

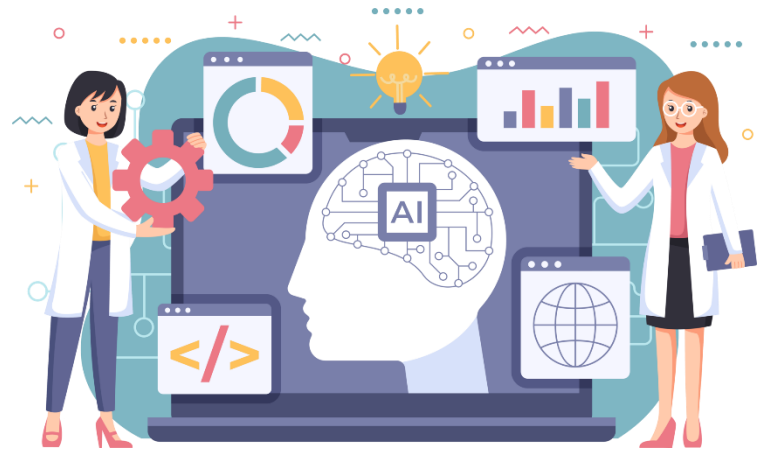


웹 크롤링

인공지능 RPA – 8주차

학습 내용

1. 파이썬 - 웹 크롤링
2. Fast API form 실습



인공지능 실습

_구글 트렌드 검색

<https://trends.google.com/trends/explore?geo=KR&hl=ko>

Google Trends

홈

탐색

실시간 인기

로그인

검색어 추가

대한민국

23. 1. 1.~23. 10. 14.

모든 카테고리

웹 검색

검색 주제 ?

급상승

다운

비교

공유

1 AI Chatbot - 주제	+4,950%	⋮
2 GPT-3 - 주제	+3,900%	⋮
3 2023 - 주제	+350%	⋮
4 채팅 - 주제	+300%	⋮
5 회사 - 조직 유형	+250%	⋮

< 25개 주제 중 1-5번 표시 중 >

검색어 ?

급상승

다운

비교

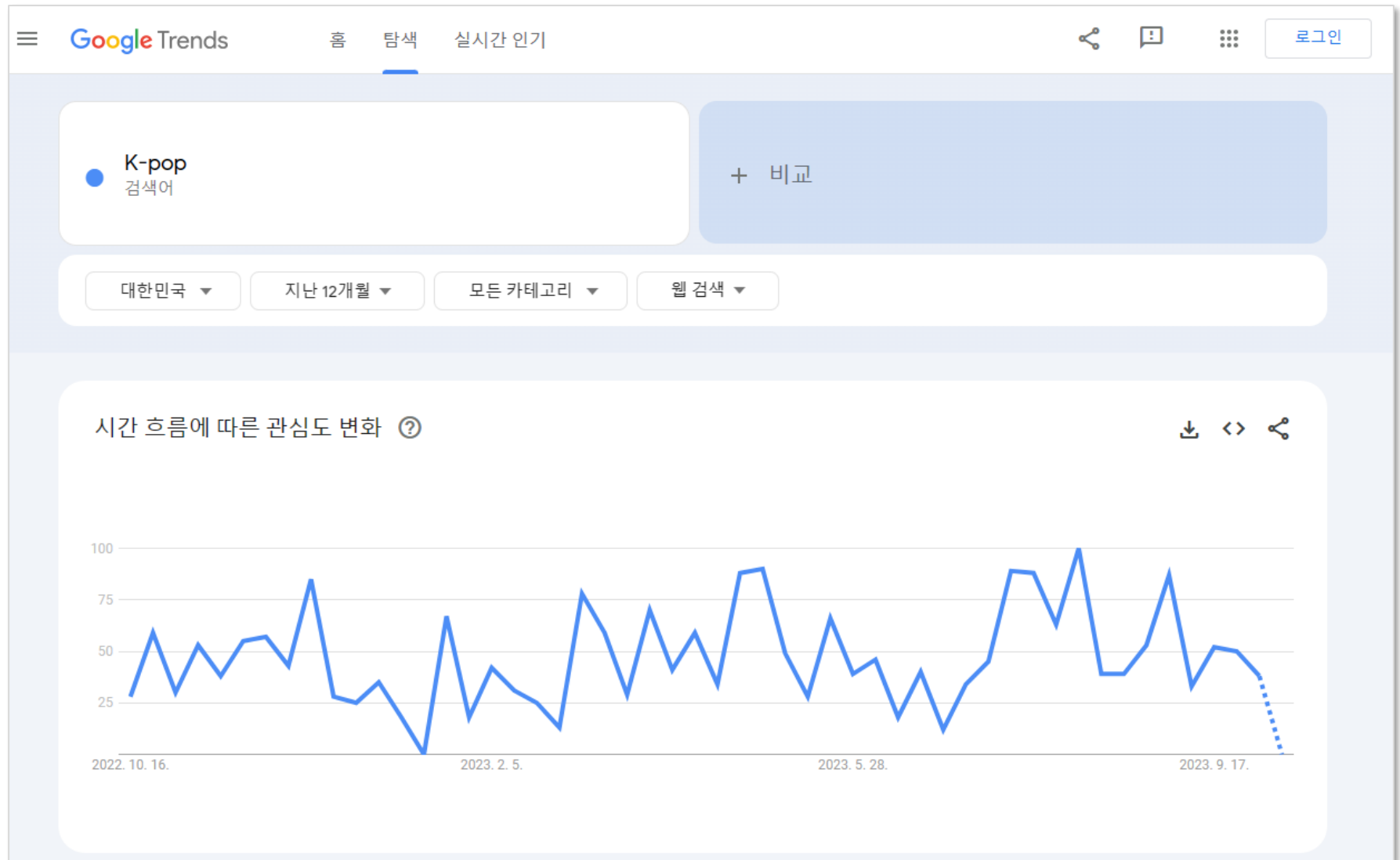
공유

1 챗 gpt	급등	⋮
2 chat gpt	급등	⋮
3 티비 몬	급등	⋮
4 chatgpt	+2,650%	⋮
5 코티 비씨	+1,950%	⋮

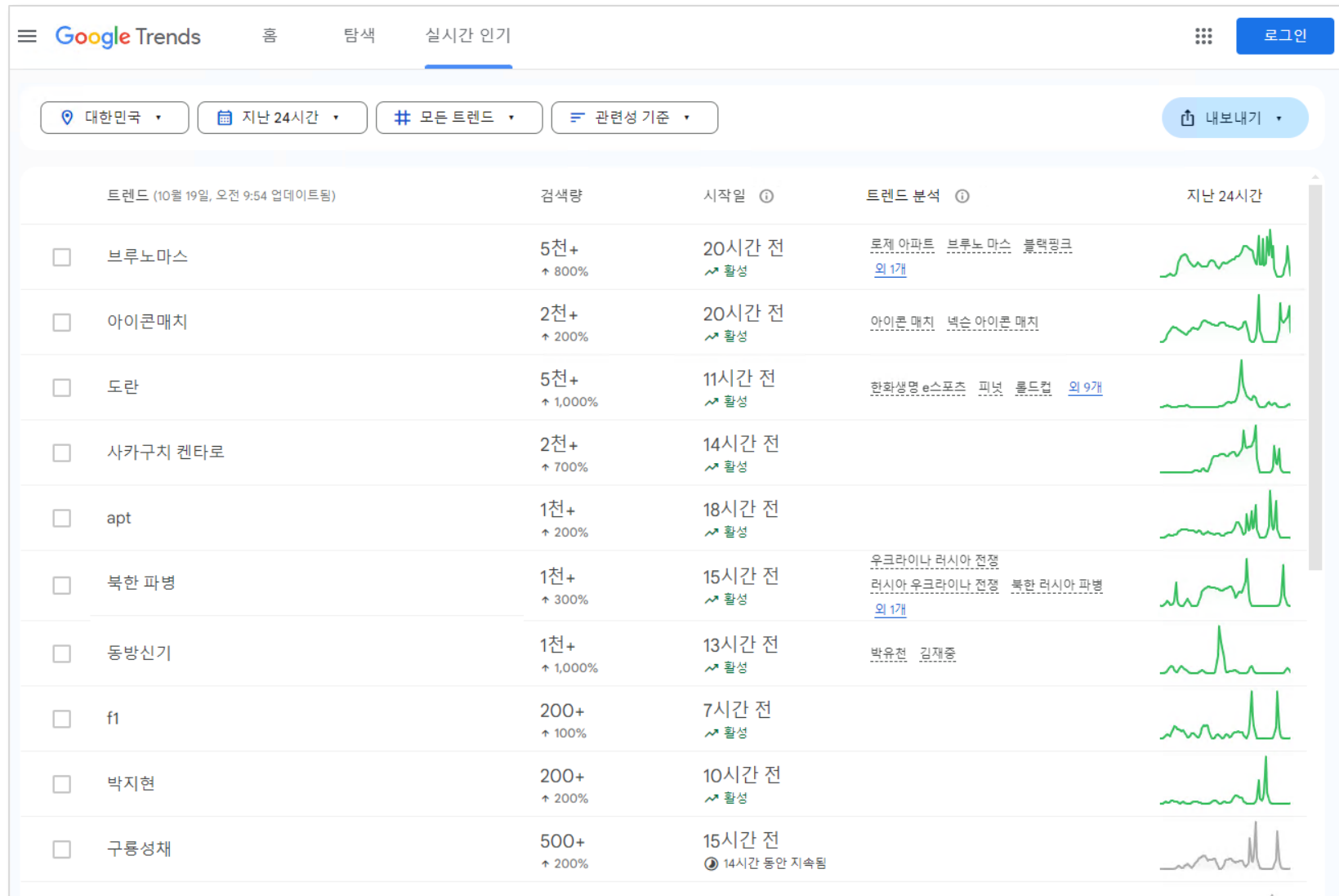
< 23개 검색어 중 1-5번 표시 중 >



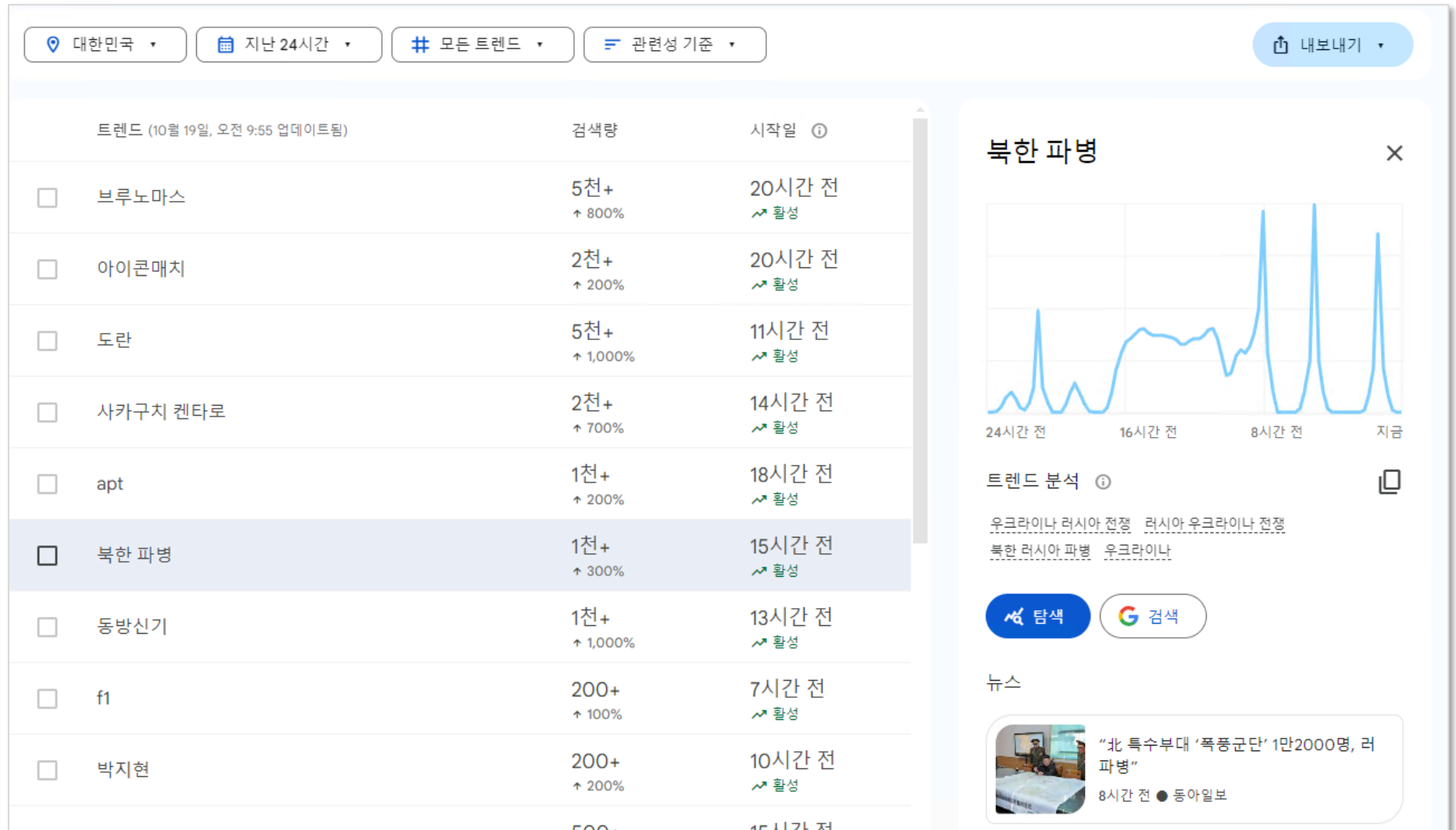
_구글 트렌드 검색



_구글 트렌드 검색



_구글 트렌드 검색



_자동화 프로그래밍 - 웹 크롤링

Web Crawling

- 특정 사이트나 웹 페이지에 접근해서 HTML 문서를 읽어 오는 것
- 웹 브라우징 조작을 자동화 시킨 것 : 조작 순서대로 명령을 배치하는 것

Web Crawling을 위해 필요한 것

- 크롬 브라우저 & selenium

```
pip install --upgrade selenium
```

- 필요한 패키지 import

```
from selenium import webdriver          # selenium의 webdriver를 사용하기 위함
```

```
from selenium.webdriver.common.keys import Keys    # selenium으로 키를 조작
```

```
import time                                     # 페이지 로딩을 기다리는 데에 사용
```



자동화 프로그래밍 - 웹 크롤링 프로그래밍

1. 웹 브라우징 순서 파악하기

2. 자동화 하기 - 1단계 : 페이지 이동

- 1단계 : 크롬 드라이버 로딩 및 사이트 접속

- 2단계 : 페이지에서 타겟 엘리먼트의 CSS Selector 찾기

- . DevTools 이용 : Copy Selector

- . find_element 함수를 이용해서 원하는 엘리먼트와 연결되는 참조변수 만들기



_자동화 프로그래밍 - 웹 크롤링 프로그래밍

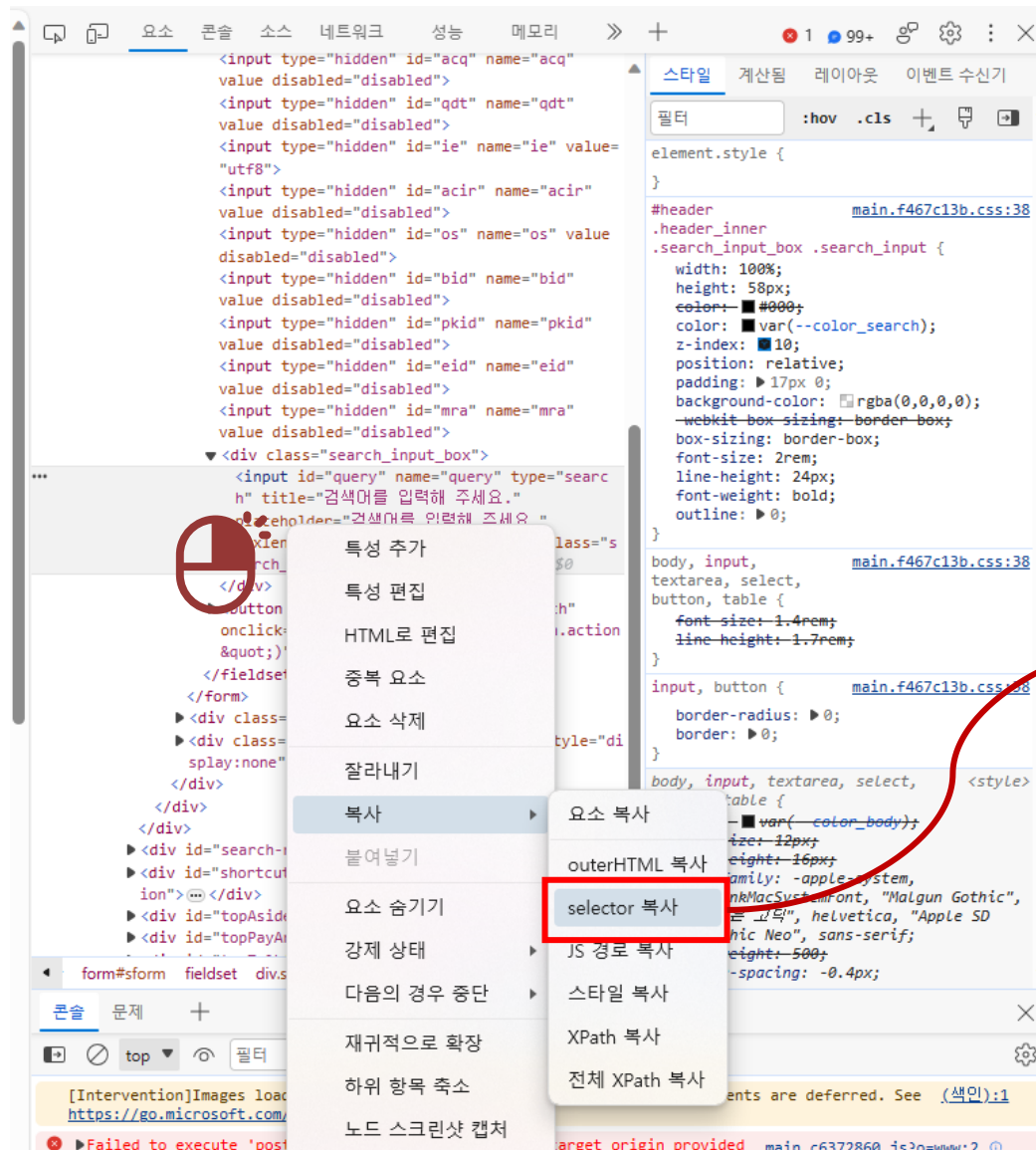
The screenshot shows the Naver homepage with a context menu open over the search bar. The menu includes the following options:

- 이모지 (Win+마침표)
- 음성 입력 (Win+H)
- 실행 취소 (Ctrl+Z)
- 잘라내기 (Ctrl+X)
- 복사 (Ctrl+C)
- 붙여넣기 (Ctrl+V)
- 일반 텍스트로 붙여넣기 (Ctrl+Shift+V)
- 모두 선택 (Ctrl+A)
- 사이드바에 페이지 추가
- 쓰기 방향
- 검사** (highlighted with a red box)

The background shows the Naver homepage with a search bar, navigation links (메일, 카페, 블로그, 쇼핑), a banner for 'GRAND OPEN 2023. 11 대전명', and a grid of news headlines. A red circle with a white 'X' is drawn over the search bar area.



_자동화 프로그래밍 - 웹 크롤링 프로그래밍



요소의 선택자 복사
#query



_자동화 프로그래밍 - 웹 크롤링 프로그래밍

2. 자동화 하기 - 2단계 : 조작 명령 배치

- 3단계 : 원하는 조작 수행 (대상은 참조변수를 통해서)

. 조작 : 마우스 조작, 키보드 조작 수행

클릭 : .click()

키 입력: .send_keys()

. 데이터 입수 : 입력된 데이터 가져오기

명령어	기능
Keys.ENTER Keys.RETURN	엔터
Keys.SPACE	스페이스
Keys.ARROW_UP Keys.ARROW_DOWN Keys.ARROW_LEFT Keys.ARROW_RIGHT	방향키(상하좌우)
Keys.BACK_SPACE Keys.DELETE	지우기 (백스페이스) 지우기 (딜리트)
Keys.CONTROL Keys.ALT Keys.SHIFT Keys.TAB	자주 사용하는 기능키(Ctrl, Alt, Shift, Tab)
Keys.PAGE_UP Keys.PAGE_DOWN	스크롤 업 스크롤 다운
Keys.F1~9	F1 부터 F9 (F+숫자)
Keys.EQUALS, Keys.ESCAPE, Keys.HOME, Keys.INSERT	기타 등등



_자동화 프로그래밍 - 웹크롤링 기본 실습

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
import time
```

```
url = 'https://www.google.co.kr/'
```

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

```
driver.get(url)
```

```
time.sleep(3)
```

```
search_box = driver.find_element(By.CSS_SELECTOR, "#APjFqb")
```

```
search_box.send_keys("KBO 한국시리즈")
```

```
search_box.send_keys(Keys.RETURN)
```

```
time.sleep(20)
```



_자동화 프로그래밍 - 웹 크롤링 실습 2

검색어를 입력 키보드로 입력 받은 후 네이버에 접속해서 입력한 검색어로 검색을 수행

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
import time
```

```
# 검색할 키워드 입력
query = input('검색할 키워드를 입력하세요: ')
```

```
# 크롬드라이버로 원하는 url로 접속
url = 'https://www.naver.com/'
driver = webdriver.Chrome()
driver.get(url)
time.sleep(3)
```

```
# 검색창에 키워드 입력 후 엔터
search_box = driver.find_element(By.CSS_SELECTOR, "#query")
search_box.send_keys(query)
search_box.send_keys(Keys.RETURN)
time.sleep(20)
```



_자동화 프로그래밍 - 웹 크롤링 실습 3

결과 읽어 오기

: HTML 요소의 속성과 CSS의 Selector 개념을 알아야 함

– 텍스트 추출 : .text

```
news_titles = driver.find_elements(By.CSS_SELECTOR, ".news_tit")
for i in news_titles:
    title = i.text
    print(title)
```

– 링크 추출 : get_attribute('href')

```
for i in news_titles:
    href = i.get_attribute('href')
    print(href)
```



_자동화 프로그래밍 - 웹 크롤링 실습 3

검색 키워드의 뉴스 검색 하기

뉴스 탭 클릭

```
driver.find_element(By.CSS_SELECTOR, '#lnb > div.lnb_group > div >  
div.lnb_nav_area._nav_area_root > div > div.api_btn_page._nav_btn_root >  
div.btn_next._next > a > span > i').click()  
time.sleep(1)
```

```
driver.find_element(By.CSS_SELECTOR, '#lnb > div.lnb_group > div >  
div.lnb_nav_area._nav_area_root > div > div.api_flicking_wrap._conveyer_root > div:nth-  
child(7) > a').click()  
time.sleep(3)
```

텍스트와 링크 가져오기

```
news_titles = driver.find_elements(By.CSS_SELECTOR, ".news_tit")  
print(news_titles)
```

```
for i in news_titles :  
    title = i.text  
    href = i.get_attribute('href')  
    print(title , " : ", href)
```



RPA 실습

_FastAPI

Fast API Form 응용 코딩

- 클라이언트 코딩(Front)
- 서버 코딩 (Back)

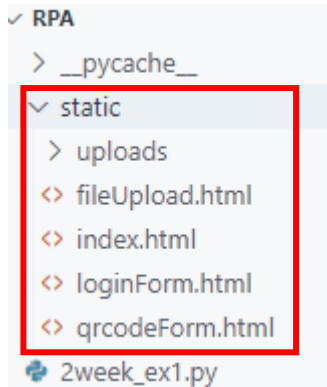
1. 클라이언트 코딩

HTML 파일 서비스 설정

```
from fastapi.staticfiles import StaticFiles
```

```
app.mount("/", StaticFiles(directory="static", html=True), name="static")
```

static 폴더안에 html 파일 생성

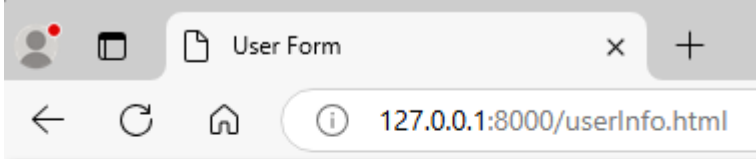


_FastAPI

User Form 편집

- Form 엘리먼트 추가

```
<form action="/user" method="post">  
  <span>학과</span>  
  <input type="text" name="major"><br>  
  <span>이름</span>  
  <input type="text" name="name"><br>  
  <span>학번</span>  
  <input type="text" name="studentcode" ><br>  
  <input type="submit" name="submit" value="제출">  
</form>
```



학과	의료정보과
이름	홍길동
학번	20000
제출	



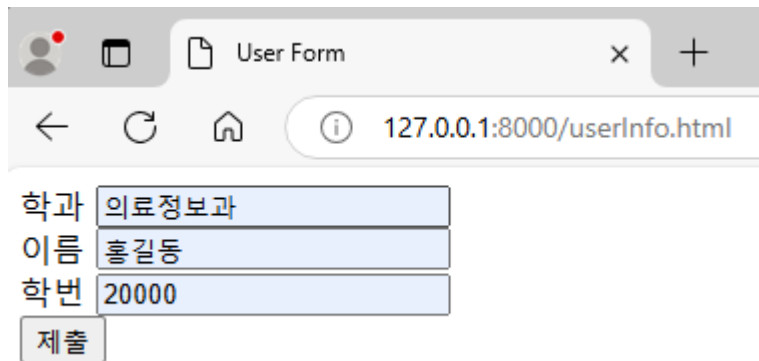
_FastAPI

2. 서버 코딩 : Form 요청에 대응하는 API

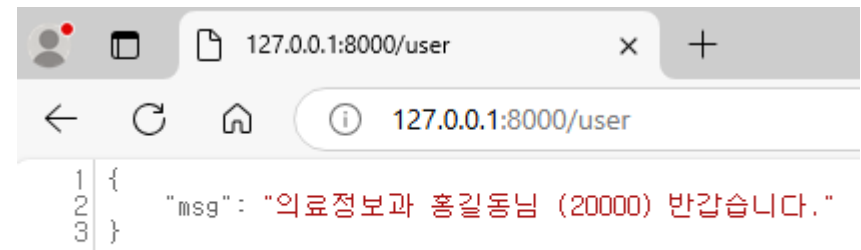
```
from fastapi import Form
```

```
@app.post("/user")
```

```
async def read_user_form(name: str = Form(...), studentcode: str = Form(...), major: str = Form(...)):  
    return {"msg": f"{major} {name}님 ({studentcode}) 반갑습니다."}
```



학과	의료정보과
이름	홍길동
학번	20000
<input type="button" value="제출"/>	



```
1 {  
2   "msg": "의료정보과 홍길동님 (20000) 반갑습니다."  
3 }
```



_실습

실습 문제 : 덧셈 계산기 만들기

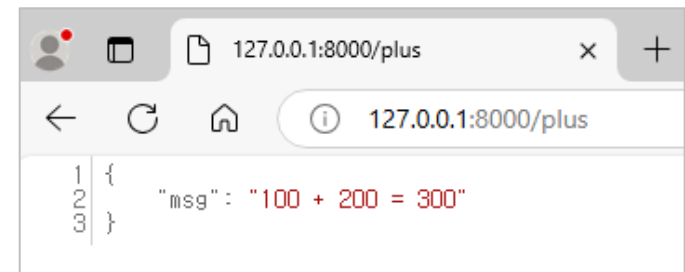
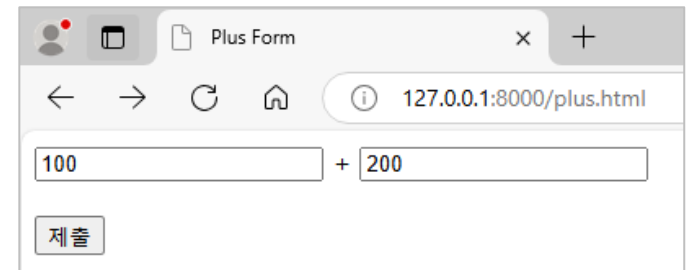
아래 그림처럼 두 수 입력 후 덧셈 결과를 출력하도록 API 함수를 만드시오.

plus.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Plus Form</title>
</head>
<body>

  <form action="/plus" method="post">
    <input type="text" name="num1">
    <span> + </span>
    <input type="text" name="num2"><br><br>
    <input type="submit" name="submit" value="제출">
  </form>

</body>
</html>
```



_실습

실습 문제 : 덧셈 계산기 만들기

아래 그림처럼 두 수 입력 후 덧셈 결과를 출력하도록 API 함수를 만드시오.

```
@app.post("/plus")
async def plus_form(num1: int = Form(...), num2 : int = Form(...)):
    result = num1 + num2
    return {"msg": f"{num1} + {num2} = {result}"}
```

