

# 파이썬을 활용한 실전 웹크롤링 CAMP

2018. 04. 28  
2주차 강의안

## 금일 강의의 목적

1. 서버 / 클라이언트 / 브라우저의 관계도 살펴보기
2. 우리가 정보를 요청하는 방식 두가지 GET / POST 이해하기
3. 웹 브라우저의 개발자도구를 사용하여 웹 문서 확인하기
4. 웹 문서의 구조를 이해하고, CSS / tag / JS 개념 학습하기
5. selenium을 사용하여 브라우저 열어보기

## 1. 이론

1. 서버 / 클라이언트 / 브라우저 관계 이해하기
2. GET / POST 방식 이해하기
3. 브라우저가 받는 HTML 웹 문서에 대한 이해
4. python에서 서버에 요청하는 방식

## 2. 실습

1. 실제 웹사이트에서 개발자도구를 통해 HTML문서 살펴보기
2. CSS / tag / JS등에 대한 이해
3. Requests를 활용한 웹 요청
4. Selenium을 활용한 웹 브라우저 접근

## 1. 서버 / 클라이언트 / 브라우저 관계 이해하기



### 서버 (웹서버)

서버란 클라이언트(사용자)에게 **네트워크를 통해 서비스를 제공**하는 **물리적인 컴퓨터**를 의미합니다. 대부분의 인터넷 서비스 업체는 서버를 통해 서비스를 제공합니다. 연말정산 시즌에 국세청 홈페이지가 다운되는 것은 바로 국세청이 이용하는 서버의 가용자원을 넘어서 사용자들의 접근이 원인이 되는 경우가 많습니다. 흔히들 말하는 ‘서버가 다운됐다’에서 말하는 서버가 웹서버입니다.



## 1. 서버 / 클라이언트 / 브라우저 관계 이해하기



### 클라이언트 (사용자)

클라이언트란 **서버에 접속하기 위한 프로그램**을 의미합니다.

게임을 할때 클라이언트를 사용하여 접근하는데, 이때 게임 클라이언트는 게임 회사의 서버에 접근하여 다른 사람들과 온라인 게임을 즐기는데 사용되는 프로그램 입니다.

물론 웹서버에 접근하는 우리에게 클라이언트 프로그램은 **브라우저**가 됩니다.

IE(internet explorer) / Chrome / FireFox 등의 다양한 브라우저가 존재하며 각 브라우저는 사용자의 의도에 따라 웹서버에 접근할 수 있게 해줍니다.

## 1. 서버 / 클라이언트 / 브라우저 관계 이해하기



### 웹 브라우저

웹 브라우저란 웹 사용자가 URL등을 사용하여 웹서버에 접근할 때 사용되는 프로그램입니다. 쉽게 설명하자면, **주소를 알려주면 알아서 찾아가 택배를 받아오는 심부름꾼**으로 생각하시면 됩니다.

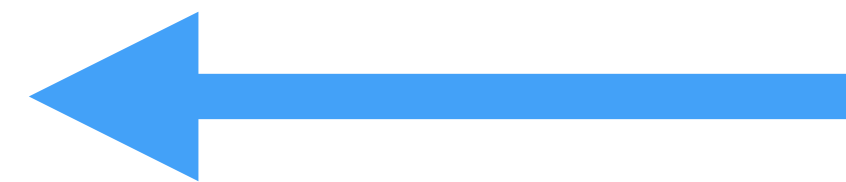
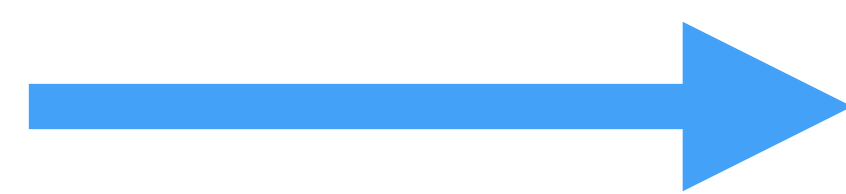
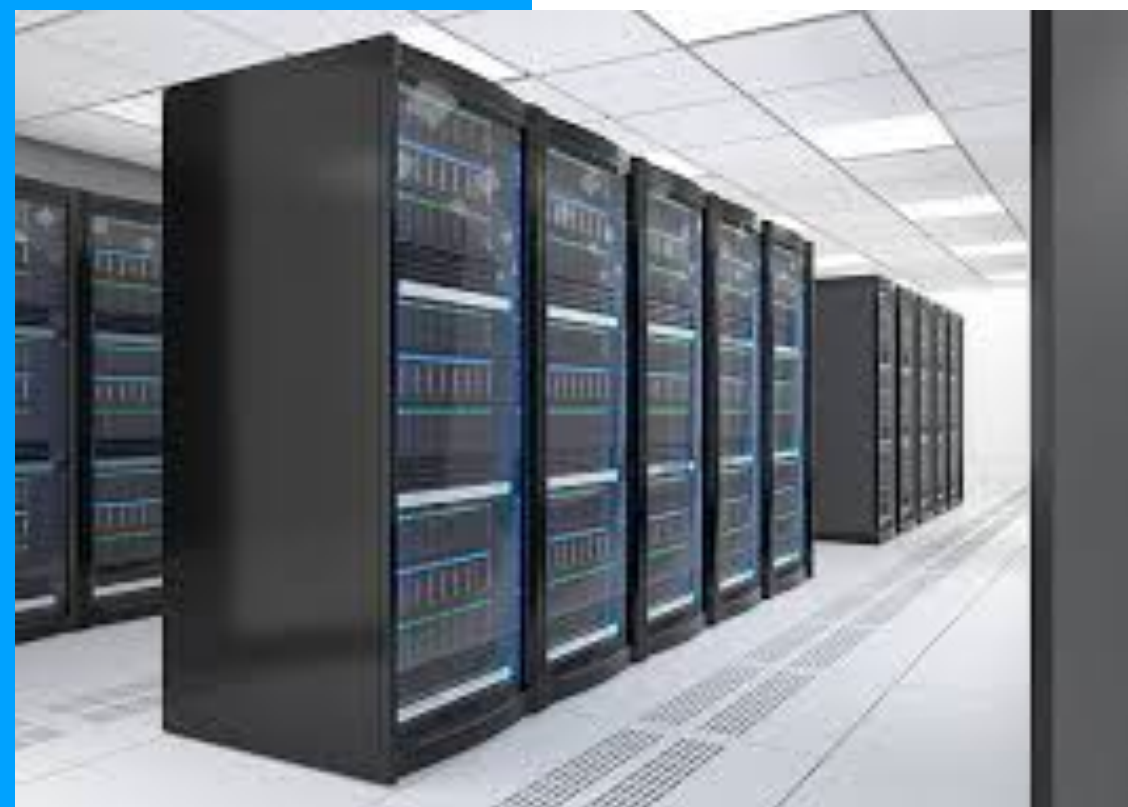
이러한 웹브라우저는 웹서버에서 사용자가 요청한 정보를 응답받아 실제 사용자가 보고 읽을 수 있는 **이미지, 텍스트 등으로 변환해주는 역할**도 수행합니다.



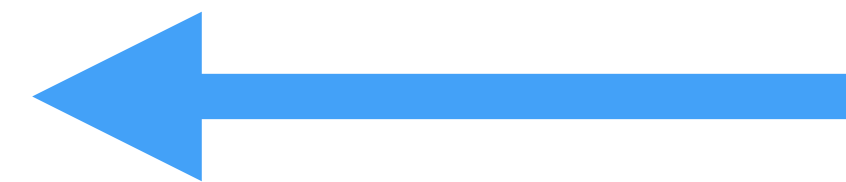
## 1. 서버 / 클라이언트 / 브라우저 관계 이해하기

브라우저에 **응답**을 보냄

브라우저에 표현된 **응답**을 확인



서버에 **요청**을 전달

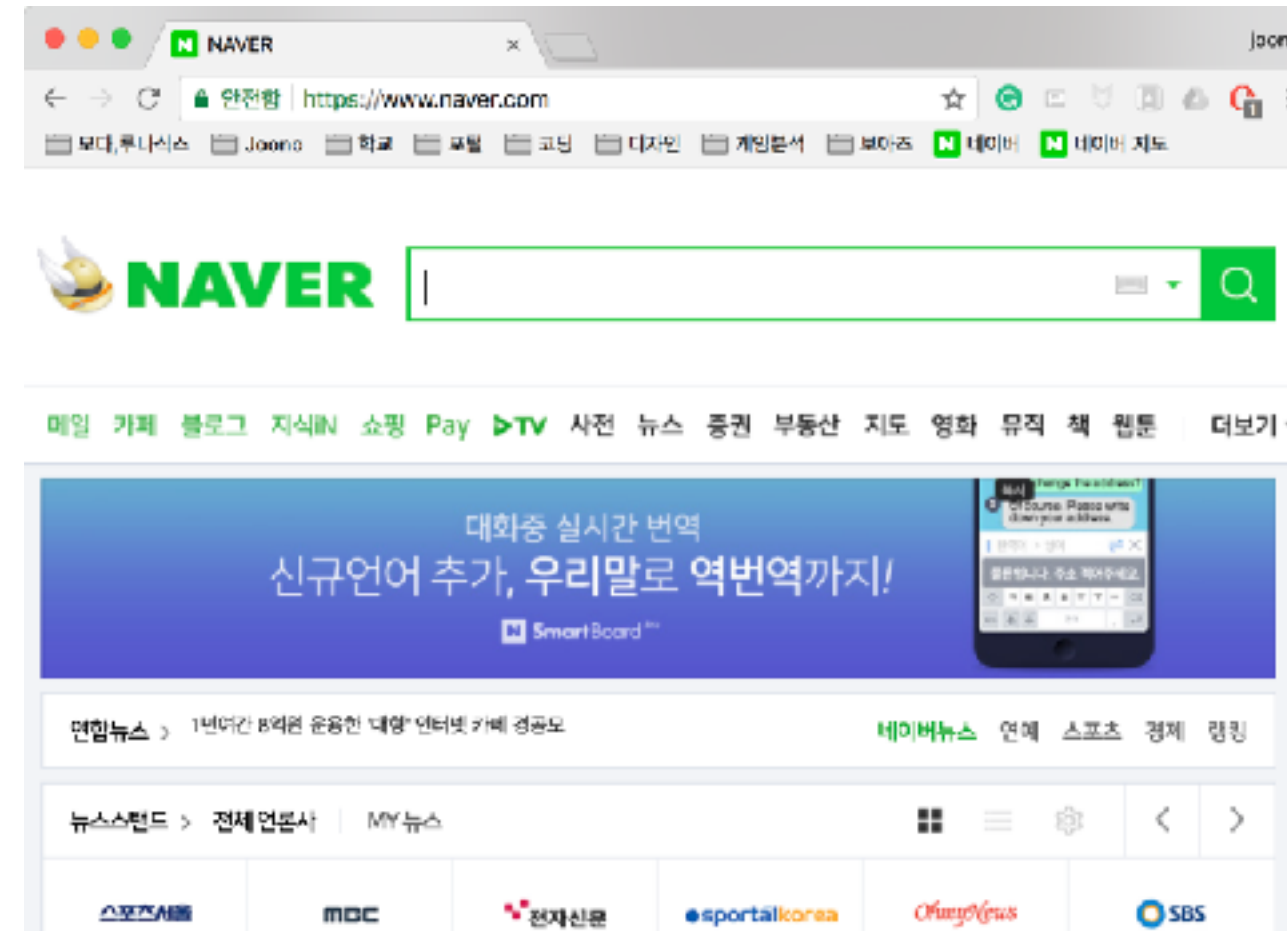


정보 **요청**

## 1. 서버 / 클라이언트 / 브라우저 관계 이해하기



크롬에서 <https://www.naver.com> 접근



네이버 웹서버에 요청을 전달

브라우저에 표현된 응답

요청에 대한 응답을 브라우저가 받음



실제 네이버의 인터넷 센터



## 2. GET / POST 방식 이해하기



웹서버로 요청을 하는  
여러가지 방식 중 두가지



## 2. GET / POST 방식 이해하기



GET 방식

심부름꾼에게 주소를 알려주면서 **주소안에 원하는 정보의 위치 등을 포함시켜**  
심부름을 보내는 것

Ex)

<http://comic.naver.com/webtoon/list.nhn?titleId=183559&weekday=mon>  
'신의 탑'이라는 웹툰의 URL 주소입니다. titleId와 weekday라는 정보를  
URL에 실어서 보내서 보냅니다.

## 2. GET / POST 방식 이해하기



POST 방식

POST방식은 이름에서도 알 수 있듯이 무언가를 게시할 때 쓰는 방식입니다. 이 요청은 게시물 작성과 같은 요청을 할 때 수행하는데, 이때 header라는 곳에 정보를 실어서 보내게 됩니다.

Ex)

네이버 카페 글 게시 요청

## 3. 브라우저가 받는 HTML 웹 문서에 대한 이해

### HTML

HyperText Markup Language의 줄임말입니다.

웹 브라우저가 서버에서 받는 정보는 HTML이라는 웹 문서로 이것을 브라우저가 해석하여 우리가 보는 실제 웹사이트로 구성합니다.

이러한 HTML 웹문서는 대부분 안에 CSS라는 언어와 JavaScript를 포함하고 있습니다.

HTML은 우리가 보는 웹사이트의 제목, 본문, 버튼, 입력칸 등의 모든 **엘리먼트(element)**를 **태그(Tag)**와 **속성(attribute)**으로 표현하고, 그 **위치나 크기, 색상 등을 CSS언어**로 표현합니다.

또한 사용자가 URL의 변화 없이, 즉 다른 웹문서를 불러오지 않고 **같은 문서 안에서 다른 정보를 사용자의 입력에 따라 표현하기 위해 JavaScript를 사용**하기도 합니다.



## 3. 브라우저가 받는 HTML 웹 문서에 대한 이해

```
<!doctype html>
```

```
<html>
```

```
<head>
```

```
<title>Hello HTML</title>
```

```
</head>
```

```
<body>
```

```
<p>Hello World!</p>
```

```
<a href="http://example.org"></a>
```

```
</body>
```

```
</html>
```

기본 구조

〈태그〉

속성 = “속성값”

〈/태그〉

## 3. 브라우저가 받는 HTML 웹 문서에 대한 이해

### 크롤링을 할때 주로 접근하는 태그

div : 공간을 의미합니다. 엘리먼트를 다른 것과 구분하기 위해 사용합니다.

p : 단락을 의미합니다. 대부분 기사의 본문이나 댓글의 본문 등에 사용됩니다.

a : **링크**를 의미합니다. 대부분 **href라는 속성**을 가지고 있는데 이 속성이 링크와 연결된 사이트의 주소를 가지고 있습니다.

td : 여러개의 게시판 목록처럼 **테이블과 같은 것들의 하나의 셀**을 의미합니다.  
주로 웹페이지의 엘리먼트를 공간을 구분지어 구성해 놓은 사이트에서 많이 사용합니다.

img : 말그대로 이미지를 나타내는 태그입니다. 이때 **src라는 속성으로 이미지의 주소**를 표현해 놓습니다.  
주소를 통해 이미지에 접근이 가능합니다.

span : 그 자체로는 무의미 하지만, 내부의 속성값을 사용하여 박스공간 등을 표현할 때 사용합니다.

## 3. 브라우저가 받는 HTML 웹 문서에 대한 이해

### CSS

Cascading Style Sheets의 줄임말입니다.

HTML 문서의 디자인을 담당하는 언어입니다.

웹페이지의 배경색을 정하거나, 텍스트의 폰트, 폰트색 등을 정할때 사용합니다.

```
p{  
    font-size: 110%;  
    font-family: garamond, sans-serif;  
}  
h2{  
    color: red;  
    background: white;  
}
```

위와 같이 CSS는 중괄호로 표현하며 **속성 : 속성값;** 으로 표현합니다.

## 3. 브라우저가 받는 HTML 웹 문서에 대한 이해

### JavaScript

Java와는 다른 언어입니다.

**<script>라는 태그에 싸여서 HTML 문서 안에서 작동하는 코드**입니다.

HTML 웹 문서는 정적인 문서(하나의 URL에는 정해진 콘텐츠가 있어서 어떤 경우에도 바뀌지 않고 표현됨)와 동적인 문서(같은 URL로 접근해도 URL안에서 다른 정보를 표현할 수 있는 문서)가 있습니다.

동적인 문서는 웹 브라우저에서 사용자의 액션에 반응하여 이미지 / 콘텐츠를 변경하거나 액션에 맞는 반응을 표현하는 문서입니다. 이러한 동적인 문서를 구성하기 위해 사용되는 언어이며, 수업에서는 JavaScript 코드를 짜기보다는 읽고, 이해하여 웹문서의 동적인 부분을 이해하는데 초점을 맞춥니다.



## 4. python에서 서버에 요청하는 방식

### Requests 패키지

파이썬의 기본 내장 패키지. get방식과 post 방식 둘다 사용가능하고, 따로 웹드라이버가 필요없습니다. 하지만 실제 브라우저를 통한 접근이 아니기 때문에 크롤링bot을 막기 위한 제어장치를 가지고 있는 웹에 접근할때는 여러가지 추가 코딩이 필요하다는 단점이 있습니다.

### Selenium 패키지

웹 개발자들이 만들어진 웹페이지를 테스트 하기위해 만들어진 패키지입니다. 웹을 코드로 작동시킬수 있기 때문에 크롤링에 사용하기에 매우 편리합니다. 하지만 post방식을 사용할때 매우 코딩이 어렵다는 단점이 있습니다. 실제 브라우저를 통해 접근하기 때문에 크롤링 bot을 막는 제어장치를 피할수 있고, 크롤링 도중에 에러가 난 경우 브라우저를 통해 에러의 원인을 살펴보고 수정하기 용이하다는 장점이 있습니다.



HTML 문서의 엘리먼트들의 태그와 속성값을 확인 가능하고,  
문서의 구조가 어떻게 짜여져 있는지 확인이 가능합니다.

또한 각 **엘리먼트들의 CSS 스타일도 확인이 가능합니다.**






# 1. 실제 웹사이트에서 개발자도구를 통해 HTML문서 살펴보기

NAVER 뉴스 | TV연예 | 스포츠 | 뉴스스탠드 | 날씨


뉴스홈 속보 정치 경제 사회 생활/문화 세계 IT/과학 오피니언 포토 TV 랭킹뉴스

신문 헤드라인 ▾ 저녁 방송 뉴스 ▾

이 시각 주요뉴스 최종 2018.04.26 04:03




공동선언문에 담길 '3대 의제'... 어디까지 와 있나?  
숨 가쁘게 달려온 1년... 27일, 한반도 항구적 평화 '첫걸음'  
'드루킹 출판사 절도사건' TV조선 압수수색, 기자들 반발에 무산  
삼성 반도체 공장 유해물질 조사 역부족... '먼지부'만 준 꼴  
'유령 ID' 무한 생성 가능... 네이버 '댓글 조작 대책' 무용지물



못 믿을 교육부... "비리 사학에 제보자 정보 넘겼다"  
경찰, '킹크랩' 서버 확보... "매크로와 같이 이용해 댓글 조작"  
장관이 업무시간에 '퇴폐 요가'... 아베 정권 또 악재  
미래에셋·삼성생명 정조준... 금감원 "지배구조 바꿔야"  
골다공증·척추 환자, 안마 의자 쓰다 '골절' 위험

정치 일반 | 국회/정당 | 청와대



'비핵화' 뺀 모든 의제 조율 마쳤다 동아일보  
안철수 "7년간 악성댓글 헤치며 살아와... 기득권 정치 바이러스 잡..." 동아일보

Elements Console Sources Network Performance Memory Application Security >> 1 X

<!DOCTYPE html>  
<html lang="ko" data-useragent="Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_13\_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/65.0.3325.181 Safari/537.36" class="gr\_news\_naver\_com">  
 >#shadow-root (open)  
 ><head>...</head>  
 ><body class="chrome" data-gr-c-s-loaded="true">  
 ><div id="wrap">  
 ><div id="da\_base"></div>  
 ><div id="da\_stake"></div>  
 ><div id="header" class="header">...</div>  
 ><div class="home\_timelate">...</div>  
 ><script type="text/javascript">...</script>  
 ><div class="main\_space"></div>  
 ><div id="container" class="main">  
 <hr>  
 ><div id="main\_content" class="main\_content main\_content\_new">  
 >><div class="main\_component droppable" id="today\_main\_news">...</div> == \$0  
 >><div class="main\_component droppable" id="section\_politics">...</div>  
 >><div class="main\_component droppable" id="section\_economy">...</div>  
 >><div class="main\_component droppable" id="section\_society">...</div>  
 >><div class="main\_component droppable" id="section\_life">...</div>  
 >><div class="main\_component droppable" id="section\_world">...</div>  
 >><div class="main\_component droppable" id="section\_it">...</div>  
 >><div id="home\_space" class="space" style="display:none"></div>  
 >><div id="home\_phantom" class="home\_move" style="left:500px; top:220px; display:none;">...</div>  
 >><div class="airs">...</div>  
 >>::after  
 </div>  
 ><div class="mainAside">...</div>  
 </div>  
 ><div class="index">...</div>  
 <hr>  
 ><div id="footer">...</div>  
 ><script type="text/javascript">...</script>  
 </div>  
 </body>  
</html>

Styles Computed Event Listeners DOM Breakpoints >>

Filter :hov .cls +

element.style {  
}  
>.main\_content\_new .main\_component { common.css:605  
 border-bottom: 1px solid #dedede;  
 margin: 0 0 15px 0;  
 padding-top: 9px;  
}>  
>.main\_component { common.css:479  
 position: relative;  
 margin-top: 16px;  
 background: #fff;  
}>  
>body, div, p, h1, h2, h3, h4, h5, h6, ul, ol, li, dl, dt, table, th, td, form, fieldset, legend, input, textarea, button { common.css:2  
 margin: 0;  
 padding: 0;  
}>  
>div { user agent stylesheet  
 display: block;  
}>  
>Inherited from div#container.main  
>#container { common.css:165  
 display: table;  
 table-layout: fixed;  
 width: 1080px;  
 margin: 0 auto;  
 text-align: left;  
}>  
>Inherited from div#wrap  
>#wrap { common.css:161  
 margin: 0 auto;  
 text-align: center;  
}>  
>Inherited from body.chrome  
>body { common.css:4  
 color: #000;  
 font-size: 12px;  
 text-align: center;  
 background: #fff;  
}>  
>body, table, table td, input, select, textarea { common.css:3  
 font-size: 12px;  
 font-family: "Helvetica Neue", "Apple SD Gothic Neo", "Malgun Gothic", "맑은 고딕", "Dotum", "돋움", "sans-serif";  
}>

div#today\_main\_news.main\_component.droppable





# 1. 실제 웹사이트에서 개발자도구를 통해 HTML문서 살펴보기

YouTube KR

검색

맞춤 동영상

심슨 짐보의 애인 사나를 사랑하게 된 바트와 드라마에 빠진 호머

심슨 에피소드

조회수 67만회 · 2주 전

박지성이 꼽은 기억에 남는 골, 환상적인 다이빙헤딩 결승골 • 맨유

HEON honey

조회수 22만회 · 10개월 전

[배그 APL] 시즌1 NTT 역전승 종합 5위에서 1위 등극! 파이널 진

[SexyPig]섹시피그

조회수 1.5만회 · 11시간 전

맛상무. 세상에서 가장비싼 라면 랍스타라면 리뷰

맛상무

조회수 34만회 · 9개월 전

카카오서버에서 일본인인척 해보았달ㅋㅋㅋ [배그랜덤매칭 10편]

스구용

조회수 31만회 · 1개월 전

투기장 12승 너무 쉽다 ^^

TV타요

조회수 1만회 · 8시간 전

더보기

TV라간 맞춤 채널

구독 8.5만

천상계 1티어 난입OP

16:31

이렐로 핵버스 태우기

29:58

리퍼궁 상시 발동하는 루난 루시만!

37:29

Elements

Console

Sources

Network

Performance

Memory

Application

3

3

✕

<!DOCTYPE html>

<html invert style="font-size: 10px;font-family: Roboto, Arial, sans-serif; background-color: #fafafa;" class="gr\_youtube\_com">

>#shadow-root (open)

><head>...</head>

...<body dir="ltr" data-gr-c-s-loaded="true"> == \$0

><script>...</script>

<!-- end of chunk -->

<script>if (window.ytcsi)

{window.ytcsi.tick("ai", null, '');}</script>

><ytd-app>...</ytd-app>

<script>if (window.ytcsi)

{window.ytcsi.tick("gcc", null, '');}</script>

<script>

window['ytInitialGuideDataPresent'] = true;

</script>

<script>if (window.ytcsi)

{window.ytcsi.tick("nc\_pj", null, '');}</script>

<script>if (window.ytcsi)

{window.ytcsi.tick("rsbe\_dpj", null, '');}

</script>

<script src="https://s.ytimg.com/yts/jsbin/desktop\_polymer-vflz45WUY/desktop\_polymer.js" type="text/javascript" name="desktop\_polymer/desktop\_polymer" class="js-httpsytimgcomytsjsbindesktop\_polymervflz45WUYdesktop\_polymerjs"></script>

<script>if (window.ytcsi)

{window.ytcsi.tick("rsae\_dpj", null, '');}

</script>

><script>...</script>

<!-- end of chunk -->

><div id="img-preload" style="display: none;">...</div>

><script>...</script>

><iron-iconset-svg style="display: none;">...</iron-iconset-svg>

<script>if (window.ytcsi)

{window.ytcsi.tick("pdc", null, '');}</script>

><div id="img-preload" style="display: none;">...</div>

><script>...</script>

><script>...</script>

><span style="display:none" id="legal-help-page">...</span>

<link rel="stylesheet" href="https://s.ytimg.com/yts/cssbin/www-main-desktop-watch-page-skeleton-2x-webp-vflS1WbUQ.css" name="www-main-desktop-watch-page-skeleton" class="css-httpswwwyoutubeocomytscssbinwwwmaindesktopwatchpageskeleton2xwebpvflS1WbUQcss">

<script>if (window.ytcsi)

{window.ytcsi.info("st", 1568, '');}</script>

</body>

Styles

Computed

Event Listeners

DOM Breakpoints

>>

Filter

:hov .cls +

element.style {

}

body {

padding: 0;

margin: 0;

}

body[Attributes Style] {

direction: ltr;

unicode-bidi: isolate;

}

body {

display: block;

margin: 8px;

}

Inherited from

html.gr\_youtube\_com

Style Attribute {

font-size: 10px;

font-family: Roboto, Arial, sans-serif;

background-color: #fafafa;

}

html:not([style-scope]):not(.style-scope) ?gl=KR&hl=ko:96 {

--yt-live-chat-action-panel-background-color: hsla(0, 0%, 93.3%, .4);

--yt-live-chat-action-panel-background-color-transparent: hsla(0, 0%, 97%, .8);

--yt-live-chat-primary-text-color: hsl(0, 0%, 6.7%);

--yt-live-chat-secondary-text-color: hsla(0, 0%, 6.7%, .6);

--yt-live-chat-tertiary-text-color: hsla(0, 0%, 6.7%, .4);

--yt-live-chat-disabled-icon-button-color: hsla(0, 0%, 6.7%, .2);

--yt-live-chat-picker-button-color: hsla(0, 0%, 6.7%, .4);

--yt-formatted-string-emoji-size: 24px;

}

html:not([style-scope]):not(.style-scope) ?gl=KR&hl=ko:90 {

--yt-button-margin: 0;

--yt-button-padding: 10px 16px;

--yt-button-border-radius: 2px;

}

html:not([style-scope]):not(.style-scope) ?gl=KR&hl=ko:78 {

--ytd-z-index-notification: 2024;

--ytd-z-index-miniplayer: 2025;

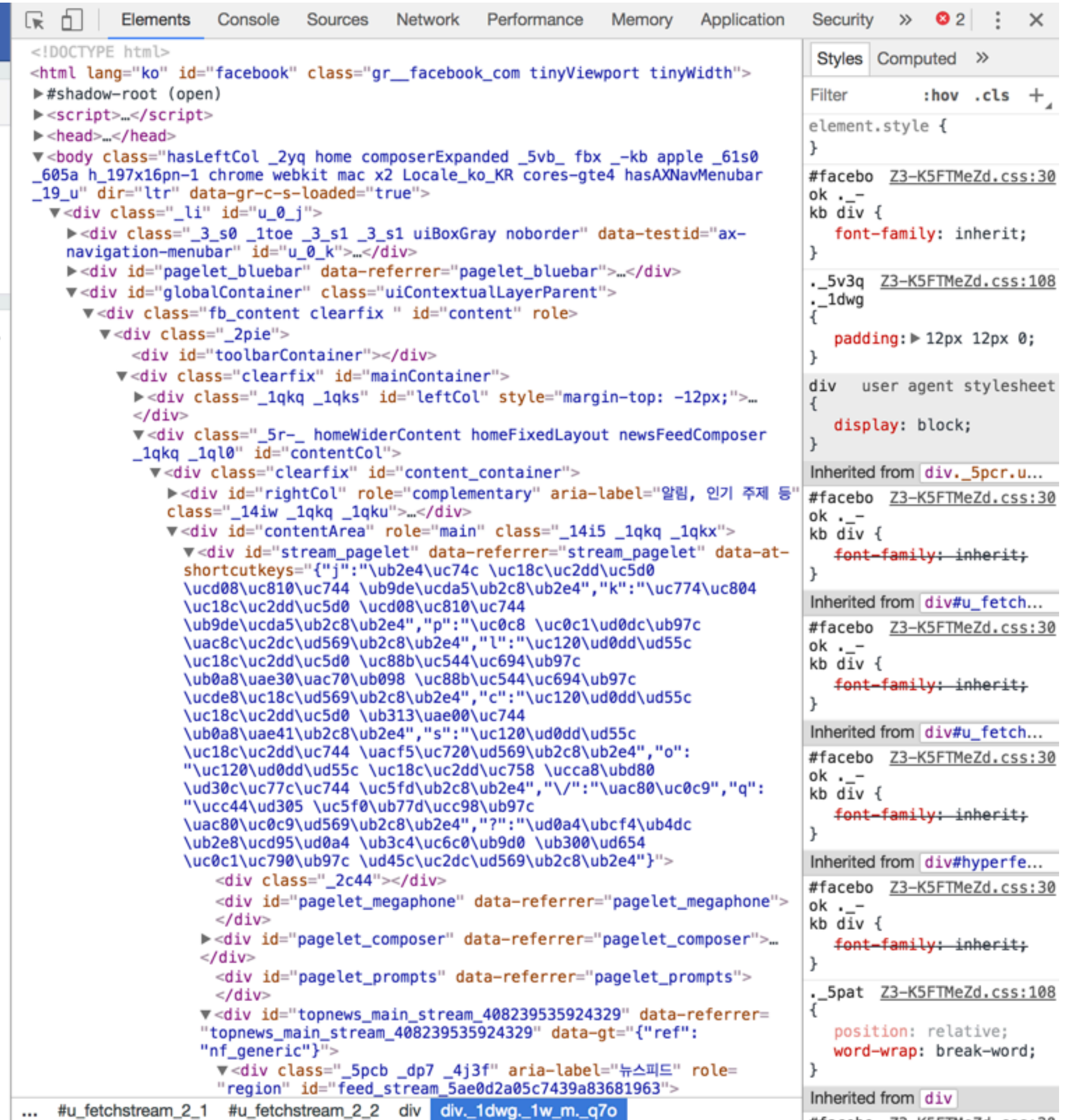
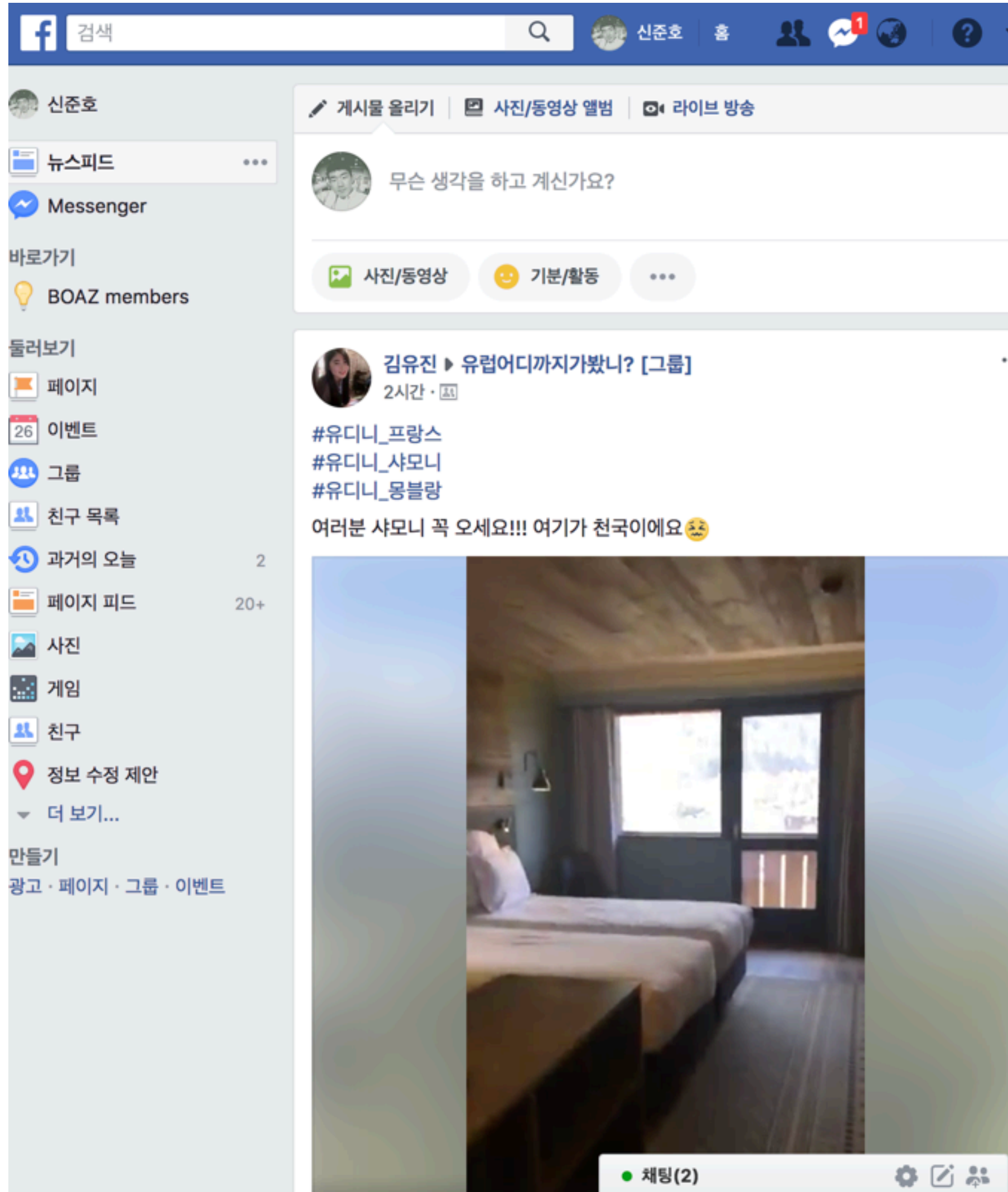
--ytd-thumbnail-height: 118px;

}





# 1. 실제 웹사이트에서 개발자도구를 통해 HTML문서 살펴보기







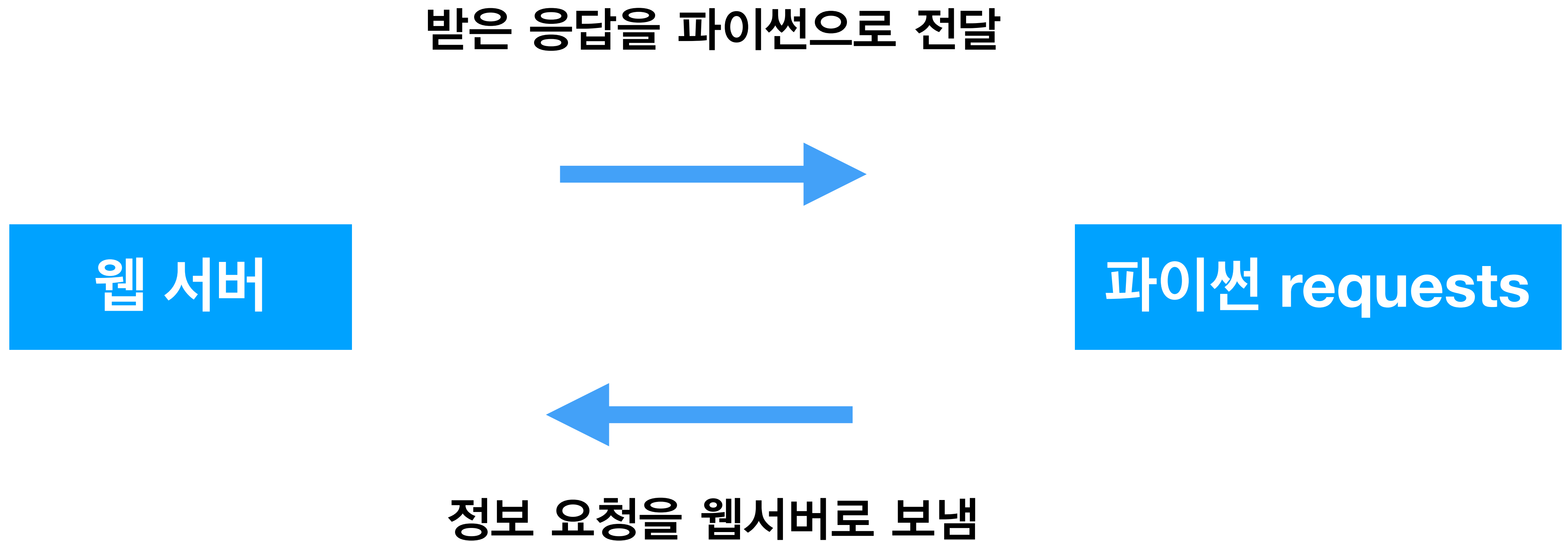
속성

## “속성값”

# CSS 스타일



### 3. Requests를 활용한 웹 요청





### 3. Requests를 활용한 웹 요청

# 기본적인 문법

#GET 방식

```
import requests  
response = requests.get('웹사이트 주소')
```

```
response_html = response.text
```

#POST 방식

```
Data = "{key값 : value값}"  
Response = requests.post('웹사이트 주소', data = data)  
response_value = Response.text  
# POST로 받는 것이 json이라면 Response.json()으로 받음.
```

```
Response_code = response.statue_code  
#statue_code는 응답이 알맞게 왔는지 확인하는 코드를 뽑아줍니다.
```

# 200은 알맞게 온 것이고, 404는 요청한 자원을 찾지 못했다는 의미입니다.

# <https://developer.mozilla.org/ko/docs/Web/HTTP/Status>

# 위 사이트에는 각종 응답코드에 대한 설명이 있고, 또한 사실 API를 사용하는 경우

# API마다의 응답 코드가 존재할 수 있습니다.





### 3. Requests를 활용한 웹 요청

```
import requests
# requests 패키지를 불러옵니다.

response = requests.get('https://www.naver.com')
# get을 사용해 naver홈페이지 정보를 요청합니다.
# 이때 웹서버의 응답은 response라는 변수에 저장됩니다!

response_code = response.status_code
# 웹서버로부터 도착한 response에 status_code를 사용해
# 요청에 대한 응답이 잘 왔는지 확인합니다.

if response_code == 200
    print(response.text)
else
    print('error code : ' + str(response_code))
```



### 3. Requests를 활용한 웹 요청

```
import requests

url = "https://openapi.naver.com/v1/datalab/search"
client_id = "X9S83UuJsvBu1zKJsdep"
client_secret = "Vph796WJWB"
# naver api를 사용하기 위해 아이디와 비밀번호를 입력합니다.

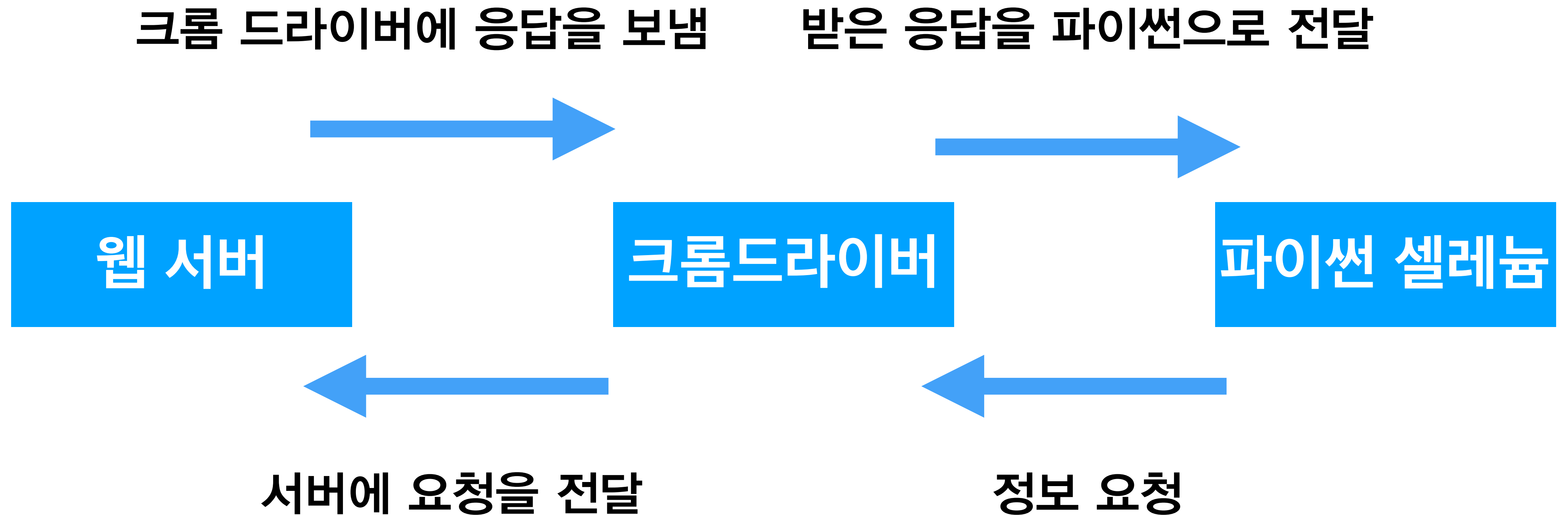
body = "{\n\"startDate\":\n\"2017-01-01\n\", \n\"endDate\":\n\"2017-04-30\n\", \n\"timeUnit\":\n\"month\n\", \n\"keywordGroups\": [\n{\n\"groupName\":\n\"배그\n\", \n\"keywords\": [\n\"배틀그라운드\n\", \n\"pubg\n\", \n\"배그\n\", \n\"battleground\n\", \n\"카배\n\"]\n}], \n\"device\":\n\"pc\n\", \n\"ages\": [\n\"1\n\", \n\"2\n\", \n\"3\n\", \n\"4\n\"], \n\"gender\":\n\"f\n\"}"
header = {"X-Naver-Client-Id" : client_id,
"X-Naver-Client-Secret" : client_secret, "Content-Type" : "application/json"}
# POST요청을 보내기 위해 택배상자 안에 넣을 정보를 따로 기입해 놓습니다.

r = requests.post(url, data = body.encode(encoding 'utf-8'), headers = header)
# post를 사용해 data라는 변수에 우리가 준비한 택배 내용물을 넣고 요청을 보냅니다.

rcod = r.status_code
if(rcod ==200)
    dc = r.json()
    for i in dc['results'] 0 ['data']:
        print(i['period'], ' : ', i['ratio'])
else
    print("Error Code:" + str(rcod))
# 위의 네이버 API는 키워드에 대한 검색 비율을 json으로 응답해주는 API입니다. 따라서 응답을 .json()으로 해석하여 뽑아냅니다.
```



#### 4. Selenium을 활용한 웹 브라우저 접근





## 4. Selenium을 활용한 웹 브라우저 접근

`driver.get(URL)`

URL에 get방식으로 접근하여 웹페이지를 여는 코드입니다.  
url에는 문자열('www.naver.com')이 들어갑니다.

브라우저에서 웹페이지를 열었다면 각종 버튼과 입력칸에 접근 할 수 있습니다.

`driver.find_element_by_id()`

엘리먼트의 속성값인 id를 사용하여 접근합니다.

`driver.find_element_by_class_name()`

엘리먼트의 속성값인 class를 사용하여 접근합니다.

`driver.find_element_by_css_selector()`

엘리먼트의 css\_selector를 사용하여 접근합니다.

`driver.find_element_by_xpath()`

엘리먼트의 xpath를 사용하여 접근합니다.

`element.text`

접근한 엘리먼트의 텍스트에 접근합니다.



## 4. Selenium을 활용한 웹 브라우저 접근

셀렉터 표현식	예시	설명
<u>.class</u>	.intro	클래스가 intro인 엘리먼트
<u>#id</u>	#firstname	id가 firstname인 엘리먼트
<u>*</u>	*	모든 엘리먼트
<u>element</u>	p	모든 p 태그
<u>element,element</u>	div, p	모든 div, p 태그
<u>element element</u>	div p	div 태그 안의 p 태그를 모두
<u>element&gt;element</u>	div > p	div 태그 바로 아래 계층의 p 태그를 모두
<u>element+element</u>	div + p	div 태그 바로 다음의 p 태그
<u>element1~element2</u>	p ~ ul	p 태그 뒤에 오는 ul 태그 모두





## 4. Selenium을 활용한 웹 브라우저 접근

셀렉터 표현식	예시	설명
<u>[attribute]</u>	[target]	target이라는 속성이 있는 엘리먼트 모두
<u>[attribute=value]</u>	[target=_blank]	target이 _blank인 엘리먼트 모두
<u>[attribute~=value]</u>	[title~=flower]	title이라는 속성에 flower가 포함되어 있는 엘리먼트 모두
<u>[attribute =value]</u>	[lang =en]	lang이라는 속성이 en으로 시작하는 엘리먼트 모두
<u>[attribute^=value]</u>	a[href^="https"]	a 태그를 고르는데 href속성이 https로 시작하는 것들
<u>[attribute\$=value]</u>	a[href\$=".pdf"]	a 태그를 고르는데 href속성이 .pdf로 끝나는 것들
<u>[attribute*=value]</u>	a[href*="w3schools"]	a 태그를 고르는데 href속성에 w3schools라는 글자가 포함



## 4. Selenium을 활용한 웹 브라우저 접근

```
from selenium import webdriver
```

```
chromedriver_address = '/Users/joono'
```

```
# 수강생님의 컴퓨터에 있는 크롬드라이버 설치 폴더 경로를 입력합니다.
```

```
# 윈도우는 폴더에서 복사해서 가져오는 경우 \를 /로 바꿔주셔야 합니다.
```

```
driver = webdriver.Chrome(chromedriver_address + '/chromedriver')
```

```
# 파이썬에서 우리가 설치한 웹드라이버를 인식할 수 있도록 경로를 알려줍니다.
```

```
web_url = 'http://news.naver.com/'
```

```
# 정보를 요청할 웹페이지의 주소를 미리 web_url에 저장합니다.
```

```
driver.get(web_url)
```

```
# get방식을 사용하여 정보를 요청합니다. 이때 크롬 브라우저가 열리면서 네이버 홈페이지에 접근합니다.
```

```
driver_source = driver.page_source
```

```
# 페이지의 소스를 가져옵니다. 이때 HTML문서를 가져오게 됩니다.
```

```
print(driver_source)
```



## 4. Selenium을 활용한 웹 브라우저 접근

```
from selenium import webdriver
```

```
chromedriver_address = '/Users/joono'
```

```
# 수강생님의 컴퓨터에 있는 크롬드라이버 설치 폴더 경로를 입력합니다.
```

```
# 윈도우는 폴더에서 복사해서 가져오는 경우 \를 /로 바꿔주셔야 합니다.
```

```
driver = webdriver.Chrome(chromedriver_address + '/chromedriver')
```

```
keyword = '패스트캠퍼스'
```

```
# 이번에는 키워드 검색을 할 수 있는 url이기 때문에 키워드 parameter에 들어갈 키워드를 미리 변수로 지정합니다.
```

```
web_url = 'https://search.naver.com/search.naver?query='+keyword '&where=news&ie=utf8&sm=nws_hyt'
```

```
# 네이버 뉴스 검색에 키워드를 넣은 주소를 web_url에 저장합니다.
```

```
driver.get(web_url)
```

```
# 키워드를 넣은 주소에 접근합니다.
```

```
driver_source = driver.page_source
```

```
# 검색 결과의 HTML문서를 가져옵니다.
```

```
print(driver_source)
```

```
# HTML 문서를 확인합니다.
```



## 4. Selenium을 활용한 웹 브라우저 접근

```
from selenium import webdriver
import time

chromedriver_address = '/Users/joono'
# 수강생님의 컴퓨터에 있는 크롬드라이버 설치 폴더 경로를 입력합니다.
# 윈도우는 폴더에서 복사해서 가져오는 경우 \를 /로 바꿔주셔야 합니다.

driver = webdriver.Chrome(chromedriver_address + '/chromedriver')

web_url = 'https://www.facebook.com/'
# 이번에는 페이스북 로그인페이지를 들어가서 아이디 입력칸의 위치를 가져와보는 실습입니다.
driver.get(web_url)
# 페이스북 로그인 페이지의 아이디 입력칸의 HTML코드입니다.
# <input type="email" class="inputtext" name="email" id="email"
# tabindex="1" data-testid="royal_email">
# 아이디 입력칸 엘리먼트는 input 태그이고, id는 email, class는 inputtext인것을 알 수 있습니다.
byid = driver.find_element_by_id('email')
byid.send_keys('id')
time.sleep(5)

bycn = driver.find_element_by_class_name('inputtext')
bycn.send_keys('cn')
time.sleep(5)

bycs = driver.find_element_by_css_selector('#email')
bycs.send_keys('cs')
time.sleep(5)
# 여러가지 접근 방법을 통해 같은 엘리먼트를 다른 코드로 접근합니다.
# .send_keys는 키보드 입력을 코드를 통해 하는 것인데, 3주차에 좀더 심화학습할 예정입니다!
driver_source = driver.page_source
# 마찬가지로 페이지 소스를 가져와 봅니다.
print(driver_source)
```



Q & A