치
  - [code.visualstudio.com](https://code.visualstudio.com) >Download for Windows
  - 왼쪽의  (확장) 클릭 후 확장 설치
    - Korean Language Pack
    - Python
    - Bracket Pair Colorizer
  - 파일 > 자동저장 클릭
  - 파일 > 새로 만들기 > 다른 이름으로 저장 > (파일이름).py 로 저장
  - 왼쪽 하단에 파이썬이 정상적으로 뜨는지 확인



# 개발환경 구축 2. VSCode 설치

- 파이썬 동작 여부 확인해보기
- 아래와 같이 작성 후 control+F5 혹은 실행>디버깅 없이 실행

```
print('Hello World')
```

- 하단의 터미널 창에 Hello World 출력 확인

# 개발환경 구축 3. WebDriver 설치

- Chrome 설치
- 구글에서 ChromeDriver 검색 후 win32 파일로 다운
  - 현재 크롬 버전과 같은 것으로 받을 것( ... → 도움말 → chrome정보 확인)
  - <https://chromedriver.chromium.org/downloads>
- 파이썬 경로로 들어가서 압축 해제한 파일을 붙여넣기
- Crawling.py 를 VSCode에서 열기
- 파일 우클릭 Run python file in Terminal 후 브라우저가 뜨는지 확인

#selenium의 web driver 라이브러리를 불러온다.

```
from selenium import webdriver
```

#driver 변수에 웹드라이버를 불러오기(크롬)

```
driver = webdriver.Chrome()
```

#가져올 URL

```
URL='https://www.google.com'
```

#url= 해당 경로의 웹페이지를 web driver를 이용해 브라우저에 띄운다.

```
driver.get(url=URL)
```

# 웹 크롤링이란

- 웹 페이지의 필요한 데이터만을 자동으로 가져오는 것.
- 검색엔진, 숙박업소 예약 앱, 날씨 앱 등에 활용
- 데이터 제공자의 동의 없이 이용할 경우 불법의 소지가 있어 주의.  
(특히 상업용)

# 웹 크롤링 방법

- Html과 CSS의 class\_name을 활용한 방법
  - 실제 웹에 표시되는 데이터는 데이터베이스에서 스크립트를 통해 전달되는 경우가 많음.
  - Python+Selenium을 많이 활용(일종의 웹 매크로로 기능함)
- API 활용하기: 프로그래밍 인터페이스
  - 데이터를 제공한 곳에서 데이터를 활용할 수 있도록 프로그래밍 해놓은 것,
  - 유/무료로 공개키나 토큰을 받아 씀.
  - 예: 오픈스트리트맵, 날씨API, 공공데이터포털 등



# 웹 크롤링을 위한 사전지식

- 웹페이지의 구성요소: Html, CSS, Javascript
- Html(HyperText Markup Language): 웹 페이지 표준 형식 문서
  - Markup: 태그 등을 활용해 문서와 데이터 구조를 체계화한 언어
- CSS(Cascading Style Sheets): 레이아웃, 디자인, 애니메이션 구현
- Javascript: 객체 지향 프로그래밍 언어,
- Html(문서의 내용)-CSS(문서의 디자인)-Javascript(기능 구현)

# Selenium 실습 1. 미세먼지 값 가져오기

- 개발자 도구 활용 필수
- 크롬: 브라우저 > 메뉴 > 도구 더보기 > 개발자 도구 혹은 shift+ctrl+c
- 가져오고 싶은 값을 클릭하면 html상의 위치를 보여줌.
- [https://github.com/cherub8128/Lecture/crawling\\_PM2.5.py](https://github.com/cherub8128/Lecture/crawling_PM2.5.py) 참고

# Selenium 실습 1. 미세먼지 값 가져오기

**div.pollutant-concentration** 294 × 18

Color ■ #1F1F1F

Font 14px Solis, Arial, Helvetica, sans-serif

ACCESSIBILITY

Contrast Aa 16.48 ✓

Name

Role generic

Keyboard-focusable ⓧ

15 µg/m³

초미세먼지는 직경이 2.5마이크로미터 이하인 것이  
마실 수 있는 오염물질 입자로서 폐와 혈류에 들어갈  
수 있으며, 이로 인해 심각한 건강 문제가 초래될 수  
...자세히

```
<div class="glacier-top content-module" data-aq-type="top" data-viewport="table">
  id="top" style="display: none;"></div>
<div class="two-column-page-content">
  <div class="page-column-1">
    <a href="/ko/kr/incheon/224032/weather-warnings/224032" class="severe-alert-ban
nt-module lbar-banner">...</a>
    <div class="content-module">
      <div id="current">...</div>
      <div class="air-quality-current-pollutants pollutants">
        <div class="tabs-nav">...</div>
        <div class="tabs-content">
          <div id="pollutants" class="tab-content pollutants">
            <div class="air-quality-pollutant" data-qa="airQualityPollutantO3">...</c
...
            <div class="air-quality-pollutant" data-qa="airQualityPollutantPM2_5">
              <h3 class="column">...</h3>
              <div class="column mobile-middle">
                <div class="pollutant-index">31</div>
                <div class="pollutant-concentration">15 µg/m³</div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

#selenium의 webdriver를 불러온다.

```
from selenium import webdriver
```

#가져올 URL을 쓴다.

```
URL='https://www.accuweather.com/ko/kr/incheon/224032/air-quality-index/224032'
```

#driver 변수에 크롬 web driver 불러오기

```
driver = webdriver.Chrome()
```

#해당 URL로 접근

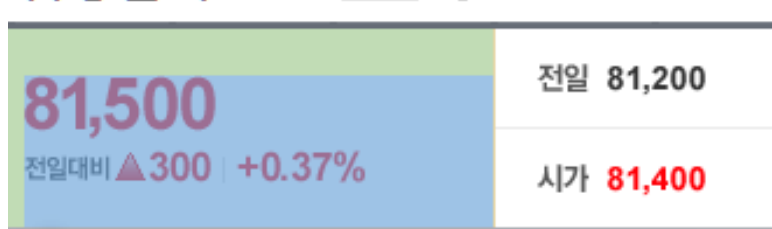
```
driver.get(URL)
```

```
#div태그로 시작하는 부분에서 data-qa
가 airQualityPollutantPM2_5인 곳을 찾아 통째로 저장한다.
partOfPM2_5 = driver.find_element_by_xpath("//div[@data-
qa='airQualityPollutantPM2_5']")
#다시 그 부분에서 class 이름이 pollutant-concentration라
는 것을 찾아 저장한다.
pollutant_conc = partOfPM2_5.find_element_by_class_name("p
ollutant-concentration")
#그것의 텍스트를 출력한다.
print("현재시각 미세먼지 pm2.5의 농
도: "+pollutant_conc.text)
#출력 결과물
현재시각 미세먼지 pm2.5의 농도: 15 µg/m³
```

# Selenium 실습 2.삼성 주가 가져오기

- 직접해 본 뒤 crawling\_stock.py 파일 참고하기

삼성전자 005930 코스피 2021.03.26 기준(장마감)



div.today

209 × 85

Color #5F5F5F  
Font 12px Dotum, 돋움, Gulim, 굴림, AppleGot...  
Background #FAFAFA  
Padding 20px 0px 0px 7px

ACCESSIBILITY

Name  
Role generic  
Keyboard-focusable

삼성전자 005930 코스피 2021.03.26 기준(장마감)



```
<div id="content">  
  <div id="chart_area" class="spot">  
    <!-- chart가 사라질때 <div class="spot spot_short">-->  
    <div class="rate_info"> == $0  
      <div class="today">  
        <p class="no_today">...</p>  
        <p class="no_exday">...</p>  
      </div>  
    </div>  
  </div>
```

```
from selenium import webdriver
URL='https://finance.naver.com/item/main.nhn?code=005930' #가
저올 URL
#driver 변수에 크롬 web driver 불러오기
driver = webdriver.Chrome()
#해당 URL로 접근
driver.get(URL)
#삼성 현재주가를 받아오기 위해 class 이름이 today인 값 찾아 저
장하기
today = driver.find_element_by_xpath("//div[@class='today']")
#p 태그를 갖고 있는 모든 값을 찾아 저장하기
p = today.find_element_by_tag_name("p")
#p의 텍스트만 받아 줄바꿈을 없애고 출력하기
samsung_stock = p.text.replace('\n', '')
print(samsung_stock)
```

# 크롤링에 사용한 함수 설명

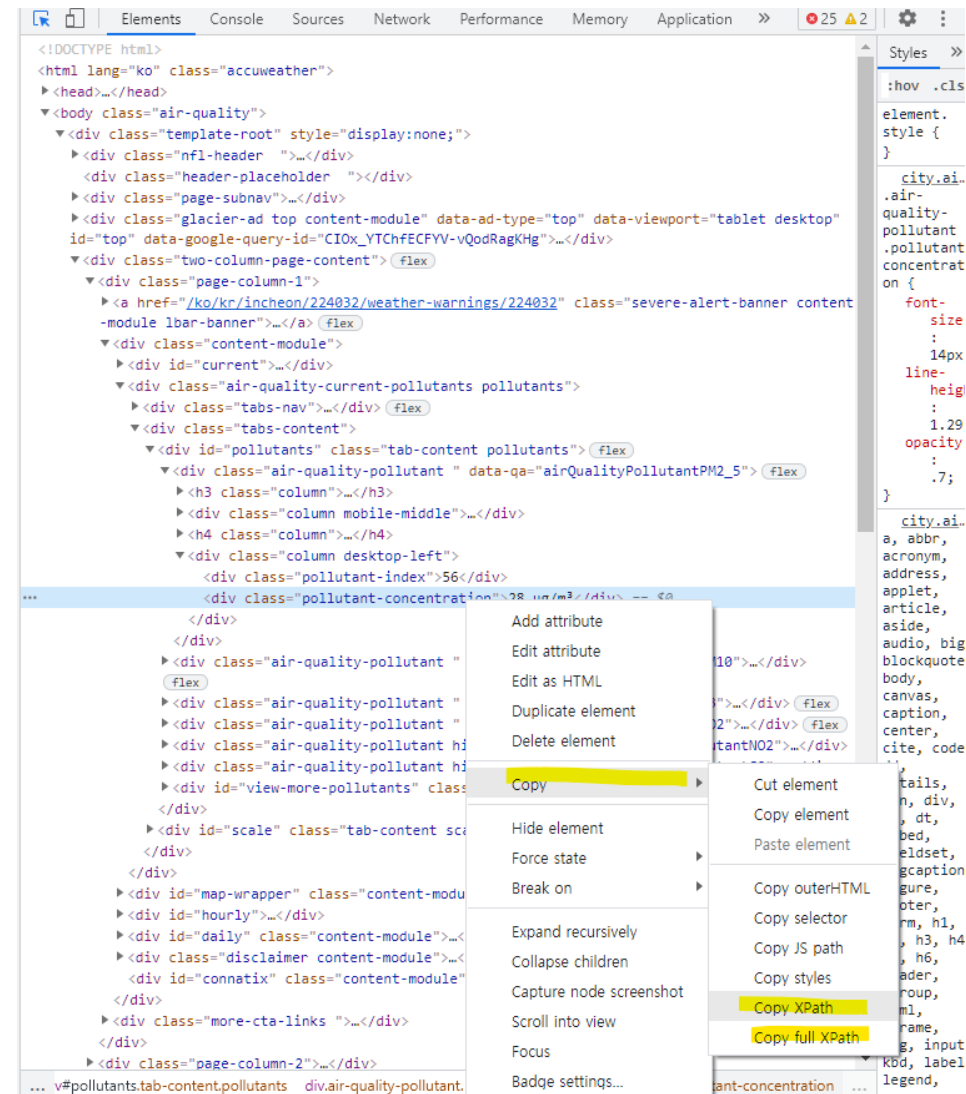
- `find_element_by_xpath()` 괄호 안에 들어간 값에 따라
  - `(//)` 문서 내에서 검색
  - `(//div[@class="name"])` div태그 안 class라는 값이 name인 것 모두 선택
  - `(//@href)` href 속성이 있는 모든 태그 선택
  - `(//a)[n]` n번째 링크 선택
  - `(//a)[position()>2]` 2번째 이후 링크 모두 선택
  - `(//table)[last()]` 문서 내 마지막 표 선택
  - `(//table/tr/*)` 문서 내 모든 표의 tr태그 모두 선택
- `find_element_by_class_name("find")` 클래스 이름이 find인 것 찾기
- `str.replace('a', 'b', n)` 텍스트 str의 a를 b로 n회 바꾼 값 n을 생략하면 모두 바뀜



# 꿀팁. 위치 가져오기

- Xpath의 경우 크롬에서 쉽게 가져올 수 있다.

1. 해당 위치에서 우클릭
2. Copy>Copy Xpath 혹은 Copy full Xpath를 클릭
3. find\_element\_by\_xpath()의 괄호 안에 그대로 붙여넣기



# Selenium 실습 3. 뽑아온 값 저장하기

- txt 파일에 저장해보기
- 콤마로 구분하여 데이터 저장시 엑셀파일로 변환 가능
- 파이썬 언어에 대한 사전 지식이 필요함
- [https://github.com/cherub8128/Lecture/crawling\\_stock\\_save.py](https://github.com/cherub8128/Lecture/crawling_stock_save.py) 참고

```
from selenium import webdriver
```

```
#현재시간을 받아올 수 있는 라이브러리를 불러온다.
```

```
from datetime import datetime
```

```
#시간 지연을 위한 라이브러리
```

```
import time
```

```
#가져올 URL
```

```
URL='https://finance.naver.com/item/main.nhn?code=005930'
```

```
#driver 변수에 크롬 web driver 불러오기
```

```
driver = webdriver.Chrome()
```

```
#해당 URL로 접근 (새로고침이 필요한 경우 while문 안으로)
```

```
driver.get(URL)
```

# 무한 반복하기

while True:

# 삼성 현재주가를 받아오기 위해 class 이름이 today인 값 찾아 저장하기

today = driver.find\_element\_by\_xpath("//div[@class='today']")

# p 태그를 갖고 있는 모든 값을 찾아 저장하기

p = today.find\_element\_by\_tag\_name("p")

# p의 텍스트만 받아 줄바꿈을 없애고 저장하기

samsung\_stock = p.text.replace('\n', '')

# 현재시간을 저장하기

now = datetime.now()

text = f"{now.hour}:{now.minute}:{now.second},삼성전자,\"{samsung\_stock}\"\\n"

# 파일을 추가 쓰기모드로 열어 text를 저장하고 닫는다.(없으면 만든다)

with open(f"삼성주가 {now.year}년 {now.month}월 {now.day}일.txt", 'a') as f:

    f.write(text)

# n초 기다리기

time.sleep(3)

# 중급 및 고급 기술

- 얻어낸 데이터를 시각화: Matplotlib
- 데이터의 수학적 및 과학적 분석: Numpy, Scipy
  - 베이지안, 빈도론, 가설검정, t검정, 추정 등
- 영상 등 비가공 데이터 추출: 영상 처리, 자연어 처리
- 패턴 분석 및 미래값 예측: 머신 러닝
- Selenium 기타 활용: 예약 구매 매크로, 인스타 좋아요 매크로 등

**감사합니다.**  
**〈질문 시간〉**