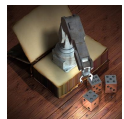
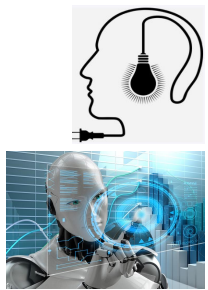


In the future, an AI agent will know that you are at work and have ten minutes free, and then help you accomplish something that is high on your to-do list.

# 데이터수집하기

강희숙



# PART1. 데이터 수집하기

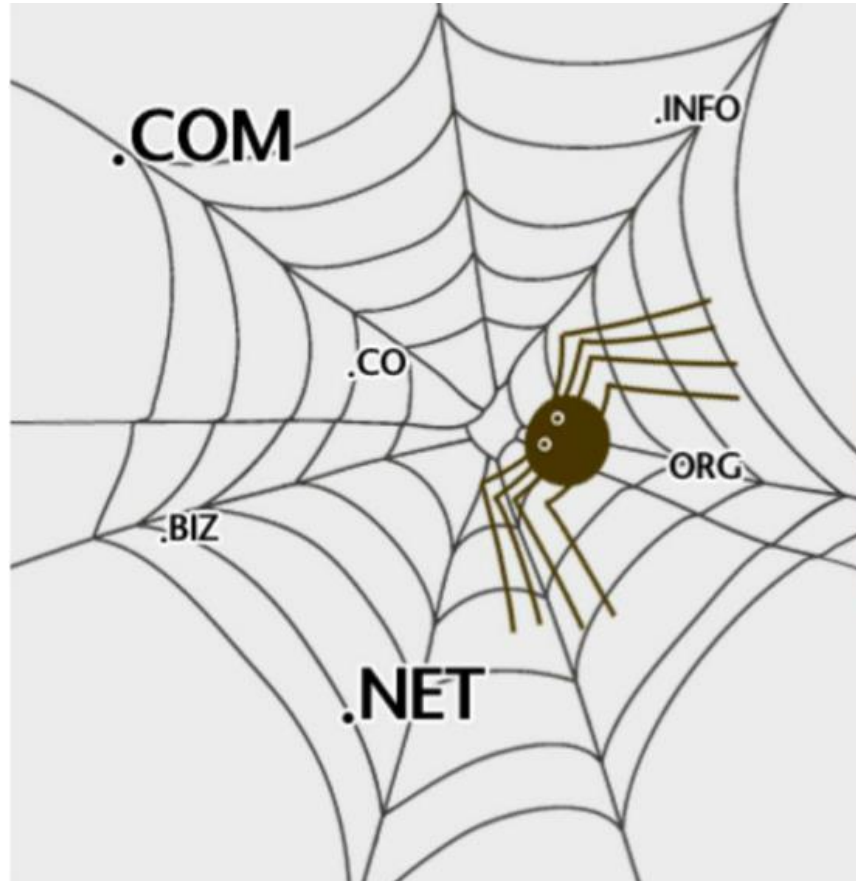
크롤링 이해 및 사용

Enjoy your possibility



# 웹 크롤러(Web Crawler)란

- Web 페이지를 방문하여 자동적으로 수집하는 프로그램



# 웹 크롤링(Crawling)이란

- Web상에 존재하는 Contents를 수집하는 작업 ( 프로그래밍으로 자동화 기능)으로 **웹 스크래핑(Web Scraping)** 이라고도 함

## 크롤링 기법

브라우저를 프로그래밍으로 조작해서 필요한 데이터만 추출하는 기법  
**Selenium 라이브러리**

사용HTML 페이지를 가져와 HTML/CSS 등을 파싱하고 필요한  
데이터만 추출하는 기법  
**BeautifulSoup 라이브러리 사용**

Open API를 제공하는 서비스에 Open API를 호출해서 받은 데이터 중  
필요한 데이터만 추출하는 기법

# 크롤링을 위한 선행 학습

---

- 다음 분야에 대해 기본 지식이 선행되어야 함

- 웹(Web)의 개념

- HTML, CSS, JavaScript 구조 및 태그

- 파이썬 기초

# 자동화 데이터 수집 절차



## 최저가 휴대폰 구매 하는 법

1. 무엇을 살지 정한다(기종/사양 등)
2. 최저가 판매 하는 곳을 찾는다
3. 찾아간다
4. 제품이 제대로 있는지,어떠한 조건인지? 확인한다(약정기간/부가서비스/지원금 등)
5. 내가 원하는 조건에 맞는지 살펴본다  
➔ 불가능하면 다른 곳을 찾는다
6. 괜찮으면 산다
7. 사용한다

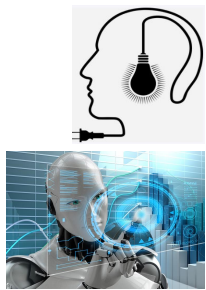
## 크롤링 하는 방법

1. 원하는 정보를 정한다
2. 정보가 있는 url을 찾는다
3. 해당 페이지에 접근한다
4. 페이지내 원하는 정보가 있는지? 어떠한 구조로 되어있는지? 살펴본다
5. 원하는 정보를 찾는다  
➔ 불가능하면 다른 방법을 찾는다
6. 맞으면 가져온다
7. 데이터 활용한다

# 크롤링(Crawling) 방식

|         | 정적 수집  | 동적 수집  |
|---------|--|--|
| 사용 패키지  | requests / urllib  | selenium   |
| 수집 커버리지 | 정적 웹 페이지   | 정적/동적 웹 페이지  |
| 수집 속도   | 빠름 (별도 페이지 조작 필요 X)  | 상대적으로 느림   |
| 파싱 패키지  | beautifulsoup  | beautifulsoup / selenium   |
| 수집 순서   | <p>1단계: 목표로 하는 웹 페이지의 html을 requests 패키지를 이용하며 받아 옴</p> <p>2단계: 가져온 html 문서 전체를 beautifulsoup4 패키지를 이용하여 파싱(parsing)함</p> <p>3단계: 필요한 정보만 골라서 리스트에 담음.</p> <p>4단계: 리스트를 print() 함수로 출력하던가, excel이나 csv 파일에 저장.</p>  | <p>1단계 : 조작을 원하는 버튼이나 입력창의 html을 파악</p> <p>2단계 : 아래의 두 함수에 html 정보를 입력해서 객체(버튼/입력창 등) 선택</p> <ul style="list-style-type: none"> <li>- find_element ( )</li> <li>- find_elements( )</li> </ul> <p>3단계 : 기능 동작 관련 함수로 원하는 기능 조</p> <ul style="list-style-type: none"> <li>- 클릭 : .click( )</li> <li>- 키 입력: .send_keys( )</li> </ul>  |





# PART1. 데이터 수집하기



Enjoy your possibility

Requests 모듈



# Requests 모듈 사용

- Url로 웹페이지 정보 요청하는 라이브러리

- 데이터와 함께 POST 요청 – 서버에 보냄

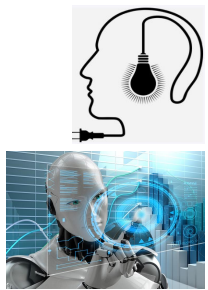
```
post(url, data=XX), post(url, files=XXX)
```

- 헤더, 쿠키와 함께 GET 요청

```
get(url, headers=XXX, cookies=XXX)
```

- 인코딩 설정

```
Encoding='UTF-8'
```



# PART1. 데이터 수집하기



Enjoy your possibility

## BeautifulSoup 모듈

# BeautifulSoup 모듈 사용

- BeautifulSoup 은 HTML 및 XML 파일에서 원하는 데이터를 손쉽게 Parsing 할 수 있는 Python 라이브러리

```
1 !pip install bs4
```

```
1 import requests
2 from bs4 import BeautifulSoup
3
4 # requests 라이브러리 활용하여 html 페이지 요청하고
5 # res 객체에 html 데이터 저장
6 res = requests.get("https://tv.naver.com/r")
```

```
1 # BeautifulSoup 라이브러리로 html 파싱
2 html = BeautifulSoup(res.text, 'html.parser')
3
4 # 필요한 데이터 검색
5 title = html.find('title')
6
7 # 필요한 데이터 추출
8 print(title.get_text())
```

# Html(Hyper Text Markup Language)

---

- 웹 문서의 구조를 정의하고 콘텐츠를 표현하는 기본 마크업 언어
- HTML 태그(tag)들로 구성되며, 각각의 태그들은 디자인이나 기능을 결정하는데 사용
- Hyper Text: 링크가 포함된 텍스트
- Markup Language : 텍스트에 의미를 부여하기 위해 주석을 다는 시스템

# Html 구조

---

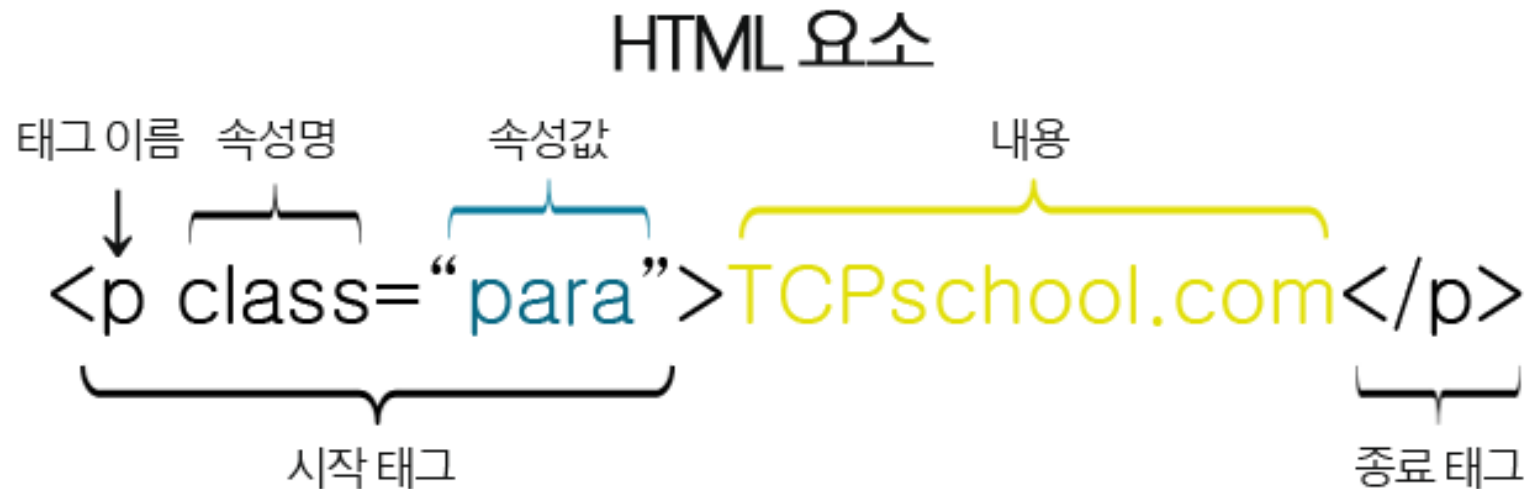
- `<!Document html>` : Html5 문서를 선언하는 구문
- `<html></html>` : HTML 문서의 시작과 끝
- `<head></head>` : CSS, JavaScript, meta, title 정보들을 설정
- `<body></body>` : 실제 홈페이지 화면에 나타나는 부분

```
<!doctype html>  <!-- HTML5 선언 -->
<html>            <!-- HTML의 시작 -->
  <head>
    <title></title>
    <!-- CSS, JAVASCRIPT, META -->
  </head>
  <body>
    <!-- 본문 -->
  </body>
</html>
```

# Html 요소 구조

- HTML 요소(element)는 여러 속성을 가질 수 있으며, 이러한 속성(attribute)은 해당 요소에 대한 추가적인 정보를 제공합니다.

참조 : [https://tcpschool.com/html/html\\_intro\\_elementStructure](https://tcpschool.com/html/html_intro_elementStructure)



- HTML 요소는 시작 태그로 시작해서 종료 태그로 끝남
- 속성은 HTML 요소 중에서도 언제나 시작 태그 내에서만 정의되며, 속성 이름과 속성값(value)으로 표현됨

# HTML의 태그(TAG)

---

## 이미지, 리스트, 표, 스타일 등 기본요소

- `<img>`
- `<ul>` `<ol>` `<li>`
- `<table>` `<tr>` `<td>`
- `<style>`

## 제목, 주석, 문단 등의 텍스트 표현 요소

- `<title>`
- `<!-- -->`
- `<p>` `<span>`

## 레이아웃을 표현하기 위한 공간 분할 요소

- `<div>`
- `<section>` `<article>`  
`<nav>`



# HTML의 태그(TAG)

| 태그           | 설명   | 사용 예   |
|--------------|--|--|
| <b>h1~h6</b> | 제목을 입력할 때 사용<br>(h1이 가장 큰 제목, h6이 가장 작은 제목)                                      | <code>&lt;h1&gt; 제일 큰 제목 &lt;/h1&gt;</code>  |
| <b>p</b>     | 하나의 문장을 입력할 때 사용   | <code>&lt;p&gt; 문장 &lt;/p&gt;</code>   |
| <b>div</b>   | 박스형태의 구역 설정 (block 요소)<br>(다른 태그들이 div 안에 모여있게 됨)                                | <code>&lt;div&gt;&lt;h3&gt; 제목 &lt;/h3&gt;&lt;p&gt; 문장 &lt;/p&gt;&lt;/div&gt;</code>   |
| <b>span</b>  | 줄 형태의 구역 설정 (inline 요소)<br>(독립적으로 사용하지 않고 p태그 안에 span 태그가 들어감)                   | <code>&lt;p&gt; 이렇게 &lt;span style="border: 3px solid red"&gt; span요소로 텍스트의 일부분 &lt;/span&gt; 만 스타일을 적용할 수 있음&lt;/p&gt;</code> |
| <b>img</b>   | 이미지와 관련된 태그<br>(속성명은 src, 속성값은 "이미지의 url 주소")<br>종료 태그(/img)가 없는 빈 태그(empty tag) | <code>&lt;img src="/img 주소.png" alt="이미지가 없을 때 이미지 대신 출력할 문장 입력"&gt;</code>  |
| <b>a</b>     | 하이퍼링크를 추가할 때 사용  | <code>&lt;a href="링크 주소"&gt;HTML 링크&lt;/a&gt;</code>   |
| <b>ul</b>    | unordered list : 기호로 된 리스트<br>li 태그가 하위 태그로 사용되어 내용을 채움                          | <code>&lt;ul&gt;&lt;li&gt;기호&lt;/li&gt;&lt;li&gt;기호&lt;/li&gt;&lt;/ul&gt;</code>   |
| <b>ol</b>    | ordered list : 순서가 있는 리스트<br>li 태그가 하위 태그로 사용되어 내용을 채움                           | <code>&lt;ol&gt;&lt;li&gt;1번&lt;/li&gt;&lt;li&gt;2번&lt;/li&gt;&lt;/ol&gt;</code>   |

# HTML의 태그(TAG)

[태그 예시1] 테이블 <table>

```

<!doctype html>
<html>
  <head>
    <title></title>
  </head>
  <body>
    <table border="1">
      <thead>
        <tr>
          <td>1</td>
          <td>2</td>
          <td>3</td>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>a</td>
          <td>b</td>
          <td>c</td>
        </tr>
      </tbody>
    </table>
  </body>
</html>

```


<thead> : 표의 제목

<tbody> : 표의 본문

<tr> : 하나의 행

<td> : 하나의 열

※ <tfoot> 표의 하



[태그 예시2] 단락 및 줄

```

<!doctype html>
<html>
  <head>
    <title></title>
  </head>
  <body>
    <p>aaa</p>
    <p>bbb</p>
    aaa<br>bbb
    <div>aaa</div>
    <div>bbb</div>
    aaa
    bbb
  </body>
</html>

```

<p> : 문단(한 줄 모두 차지)

<br> : 줄바꿈

<div> : 하나의 영역(한줄 모두 차지)



# CSS(Cascading Style Sheets)

---

- HTML 문서에 각종 시각적 요소를 정의하기 위한 스타일 시트 언어
- CSS를 사용하는 이유

웹 문서의 내용과 상관없이 디자인만 변경하거나 디자인 변경없이 웹의 내용만 변경하고 싶을 때 사용

디자인과 콘텐츠 분리를 통한 유지보수 및 재사용성 증가

PC, 태블릿, 스마트폰 등 다양한 환경에서 사용하고 싶을 때 반응형 디자인으로 설계 및 사용

동일한 콘텐츠 및 구조를 가지고 서로 다른 CSS 테마 적용 가능

# CSS 구조

- CSS란 Cascading Style Sheets의 약자입니다.  
CSS는 HTML 요소들이 각종 미디어에서 어떻게 보이는가를 정의하는 데 사용되는 스타일 시트 언어입니다.

참조 : [https://tcpschool.com/css/css\\_intro\\_syntax](https://tcpschool.com/css/css_intro_syntax)



- CSS의 문법은 선택자(selector)와 선언부(declaratives)로 구성됨.
- 선택자는 CSS를 적용하고자 하는 HTML 요소(element)를 가리킴.
- 선언부는 하나 이상의 선언들을 세미콜론(;)으로 구분하여 포함할 수 있으며, 중괄호({ })를 사용하여 전체를 둘러쌈.
- 각 선언은 CSS 속성명(property)과 속성값(value)을 가지며, 그 둘은 콜론(:)으로 연결됨.
- 이러한 CSS 선언(declaration)은 언제나 마지막에 세미콜론(;)으로 끝마침.

# 선택자(Selector)

- CSS 및 JavaScript 적용 위해 적용 대상 필요- 선택자 활용
- 태그, 아이디, 클래스를 선택자로 사용

| 분류        | 설명                           | 예   |
|-----------|------------------------------|---|
| 태그 선택     | 특정 태그를 선택                    | div --> <div> 태그를 선택                              |
| 아이디 선택    | id='속성값'인 태그를 선택             | #query --> id의 속성값이 query인 태그 선택                  |
| 클래스 선택    | class='속성값'인 태그를 선택          | .title --> class의 속성값이 title인 태그 선택               |
| 태그+아이디 선택 | 특정 태그 중 id가 '속성값'인 태그를 선택    | input#query --> input 태그 중, id의 속성값이 query인 태그 선택 |
| 태그+클래스 선택 | 특정 태그 중 class가 '속성값'인 태그를 선택 | p.title --> p 태그 중, class의 속성값이 title인 태그 선택      |

## 선택자(Selector) 종류

- 클래스는 '.' 아이디는 '#'을 사용
- 태그는 조합을 통해 하나 이상을 조합해서 사용 가능

| 선택자    | 사용 예   | 설명                             |
|--------|--------|--------------------------------|
| .class | .abc   | html 태그 중 class="abc"로 된 모든 태그 |
| #id    | #aaa   | html 태그 중 id="aaa"로 된 모든 태그    |
| *      | *      | html 내 모든 요소                   |
| 태그     | p      | html 내 모든 p 태그                 |
| 태그, 태그 | ol, ul | ol 태그와 ul 태그                   |
| 태그 태그  | ul li  | ul 태그 내 li 태그                  |

# JavaScript

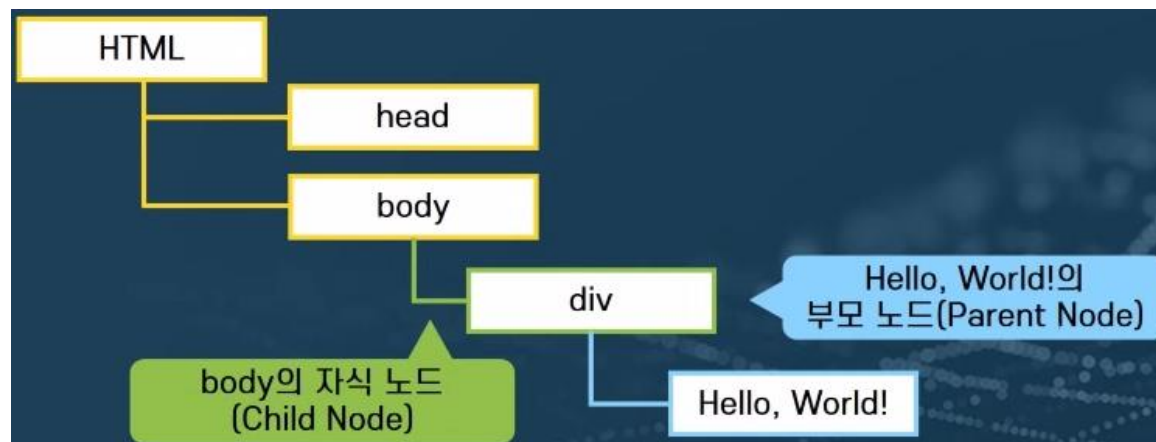
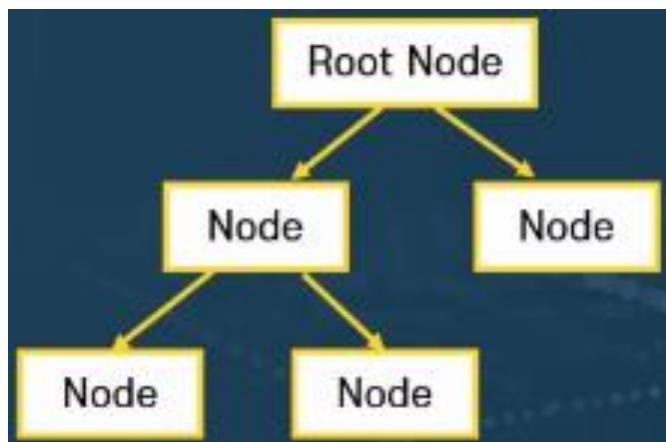
---

- 정적인 HTML 문서를 동적으로 변경하거나 사용자와 상호작용을 통해 콘텐츠를 변경하기 위한 언어
- HTML이나 CSS와는 달리 C언어, JAVA등과 같은 일반적인 프로그래밍 언어와 비슷한 구조
- `<script>` 태그 내부에 작성 - 해당 태그가 없으면 단순한 HTML 콘텐츠로 간주함
- React와 같은 프론트엔트 프레임워크부터 Node.js 같은 서버 프로그래밍까지 전반적으로 JavaScript가 사용



# BeautifulSoup 모듈 응용 - DOM 정의

- DOM(Document Object Model)는 트리 구조로 형성되어 있음
- HTML, XML문서의 프로그래밍 인터페이스
  - 구조화된 표현 및 프로그래밍 언어가 DOM 구조에 접근할 수 있는 방법을 제공함



# JavaScript 특징

---

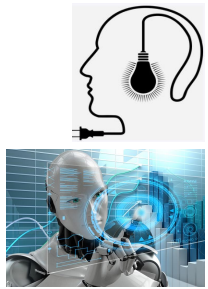
- 인터프리터 언어이며 동적인 자료형을 가지고 있음
- 객체지향 프로그래밍, 함수형 프로그래밍 모두 표현 가능
- HTML의 내용, 속성, 스타일 등을 변경할 수 있음
- 마우스 클릭, 키보드 입력등 이벤트 처리 및 사용자와 상호작용
- AJAX 기술로 서버와 실시간 통신 가능

# HTTP 요청/응답 구조 - 요청

---

- 사용자가 서버에 요청을 할 때 다음 4가지 메서드로 구분하여 URL로 요청

| 메서드    | 설명                |
|--------|-------------------|
| GET    | 정보를 가져오기 위해 요청    |
| POST   | 새로운 정보를 보내기 위해 요청 |
| PUT    | 수정할 정보를 보내기 위해 요청 |
| DELETE | 정보를 삭제하기 위해 요청    |



# PART1. 데이터 수집하기

Selenium 모듈

Enjoy your possibility



# Selenium 모듈 사용

- Selenium은 웹 앱을 테스트할 때 주로 사용하는 프레임워크로 일종의 자동화 프로그램

1

Selenium 설치하기

```
pip install selenium
```

2

브라우저 드라이버 설치하기 (chromedriver download 검색 후 설치 )

```
https://sites.google.com/a/chromium.org/chromedriver/downloads
```

3

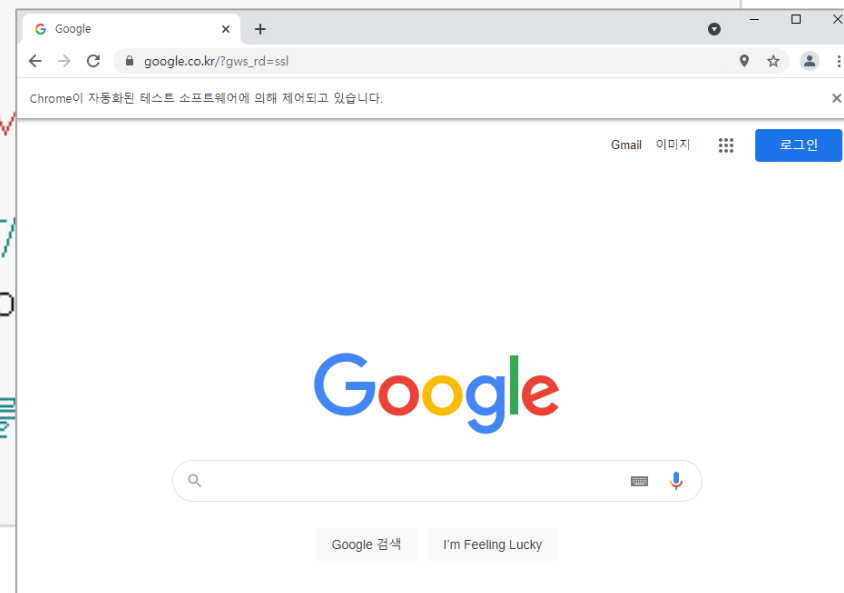
selenium 모듈에서 webdriver를 불러오기

```
from selenium import webdriver
```

# Selenium 사용해보기 (1)

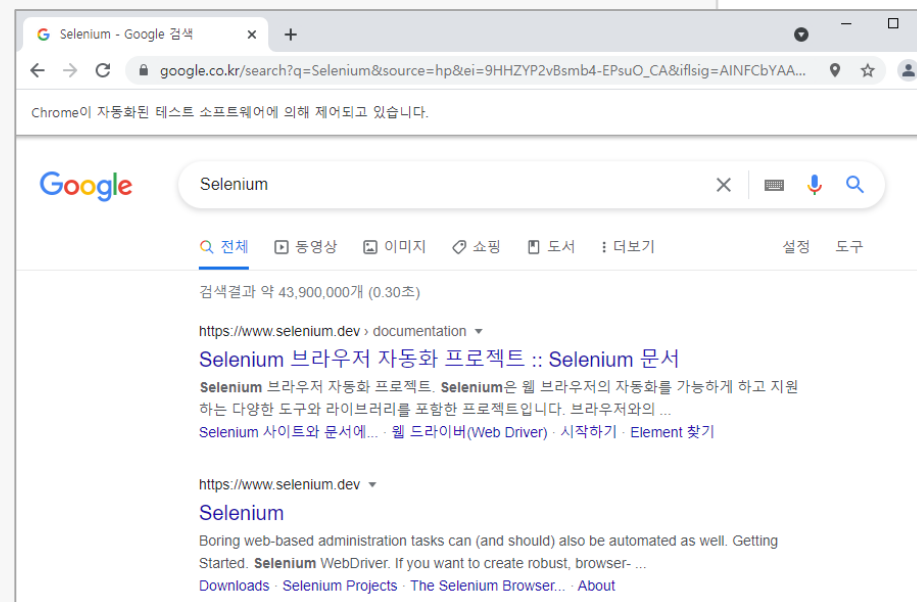
```
1 !pip install selenium
```

```
1 import os
2 from selenium import webdriver
3
4 # chromedriver.exe 경로 지정하기
5 path = os.path.join(os.getcwd(), "chromedrив
6
7 # 서비스를 생성하고 chromedriver의 경로를 지
8 service = webdriver.chrome.service.Service(p
9
10 # 지정된 서비스를 사용하여 Chrome 브라우저를
11 driver = webdriver.Chrome(service=service)
```



# Selenium 사용해보기 (2)

```
1 #검색 입력 부분에 커서를 올리고
2 #검색 입력 부분에 다양한 명령을 내리기 위해 elem 변수에 할당
3 elem = driver.find_element_by_name("q")
4
5 #입력 부분에 지운다-default로 값이 있을 수 있는 경우 고려
6 elem.clear()
7
8 #검색어 입력
9 elem.send_keys("Selenium")
10
11 #검색 실행
12 elem.submit()
13
14 #검색이 제대로 됐는지 확인
15 assert "No results found." not in driver.page_source
16
17 #브라우저 종료
18 #driver.close()
```





# Bing Image Downloader 사용해 동물 이미지 크롤링하기

**Bing Image Downloader** 을  
사용하여 강아지, 고양이, 토끼를  
검색하여 각각 이미지를  
10개씩 저장해 보세요

