

# 주식 시세 현황 분석



소프트웨어융합대학원  
진혜진



**1. 프로젝트 개요**

**2. 웹 기본 지식 이해하기**

**3. 웹 스크래핑**

**4. 데이터 수집 및 가공**

**5. 데이터 시각화**



# 웹 스크래핑

# 1. 웹 스크래핑 대상 찾기

## ■ 주식 종목 선택하기

- 실제로 사용할 웹사이트를 대상으로 HTML 페이지를 주고받는 방식 학습
- 금융 지표 확인 사이트 : 네이버 증권(<https://finance.naver.com>) 접속
  - 오른쪽 하단 → '인기검색 종목' : '삼성전자'를 선택
  - ✓ 조회 시점에 따라 순위는 달라질 수 있음

← → ↻ 🔍 finance.naver.com

**N pay 증권**

증권 홈    국내증시    해외증시

최근조회종목

MYSTOCK

삼성전자 ☒  
55,100 ▲ 0.73%

인기 검색종목

더보기

1. 상지건설	43,400	↑ 10,000
2. 삼성전자	55,200	▲ 500
3. 현대로템	115,300	▲ 9,700
4. SK하이닉스	173,400	▼ 600
5. 포바이포	17,130	↑ 3,950

# 1. 웹 스크래핑 대상 찾기

## ■ 일별 시세 찾기

- 선택한 종목(삼성전자)의 정보가 열림
  - 주소 창에 현재 페이지의 웹 주소 :
  - (<https://finance.naver.com/item/main.naver?code=005930>) 가 표시됨
- 주가 그래프 아래 [시세] 탭을 클릭→ 제일 하단에 '일별 시세' 데이터가 보임

삼성전자 005930 코스피 2025.4.17		
KRX		
55,100		
전일대비 ▲400   +0.73%		
종합정보		
시세		
주요시세		
현재가	55,100	매
전일대비	▲ 400	매
등락률(%)	+0.73%	전
거래량	8,477,767	시
거래대금(백만)	465,863	고
액면가	100원	저
상한가	71,100	전

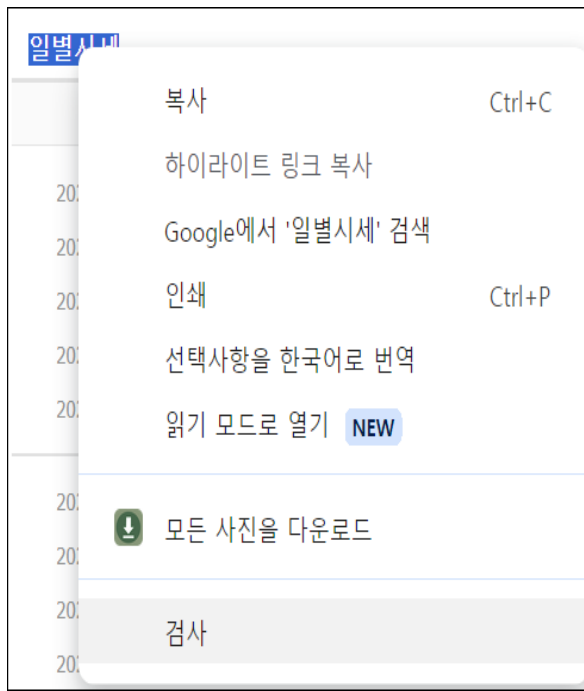
### 일별시세

날짜	종가	전일비	시가	고가	저가
2025.04.17	55,100	▲ 400	54,700	55,300	54,500
2025.04.16	54,700	▼ 1,900	56,000	56,200	54,500
2025.04.15	56,600	▲ 400	56,300	57,100	56,200
2025.04.14	56,200	▲ 1,000	56,300	56,700	55,800
2025.04.11	55,200	▼ 1,200	55,600	55,700	54,800
2025.04.10	56,400	▲ 3,400	56,600	56,700	55,000
2025.04.09	53,000	▼ 500	53,300	54,300	52,900
2025.04.08	53,500	▲ 300	55,000	55,300	53,300
2025.04.07	53,200	▼ 2,900	53,300	54,100	53,100
2025.04.04	56,100	▼ 1,500	56,200	58,200	55,700

# 1. 웹 스크래핑 대상 찾기

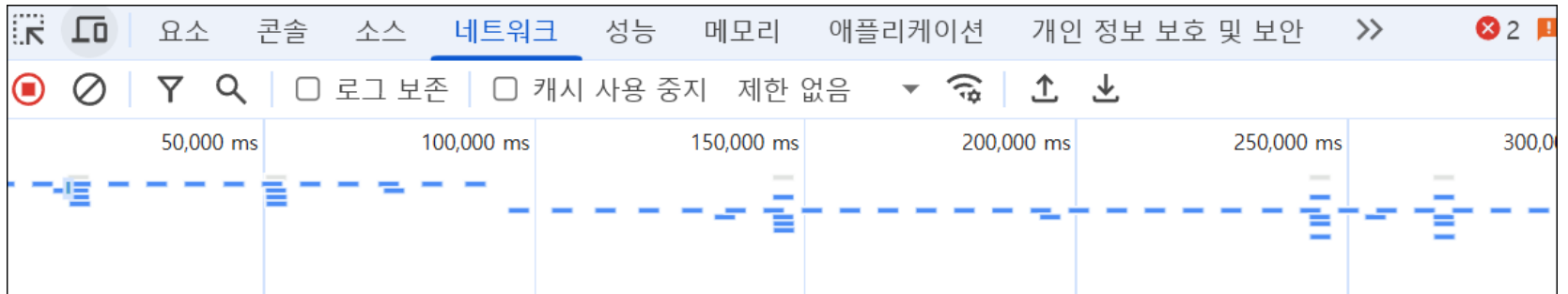
## ■ 일별 시세만 별도로 업데이트하는 URL 주소 찾기

- 웹 브라우저의 개발자 도구를 이용하여 추출
  - '일별 시세' 섹션에서 마우스 오른쪽 버튼을 클릭 → [검사] 또는 <Ctrl> + <Shift> + <I> 입력
  - 개발자 도구 상단 → [Network] 탭 클릭
  - 페이지 번호 '2'를 클릭 → 개발자 도구의 [Network] 탭 하단에서 웹 요청 정보들을 확인할 수 있음



일별시세		
날짜	종가	전일비
2025.04.17	55,100	▲ 400
2025.04.16	54,700	▼ 1,900
2025.04.15	56,600	▲ 400
2025.04.14	56,200	▲ 1,000
2025.04.11	55,200	▼ 1,200
2025.04.10	56,400	▲ 3,400
2025.04.09	53,000	▼ 500
2025.04.08	53,500	▲ 300
2025.04.07	53,200	▼ 2,900
2025.04.04	56,100	▼ 1,500
1   2   3   4   5   6		

# 1. 웹 스크래핑 대상 찾기



이름
sise_day.naver?code=005930&page=2
<input checked="" type="checkbox"/> newstock.css
<input checked="" type="checkbox"/> common.css
<input checked="" type="checkbox"/> finance_header.css
<input checked="" type="checkbox"/> layout.css
<input checked="" type="checkbox"/> main.css
<input checked="" type="checkbox"/> newstock2.css
<input checked="" type="checkbox"/> newstock3.css
<input checked="" type="checkbox"/> world.css
▶ bu_pgarR.gif
▶▶ bu_pgarRR.gif
<input checked="" type="checkbox"/> jindo.min.ns.1.5.3.euckr.js
◀◀ bu_pgarLL.gif
<input checked="" type="checkbox"/> ntm.js
<input checked="" type="checkbox"/> nLog.js
spr_bu.gif
<input checked="" type="checkbox"/> ntm_d1e43d133ced.js
<input checked="" type="checkbox"/> nlog.js

헤더	페이로드	미리보기	응답	시작점	타이밍	쿠키
▼ 일반						
요청 URL:	https://finance.naver.com/item/sise_day.n... 30&page=2					
요청 메서드:	GET					
상태 코드:	● 200 OK					
원격 주소:	117.52.137.138:443					
리퍼러 정책:	unsafe-url					
▼ 응답 헤더						
Cache-Control:	no-cache					
Content-Encoding:	gzip					
Content-Language:	ko					
Content-Type:	text/html; charset=EUC-KR					
Date:	Thu, 17 Apr 2025 06:02:26 GMT					
Expires:	Thu, 01 Jan 1970 00:00:00 GMT					
Referrer-Policy:	unsafe-url					
Server:	nfront					
Set-Cookie:	JSESSIONID=FEB10E97855347807E34788...					

# 1. 웹 스크래핑 대상 찾기

## ■ 일별 시세만 별도로 업데이트하는 URL 주소 찾기

- 상세 정보 확인하기

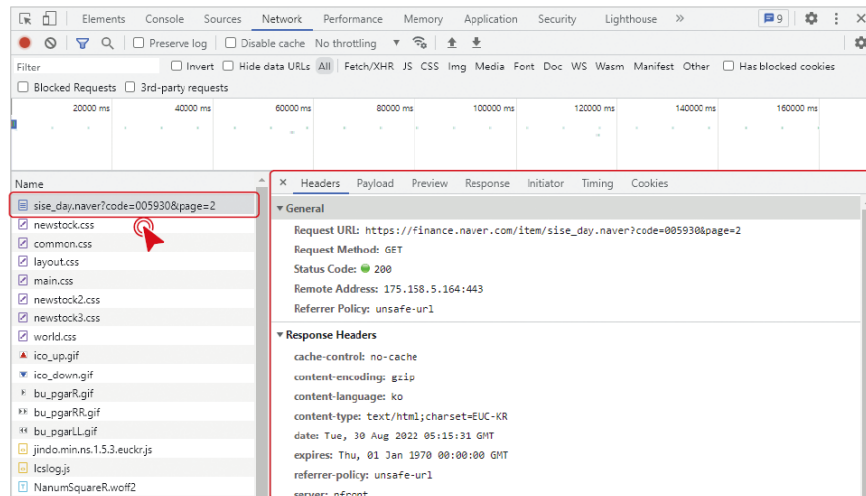
→ HTML 페이지를 의미하는 아이콘(📄)과 Name : 'sise\_day.naver?code=' 행을 찾아 클릭

→ 상세한 정보를 확인할 수 있는 창이 오른쪽에 열림

→ 제일 먼저 보이는 내용은 [Headers] 탭에서 확인할 수 있는 웹 요청 기본 정보임

- 확인할 수 있는 웹 요청 → 웹 스크래핑을 수행할 대상

→ 웹 스크래핑을 제대로 수행하려면 [Headers] 탭에 있는 정보를 이해해야 함



웹 요청 기본 정보 확인하기



## 2. 웹 페이지를 주고받는 규약, HTTP

### ■ 웹 사이트 기본 동작 방식

- '일별 시세' 데이터를 업데이트한 과정
  - ① 웹 브라우저에서 페이지 번호 클릭
  - ② 웹 페이지 데이터 변경
    - 이 과정은 간단해 보이지만 꽤 복잡한 단계를 거쳐 진행됨
    - 웹에서 데이터를 주고받는 데는 HTTP라는 프로토콜이 사용됨

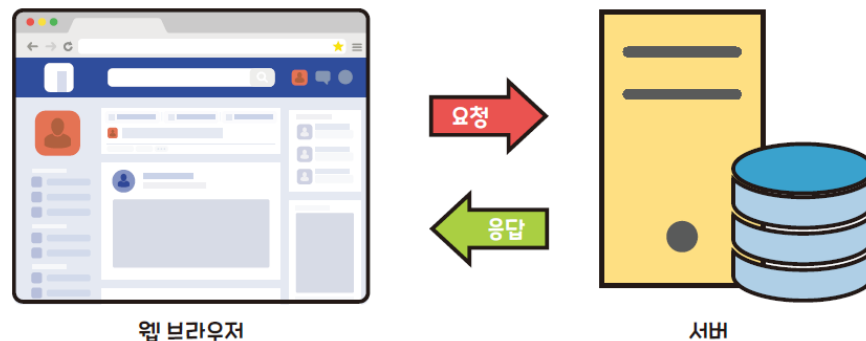
### ■ HTTP(Hypertext Transfer Protocol)

- 인터넷을 통해 데이터를 주고 받기 위해 미리 정의한 약속 : 일종의 규약을 의미함
  - 앞에서 살펴본 데이터 수집 대상의 요청은 'http://' 혹은 'https://'로 시작함
  - 이 주소는 웹 브라우저 주소 창에 입력하는 URL 주소와 같음

## 2. 웹 페이지를 주고받는 규약, HTTP

### ■ 웹 브라우저와 서버의 통신 방식

- 웹 브라우저 : 사용자가 HTTP 프로토콜을 사용한 요청(Request)을 서버로 전달  
→ 응답(Response)을 받아 화면에 출력하는 역할을 함
- 웹 브라우저와 서버의 응답-요청 예시
  - ① 화면에서 페이지 번호 '2'를 클릭→ 웹 브라우저는 마우스 클릭 이벤트를 감지
  - ② HTML에서 숫자 2에 미리 정의해놓은 주소로 서버 요청을 보냄
  - ③ 서버는 전송된 요청을 처리→ 원하는 정보를 웹 브라우저에 응답으로 보냄
  - ④ 서버로 부터 응답받은 웹 브라우저는 전송 받은 데이터를 화면에 출력



웹 브라우저와 서버의 통신 방식

## 2. 웹 페이지를 주고받는 규약, HTTP

### ■ HTTP 요청 주소 확인하기

- 웹 브라우저 개발자 도구의 웹 요청 상세 정보 확인
- 첫 번째 탭 [Headers] : 웹 스크래핑 정보 → HTTP 요청을 보낸 주소(Request URL)

x 헤더 페이로드 미리보기 응답 시작점 타이밍 쿠키	
▼ 일반	
요청 URL:	https://finance.naver.com/item/sise_day.naver?code=005930&page=2
요청 메서드:	GET
상태 코드:	● 200
원격 주소:	117.52.137.138:443
리퍼러 정책:	unsafe-url
▼ 응답 헤더	
Cache-Control:	no-cache
Content-Encoding:	gzip

- 페이지 번호 '2'를 클릭해 요청한 웹 주소는 다음과 같음

[https://finance.naver.com/item/sise\\_day.naver?code=005930&page=2](https://finance.naver.com/item/sise_day.naver?code=005930&page=2)

## 2. 웹 페이지를 주고받는 규약, HTTP

### ■ Header: HTTP 요청 헤더 및 user-agent

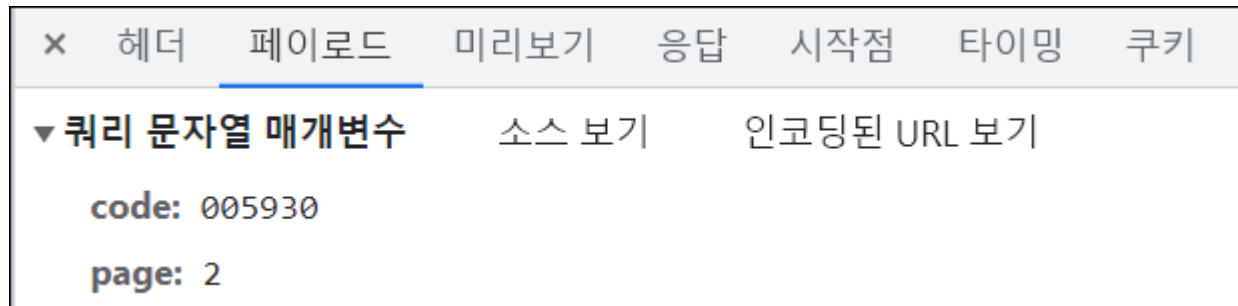
- 웹 스크래핑 → 웹 요청을 웹 브라우저가 아니라 파이썬 소스 코드로 보내야 함
- 이 때문에 요청 헤더에 무슨 값이 있는지 살펴보는 것이 좋음
- 요청을 보내는 주체가 무엇인지 설명하는 user-agent 값이 중요함

▼ 요청 헤더	
:Authority:	finance.naver.com
:Method:	GET
:Path:	/item/sise_day.naver?code=005930&page=2
:Scheme:	https
Accept:	text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding:	gzip, deflate, br
Accept-Language:	ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
Cookie:	NNB=VZC5GTPAN5MGI; nx_ssl=2; ASID=3ae63c2a00000188000edbf00000059; naver_stock_codeList=005930%7C; page_uid=ibx3+dprvTVssjvdc9CssssstnN-393415; _naver_usersession_=sabsx6oOISDvC6z7pRb7TA==; JSESSIONID=D08D9F3C97B6DFDD1E0416F9F847271D
Referer:	https://finance.naver.com/item/sise_day.naver?code=005930
Sec-Ch-Ua:	"Google Chrome";v="113", "Chromium";v="113", "Not-A.Brand";v="24"
Sec-Ch-Ua-Mobile:	?0
Sec-Ch-Ua-Platform:	"Windows"
Sec-Fetch-Dest:	iframe
Sec-Fetch-Mode:	navigate
Sec-Fetch-Site:	same-origin
Sec-Fetch-User:	?1
Upgrade-Insecure-Requests:	1
User-Agent:	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.0.0 Safari/537.36

## 2. 웹 페이지를 주고받는 규약, HTTP

### ■ Payload: HTTP 요청 페이로드

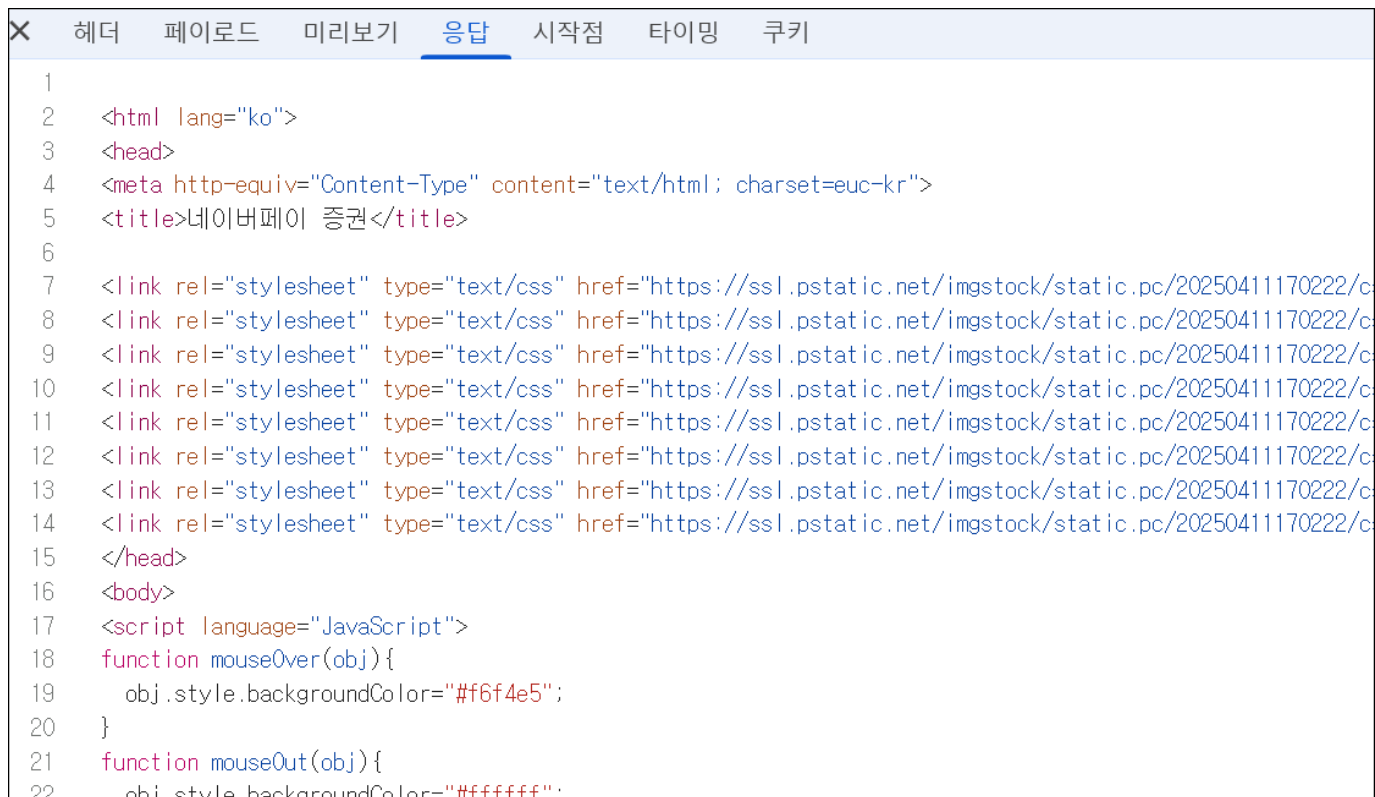
- 페이로드(Payload) : HTTP 요청을 보낼 때 함께 전송하는 데이터
- 개발자 도구 [Payload] 탭을 클릭
  - 필요한 정보를 확보하기 위해 주식 코드와 표기할 페이지가 전송된 것을 확인할 수 있음
  - ✓ 이 정보는 웹 스크래핑을 수행할 때 어떤 데이터를 함께 보내야 할지 이해하는 데 도움이 됨



## 2. 웹 페이지를 주고받는 규약, HTTP

### ■ Response: HTTP 응답

- 서버가 웹 브라우저로 보낸 응답 내용을 확인하기
- 웹 브라우저 개발자 도구의 [Response] 탭 클릭 → 서버에서 전송한 HTML 페이지  
→ 서버가 데이터를 처리해 웹 브라우저로 보낸 응답의 실체임



The screenshot shows the 'Response' tab in a web browser's developer tools. The code is HTML, starting with a DOCTYPE declaration and a head section containing meta tags for content type and charset, and a title. The body contains several link tags for stylesheets and a script tag for JavaScript. The JavaScript code defines two functions, mouseOver and mouseOut, which modify the background color of an object.

```
1
2 <html lang="ko">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=euc-kr">
5 <title>네이버페이 증권</title>
6
7 <link rel="stylesheet" type="text/css" href="https://ssl.pstatic.net/imgstock/static.pc/20250411170222/c
8 <link rel="stylesheet" type="text/css" href="https://ssl.pstatic.net/imgstock/static.pc/20250411170222/c
9 <link rel="stylesheet" type="text/css" href="https://ssl.pstatic.net/imgstock/static.pc/20250411170222/c
10 <link rel="stylesheet" type="text/css" href="https://ssl.pstatic.net/imgstock/static.pc/20250411170222/c
11 <link rel="stylesheet" type="text/css" href="https://ssl.pstatic.net/imgstock/static.pc/20250411170222/c
12 <link rel="stylesheet" type="text/css" href="https://ssl.pstatic.net/imgstock/static.pc/20250411170222/c
13 <link rel="stylesheet" type="text/css" href="https://ssl.pstatic.net/imgstock/static.pc/20250411170222/c
14 <link rel="stylesheet" type="text/css" href="https://ssl.pstatic.net/imgstock/static.pc/20250411170222/c
15 </head>
16 <body>
17 <script language="JavaScript">
18 function mouseOver(obj){
19     obj.style.backgroundColor="#f6f4e5";
20 }
21 function mouseOut(obj){
22     obj.style.backgroundColor="#ffffff";
```