

Baf

## 웹 데이터 크롤링 -Web Data Crawling-

- 웹 크롤링 이해
- R을 이용한 네이버 영화 댓글 크롤링 실습
- Selenium

비어플 3기 | 김영석

---

# 웹 크롤링 이해

---

01

02

03

## 텍스트 크롤링 (Crawling)

- 텍스트 분석의 대상으로부터 분석에 사용할 수 있는 형태로 텍스트를 가져오는 것
- 크롤링의 대상: PDF파일, HWP파일, 웹 사이트, Social Media, 신문기사, 블로그 등
- 다양한 크롤링 도구(Tools)가 존재
- 무료 통계 패키지 R에도 웹 크롤링을 위한 패키지가 존재 (현재로는 완벽하지는 않음)
- 크롤링에 관해 다음과 같이 다양한 issue가 존재함

01

02

03

## 웹 페이지의 구성

### 콘텐츠 형식 중 text 관련

text/html - 웹 페이지상에서 문단, 제목, 표, 이미지, 동영상 등을 정의하고 그 구조와 의미를 부여하는 마크업 언어 like 사람

text/css - 배경색, 폰트, 콘텐츠의 레이아웃 등을 지정하여, HTML 콘텐츠를 꾸며주는 스타일 규칙 언어 like 패션

Text/plain - JavaScript - 동적으로 콘텐츠를 바꾸고, 멀티미디어를 다루고, 움직이는 이미지 등 웹 페이지를 꾸며주도록 하는 프로그래밍 언어 like 근육

01

## 웹 페이지의 구성

02

### 1. Html로 뼈대 만들기

03

```
1 | <p>Player 1: Chris</p>
```

Player 1: Chris

### 2. Css를 통해 꾸며주기

```
1 | p {
2 |   font-family: 'helvetica neue', helvetica, sans-serif;
3 |   letter-spacing: 1px;
4 |   text-transform: uppercase;
5 |   text-align: center;
6 |   border: 2px solid rgba(0,0,200,0.6);
7 |   background: rgba(0,0,200,0.3);
8 |   color: rgba(0,0,200,0.6);
9 |   box-shadow: 1px 1px 2px rgba(0,0,200,0.4);
10 |   border-radius: 10px;
11 |   padding: 3px 10px;
12 |   display: inline-block;
13 |   cursor:pointer;
14 | }
```

PLAYER 1: CHRIS

### 3. JavaScript를 이용하면 클릭을 통해 이동하는 효과 등 동적으로 구현이 가능함

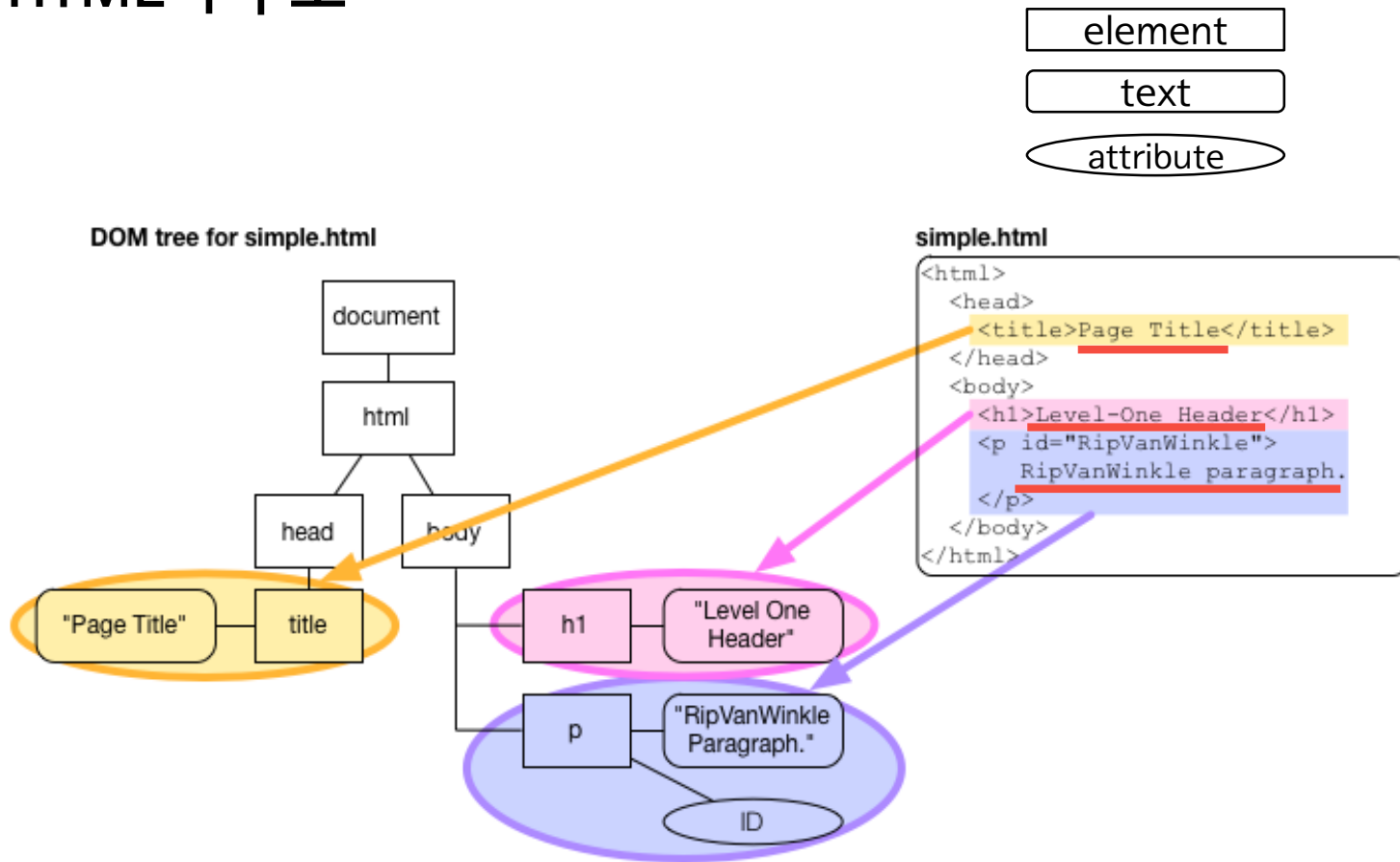
크롤링의 목적 : html 뼈대 사이의 text를 얻고자 함

01

# HTML의 구조

02

03



출처: <<http://dartdoc.takyam.com/docs/tutorials/connect-dart-html/>>

01

## HTML의 구성요소 및 예시

02

**element** : HTML에서 시작 태그와 종료태그로 이루어진 모든 명령어들을 의미

03

**tag** : element의 일부로 시작태그와 종료태그 두 종류가 있다. ex) `<script>`, `<dl>`, `</script>`, `</dl>`

**attribute** : 요소의 시작 tag 안에서 사용되는 것으로 좀 더 구체화된 명령어 체계를 의미 ex) `class`

**arguments** : attribute와 관련된 값 ex) `"boardViewSkin9_title"`

```
▷ <script>//<!-- var arrResizeImage = ...</script>
```

```
▲ <dl class="boardViewSkin9_title">
```

```
    <dt>기상업무 종사자 등의 교육훈련사업 위탁기관 지정 공고</dt>
```

```
    <dd>2018/10/01</dd>
```

```
</dl>
```

01

## CSS 선택자

원하는 정보를 뽑아내기 위해선 CSS 선택자를 알아야 함

1. 타입 선택자 – 특정 element를 선택 ex) `title`, `div`, `article`
2. 클래스(class) 선택자 – 특정 값을 class 속성(attribute)의 값으로 갖는 element를 선택 ex) `li.sitemap` : “.” 이 class를 의미
3. 아이디(id) 선택자 – 특정 값을 id 속성(attribute)의 값으로 갖는 element를 선택 ex) `div#title` : “#” 이 id를 의미
4. 속성 선택자 – 특정 속성을 갖고 있거나 특정 속성이 특정 값을 갖고 있는 element를 선택 ex) `label[for=sitelink5]`

```
<label class="blind" for="sitelink5">기상관련단체 바로가기</label>
```



01

02

03

## rvest 패키지의 용어

**node** - html에서 tag라고 불리는 것

**attr** - html의 attribute

**text** - 시작 태그와 종료 태그 사이에 있는 글자

ex) `<dl class="arg"> 안녕하세요 </dl>`

## rvest의 동작 순서

1. html 문서 데이터 가져오기
2. 필요한 노드 선택하기
3. 노드 내의 text 가져오기(attribute 가져오기)

ex) `read_html(url) %>% html_nodes("dl.arg") %>% html_text`

---


# R을 이용한 네이버 영화 댓글 크롤링 실습

---

01

02

03

네이버 영화 

다른 사이트를 보시려면 클릭하세요. [다른 사이트 더보기](#)



**아이언맨 3** (Iron Man 3, 2013)

네티즌 ★★★★★ 8.86 (15,447) | 기자평론가 ★★★★★ 7.53 (9) 평점주기▶

SF, 액션, 모험 | 2013.04.25. 개봉 | 129분 | 미국 외 | 12세 관람가

감독 **세인 블랙**

관객수 9,001,679명

내용 <어벤져스> 뉴욕 사건의 트라우마로 인해 영웅으로서의 삶에 회... **더보기**

관련정보 [명대사 보기](#)

↓ 다운로드

♡ 3,666

출연

**관람객 평점**

포토/동영상

시리즈 작품

AiTEMS 추천영화

편성표

★★★★★ 10 | chld\*\*\*\* | 👍 1,281

최고다. 후속작은 재미없을 거라는 편견을 깨준영화.

★★★★★ 10 | trex\*\*\*\* | 👍 753

자, 이제 어벤져스2가 기대된다.

★★★★★ 10 | hana\*\*\*\* | 👍 604

아이언맨 기대를 저버리지 않는군

14,452개 평점 전체보기

‘아이언맨3’ 검색 후

‘더보기’ 클릭

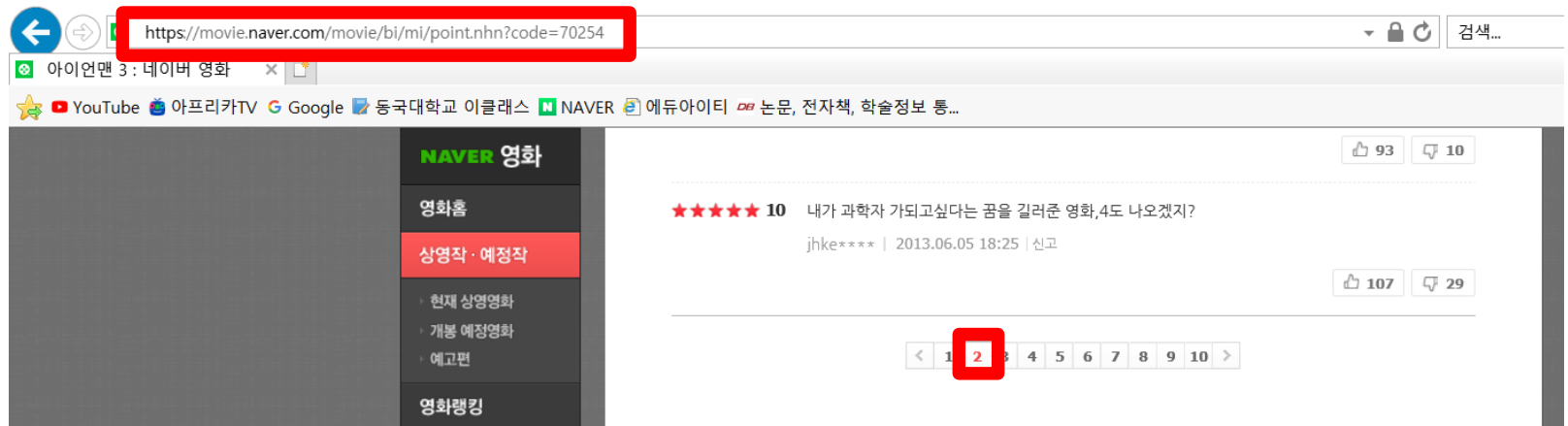
[정보오류 수정요청](#)

## '펼쳐' 클릭

01

02

03



URL 즉, 주소가 페이지에따라 바뀌지 않음

01

02

03

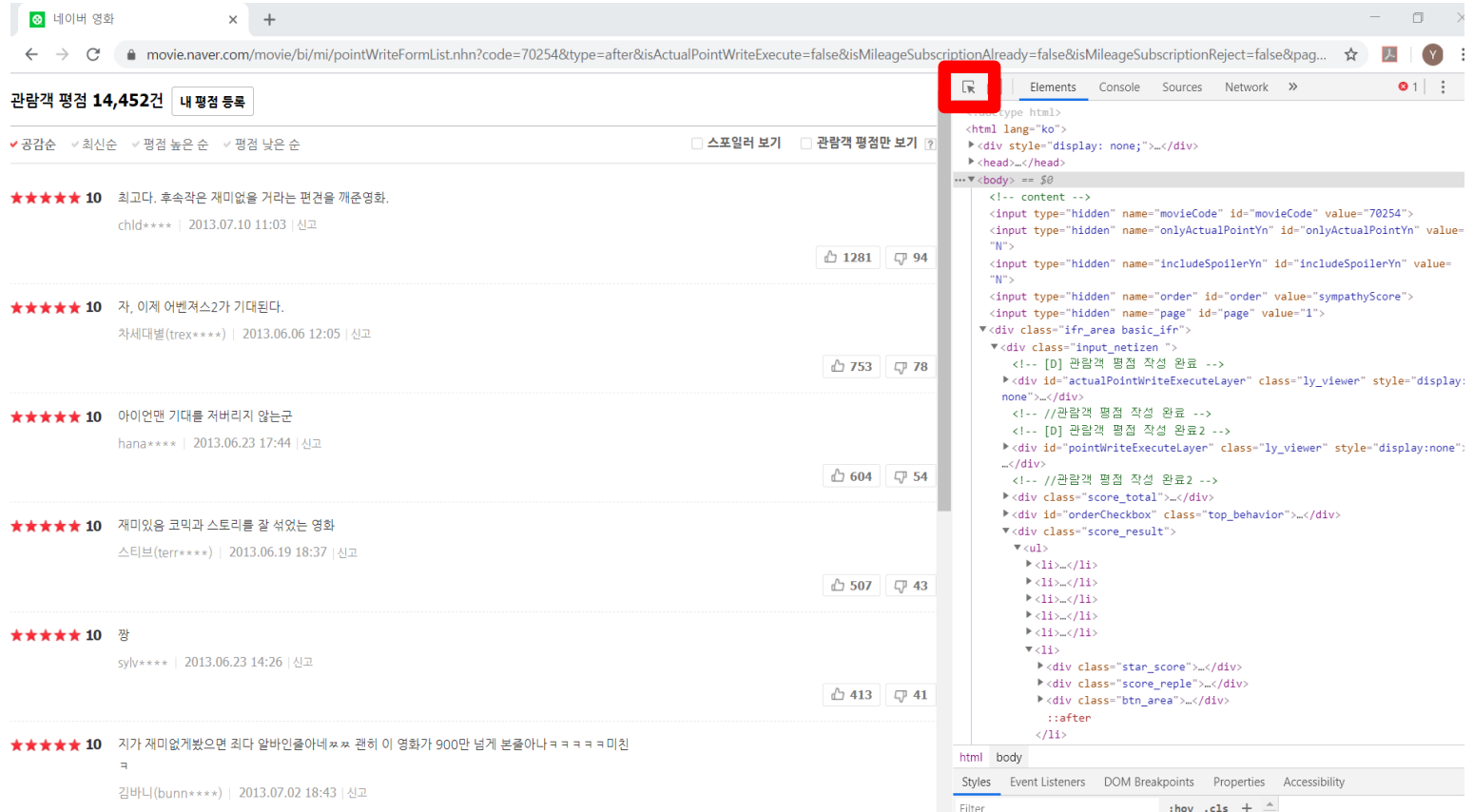


하단의 '번호' 우클릭 후  
'새 탭에서 열기' 클릭

01

02

03



‘F12’ 누른 후

⑤  ’를 클릭하여 ‘요소’ 확인

01

02

03

The screenshot shows a web browser window displaying a movie review page on Naver. The URL is `movie.naver.com/movie/bi/mi/pointWriteFormList.nhn?code=70254&type=after&isActualPointWriteExecute=false&isMileageSubscriptionAlready=false&isMileageSubscriptionReject=false&pag...`. The page shows a list of reviews for a movie. The first review is highlighted with a yellow box. A yellow arrow points from this review to the DOM tree on the right. The DOM tree shows the HTML structure, with the highlighted review text corresponding to a `span` element with `id="filtered_ment_0"`.

⑤ 첫 번째 댓글에 '🖱'를 올려 '요소' 확인

이 때, 두번째 댓글도 확인하여 어떤것이 바뀌는지 확인



01

02

03

네이버 영화

movie.naver.com/movie/bi/mi/pointWriteFormList.nhn?code=70254&type=after&isActualPointWriteExecute=false&isMileageSubscriptionAlready=false&isMileageSubscriptionReject=false&pag...

관람객 평점 14,440건 내 평점 등록

공감순 최신순 평점 높은 순 평점 낮은 순 스포일러 보기 관람객 평점만 보기

★★★★★ 10 최고다. 후속작은 재미없을 거라는 편견을 깨준 영화.  
span#\_filtered\_ment\_1 173.33 × 15.2  
Color #333333  
Font 13px 나눔고딕, NanumGothic, 돋움, Dotu...

★★★★★ 10 장...이제 이번저스2가 기대된다  
차세대별(trex\*\*\*\*) | 2013.06.0... | 신고

★★★★★ 10 아이언맨 기대를 저버리지 않는군  
hana\*\*\*\* | 2013.06.23 17:44 | 신고

★★★★★ 10 재미있음 코믹과 스토리를 잘 섞었다는 영화  
스티브(terr\*\*\*\*) | 2013.06.19 18:37 | 신고

★★★★★ 10 짱  
sylv\*\*\*\* | 2013.06.23 14:26 | 신고

★★★★★ 10 지가 재미있게봤으면 최다 알바인줄아네ㅋㅋ 관해 이 영화가 900만 넘게 본줄아나ㅋㅋㅋㅋ미친  
김바니(bunn\*\*\*\*) | 2013.07.02 18:43 | 신고

Elements

```

none > </div>
<!-- //관람객 평점 작성 완료 -->
<!-- [D] 관람객 평점 작성 완료2 -->
<div id="pointWriteExecuteLayer" class="ly_viewer" style="display:none">
</div>
<!-- //관람객 평점 작성 완료2 -->
<div class="score_total"></div>
<div id="orderCheckbox" class="top_behavior"></div>
<div class="score_result">
<ul>
<li></li>
<li>
<div class="star_score"></div>
<div class="score_reple">
<p>
<span id="filtered_ment_1">...</span>
</p>
<dl></dl>
</div>
<div class="btn_area"></div>
::after
</li>
<li></li>
<li></li>
<li></li>
<li></li>
</ul>

```

html body div div div.score\_result ul li div.score\_reple p span#\_filtered\_ment\_1

Styles Event Listeners DOM Breakpoints Properties Accessibility

Filter :hov .cls

Console What's New

Highlights from the Chrome 80 update

Support for let and class redeclarations  
When experimenting with new code in the Console, repeating let or class declarations no longer causes errors.

Improved WebAssembly debugging  
The Sources panel has increased support for stepping over code, setting breakpoints, and resolving stack traces in source languages.

가장 뒤의 숫자가 1로 바뀜

01

02

03

The screenshot shows a web browser with the URL `movie.naver.com/movie/bi/mi/pointWriteFormList.nhn?code=70254&type=after&isActualPointWriteExecute=false&isMileageSubscriptionAlready=false&isMileageSubscriptionReject=false&pag...`. The page displays a list of movie reviews. The first review has a 10-star rating. A yellow arrow points from this rating to the browser's developer tools, which shows the HTML structure. The HTML element for the 10-star rating is `<em>10</em>`.

⑤ 첫 번째 평점에 '10'를 올려 '요소' 확인

01

02

03

The screenshot shows a web browser displaying a list of movie reviews on Naver. The browser's developer tools are open, showing the DOM tree. A yellow box highlights the number '10' in a review, and a red box highlights the corresponding HTML path in the DOM tree: `html > body > div > div.input_netizen > div.score_result > ul > li > div.star_score > em`.

⑥하단의 요소를 참고하면

쉽게 해당 요소를 찾을 수 있습니다.

01

02

03

- `library(rvest)`
  - 크롤링을 위한 패키지

01

02

03

# 크롤링에 필요한 함수들

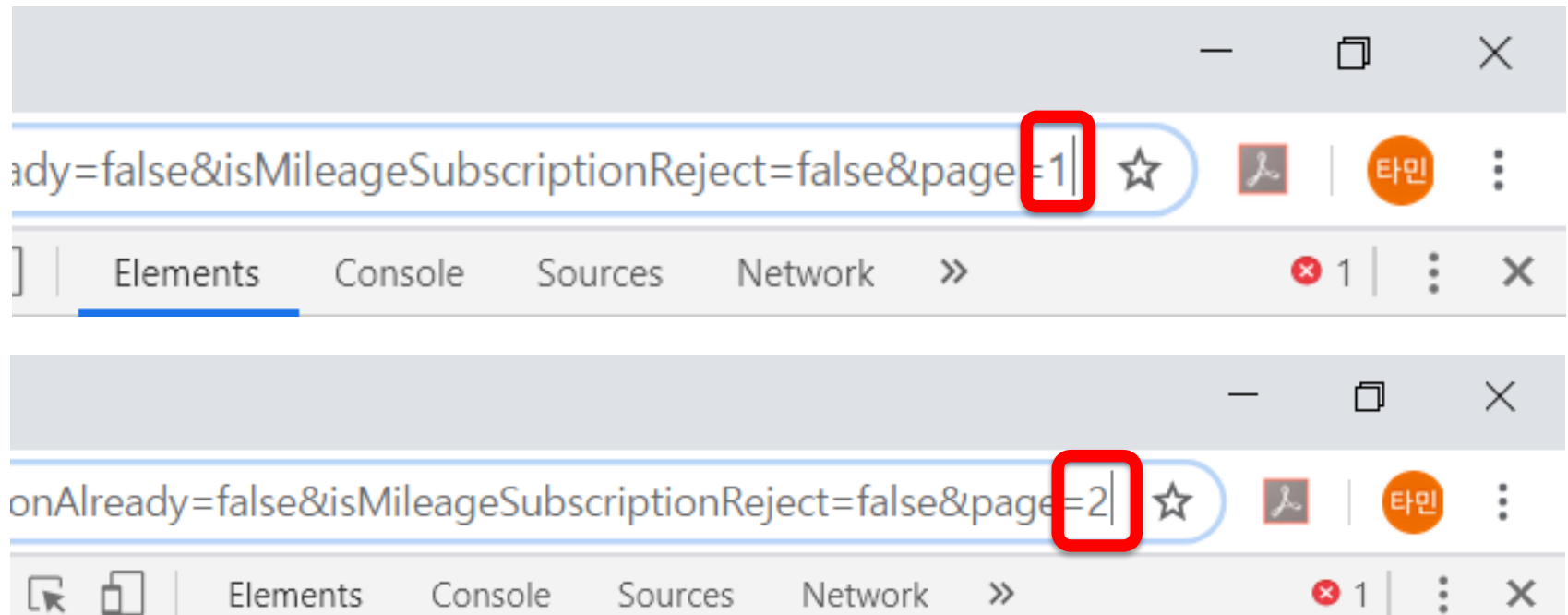
- R에서 해당 URL의 html 소스코드를 가져오는 함수 `read_html()`
- 특정 태그가 포함하고 있는 소스코드 및 속성을 추출할 때 사용하는 함수 `html_nodes()`
- 해당 html에서 텍스트만 추출할 때 사용하는 함수 `html_text()`

01

02

03

# 1페이지와 2페이지의 주소 변화



01

02

03

```
url_base<-
```

```
https://movie.naver.com/movie/bi/mi/pointWriteFormList.  
nhn?code=70254&type=after&isActualPointWriteExecute  
=false&isMileageSubscriptionAlready=false&isMileageSub  
scriptionReject=false&page=
```

주소는 여러분 하고 싶은 영화 아무거나

**paste 함수와 for문을 이용해  
페이지를 바꿀겁니다.**

01

02

03

paste(url\_base, 1, sep="")

fileageSubscriptionReject=false&page=1'

붙었죠?



## 한 페이지 읽어오기

#주소설정

```
url<-paste(url_base,1,sep="")
```

#html 읽어오기

```
htxt<-read_html(url,encoding="UTF-8")
```

#node 읽기

```
table<-html_nodes(htxt,".score_result")
```

```
content<-html_nodes(table,".score_reple")
```

```
content2<-
```

```
html_nodes(content,paste("#_filtered_ment_",1,sep=""))
```

#text읽기

```
reviews<-html_text(content2) ; reviews
```

01

02

03

## for문을 활용한 댓글 크롤링

```
all_reviews<-c()
for(page in 1:10){
  for(num in 1:9){
    url<-paste(url_base,page,sep='')
    htxt<-read_html(url,encoding="UTF-8")
    table<-html_nodes(htxt,".score_result")
    content<-html_nodes(table,".score_reple")
    content2<-
html_nodes(content,paste("#_filtered_ment_",num,sep=''))
    reviews<-html_text(content2)
    if(length(reviews)==0){break}
    all_reviews<-c(all_reviews,reviews)
    print(page)
  }
}
```

각각 댓글 번호와 페이지를  
바꿔가며 크롤링

```
> head(all.reviews)
```

[illegible][illegible][illegible][illegible]

```
> data<-gsub("[[:cntrl:]]","",all.reviews)
```

```
> head(data)
```

[1] "자, 이제 어벤져스2가 기대된다. "

[2] "아이언맨 기대를 저버리지 않는군 "

[3] "재미있음 코믹과 스토리를 잘 섞었는 영화 "

[4] "짱"

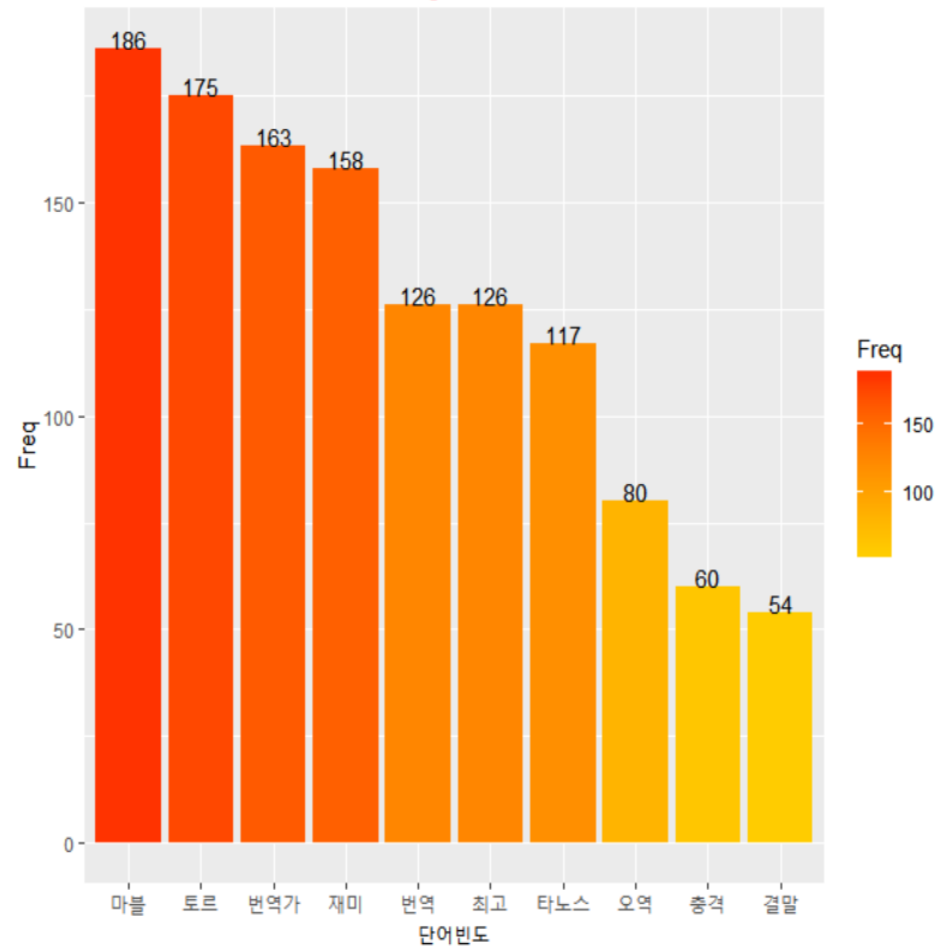
[5] "지가 재미없게봤으면 죄다 알바인줄아네 xx xx 괜히 이 영화가 900만 넘게 본줄아나 ㅋㅋㅋㅋ미친ㅋ"

[6] "1편, 2편은 3편을위해태어났다 "

## 결과 활용



## 어벤져스리뷰 단어빈도



01

02

03

**리뷰 크롤링 했던 코드를 가지고  
평점을 크롤링 해보세요.**

01

02

03

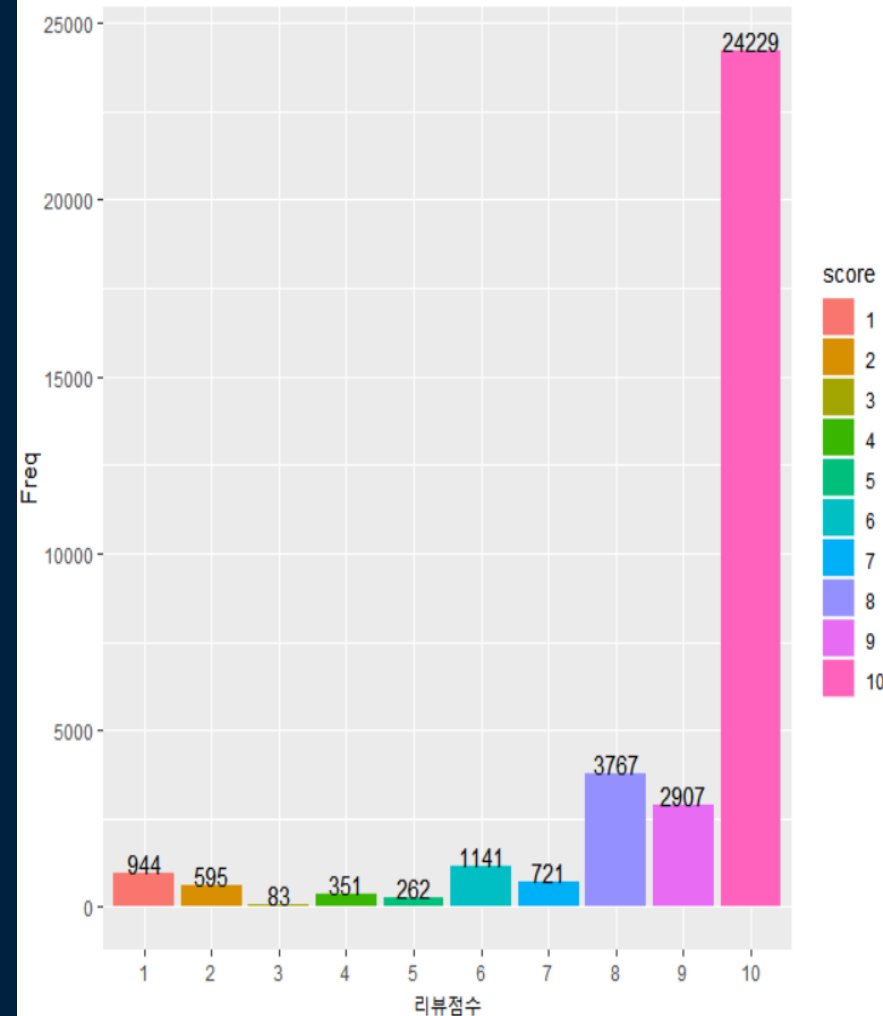
```

all_score<-c()
for(page in 1:10){
  url<-paste(url_base,page,sep="")
  htxt<-read_html(url,encoding="UTF-8")
  html<-html_nodes(htxt,".score_result")
  content<-html_nodes(html,".star_score")
  content2<-html_nodes(content,"em")
  score<-html_text(content2)
  all_score<-c(all_score,score)
  print(page)
}

as.numeric(all_score)
hist(as.numeric(all_score))

```

## 어벤져스 평점



01

02

03

## dplyr을 활용한 코드 간소화

```
all_score2<-list() #append 대신 list형식을 활용
for(page in 1:10){
  all_score2[[page]]<-read_html(paste(url_base,page,sep=""),encoding="UTF-
8") %>% html_nodes(".score_result") %>%
  html_nodes(".star_score") %>% html_nodes("em") %>% html_text()
}
```

```
all_score2
unlist(all_score2) #리스트 형식이기 때문에 unlist
```



# Selenium





01

02

03

# Selenium 이란?

여러 언어에서 웹드라이버를 통해 웹 자동화 테스트 혹은 웹 자동화를 도와주는 라이브러리.

즉, 여러 플랫폼의 브라우저 자동화를 지원하는 자동화도구!

# 예시)

- 1 세분류명 검색
- 2 기간입력
- 3 게시물 수 수집

NAVER 블로그

글

1

홈쇼핑 멀치

Q

통합검색

블로그 홈

주제별 보기

이달의 블로그

공식블로그

파워블로그

챌린지 프로그램

글

블로그

별명·아이디

홈쇼핑 멀치에 대한 검색결과입니다. 57건

3

✓ 정확도

✓ 최신순

기간 입력 ^

기간 전체

최근 1주

최근 1개월

기간 입력

2017-05-01

2017-05-31

적용

강원도 감자옹심이 감자전 공영홈쇼핑 아임쇼핑 2017. 5. 22.

감자옹심이 300g 10팩+감자전 240g 3팩+감자옹심이 소스 30g 10팩으로 구성 홈쇼핑 판매가 : 40,900원(자동 주문 시 1,000원 할인 39,900원) 모바일 구매... = 감자옹심이 = [재료] 감자옹심이 감자옹심이 소스 당근 애호박 양파 마늘 간장 다시 국물(다시 멀치, 북어 대가리, 다시마, 물) 당근, 양파...

풀향기

풀향기 | 풀향기의 맛있는 이야기

강원도 감자 진짜 옹심이와 강원도 감자 전통 감자전 공영홈쇼핑 판매~ 2017. 5. 21.

공영홈쇼핑에서 방송예정인... 강원도 감자 진짜 옹심이와 전통 감자전을... 받아보게 되었어요... 아이스 박스에 아이스팩을 넣어 냉동상태로... 만들어둔 멀치육수가 있어서... 끓고 있는 육수에 냉동상태의 옹심을 넣고... 애호박, 양파, 당근, 대파, 다진마늘을 넣어주고... 간은 소금으로...

아과마린

아과마린의 쉬운 일상요리~

34

# 사용되는 함수- library(RSelenium)

remoteDriver(port, browserName) : 처음 킬 때 포트 지정 및 어떤 플랫폼 사용할지 지정. 이 함수를 지정한 객체로 뒤에 함수를 구성하게 됩니다.

예) `remDr <- remoteDriver(port=4445L, browserName="chrome")`

`remDr$open()` : 창 열기

`remDr$navigate(주소)` : 입력 주소로 이동

`$findElement(using, value)` : 요소를 찾는 함수

(using : "xpath", "css selector" 등, value : 앞에서 했던 node 값)

`$clickElement()` : 마우스로 클릭

`$sendKeysToElement( list("검색어") 혹은 list(key="enter") 등등)` : 키보드 입력

그 외 스크롤 내리기 등 다양하게 있습니다.

01

02

03

## Css selector와 Xpath의 차이점

Css selector : 대량으로 최적화되어 있으며 브라우저에 내장되어 있음.  
속도가 매우 빠름.

Xpath : 모든 브라우저에 내장되어 있는 것은 아니며, 특히 IE에서는  
Xpath를 이용하려면 먼저 JavaScript-Xpath와 같은 도구를 사  
용 해야한다

하지만 Xpath에는 두드러지는 장점이 있다.

1. `//table/tr/td[contains., "foo")/../td[2]/input.` 과 같은 텍스트 내용을 기반으로 한 요소를 찾기 쉽다.
2. `./../div[2].` 과 같은 구조에서 관련된 상위 요소를 찾거나 반복문을 사용하기 좋다.

01

02

03

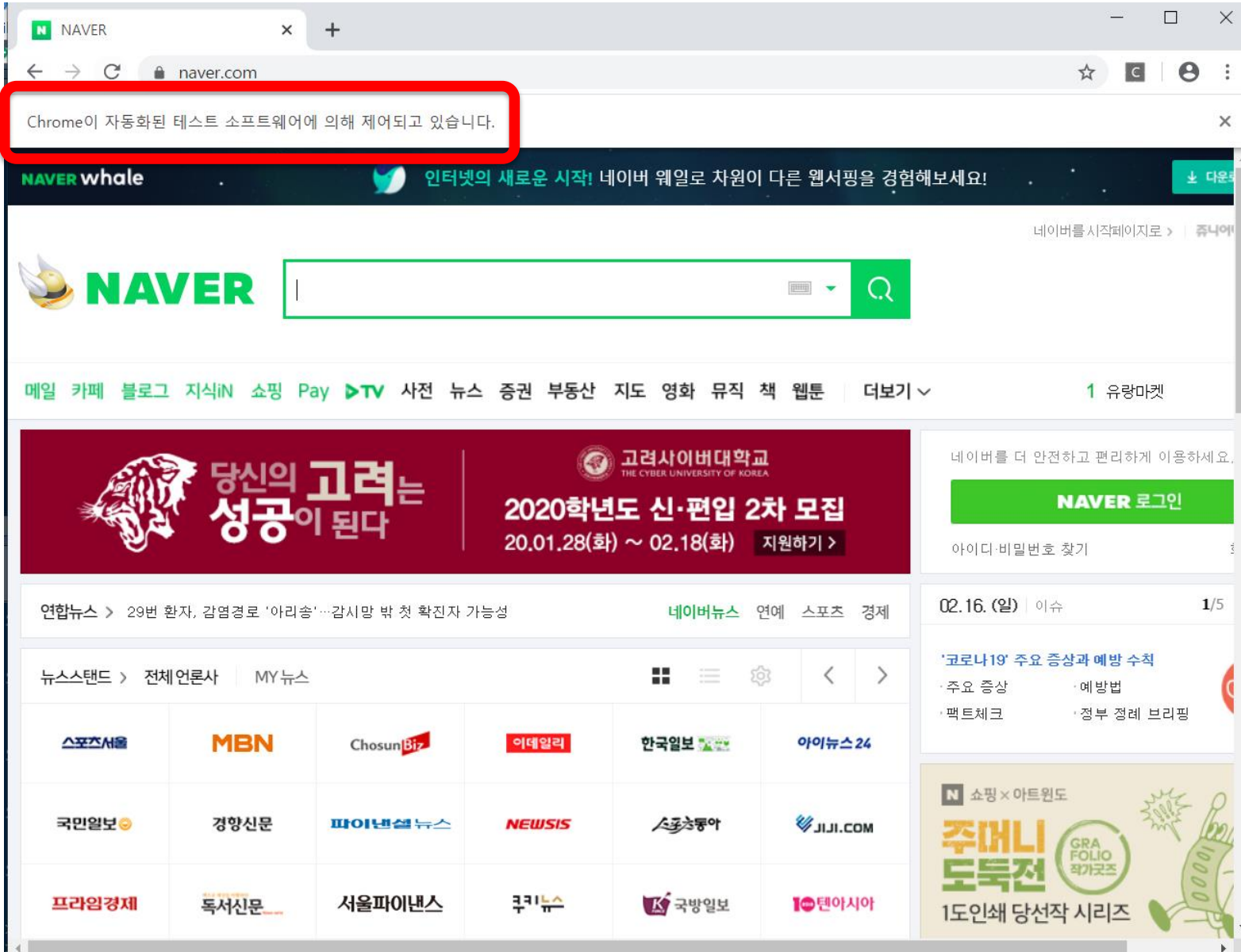
## 예시

```
remDr<-  
remoteDriver(port=4445L,browserName="chrome")  
remDr$open()  
remDr$navigate("http://www.naver.com")
```

01

02

03



01

02

03

## 예시

```
blogButton<-  
remDr$findElement(using="xpath",  
value='노드에서 우클릭 -> Copy  
-> Copy XPath')
```

이렇게 하면 블로그 버튼에 마우스가 올라갔다고  
생각하시면 됩니다.

01

02

03

# 예시

## blogButton\$clickElement()

블로그 버튼이 눌러질겁니다.



# 예시

이제 검색을 해야겠죠? 먼저 검색창에 접근할게요.  
이번엔 **css selector**로 해볼게요

```
webElemButton<-  
remDr$findElement(using="css  
selector",value= ' 검색창 노드를 찾  
아서 우클릭 -> Copy  
-> Copy selector')
```

01

02

03

## 예시

반복문을 쓸거니까 앞에 뭔가 쓰여져 있는 것을 지우겠습니다.  
Shift + home + delete 하면 모든 글자가 지워져요.

sendKeysToElement의 경우 list 형식의 값만 받습니다

```
webElemButton.sendKeysToElement(list(key='shift',key='home',  
key='delete'))
```

# 예시

이제 검색창에 검색어를 입력해야겠죠?

```
webElementButton.sendKeysToElement(list('자기이름'))
```

지금까지 코드를 잘 생각해보면 접근한 노드를 지정한 객체를 앞에 써주고 \$ 하고 어떤 행동을 할건지 적어주면 됩니다.

01

02

03

## 정리해보면

1. 내가 필요한 요소로 접근한다.
2. 내가 하고 싶은 행동을 취한다.

이 두 가지만 반복적으로 코딩해주면 됩니다.

내가 직접 할 일을 코딩한다고 생각하고 하세요.

쉽죠?

01

02

03

## 이제 직접 해봅시다.

1. 검색한다.
2. 날짜 지정란을 누른다.
3. 날짜를 지정한다.
4. 적용하기를 누른다.

01

02

03

## 예시

검색 버튼을 누릅니다. 검색버튼에 접근할게요.

```
click_button<-  
remDr$findElement(using="css  
selector",value= ' 앞에서 했던 것과 똑같  
이 검색버튼 노드 찾아서 Copy selector ')
```

01

02

03

# 예시

클릭하면 되겠죠?

`click_button$clickElement()`

01

02

03

## 예시

여기서 주의할 점이 있습니다!!! 창이 넘어가고 있는 도중에 코드가 돌아가면 오류가 나겠죠?

```
click_button$clickElement() ; Sys.sleep(2)
```

Sys.sleep()으로 R에서 코드 입력을 잠깐 쉬게 하는 겁니다.

또한, 같은 작업을 매우 많이 반복할 경우 트래픽 관련하여 IP를 막아버릴 수 있으니 적절히 Sys.sleep()을 넣어줍시다.



01

02

03

## 예시

날짜버튼 클릭

```
range_button<-  
remDr$findElement(using="css  
selector",value= ' 앞에서 했던 것과 똑같이 날  
짜버튼 노드 찾아서 Copy selector ')  
range_button$clickElement()
```

# 예시

그 다음은 모두 앞에서 한 것과 같습니다.

```
start_date_button<-remDr$findElement(using="css selector",value='#search_start_date')
start_date_button$sendKeysToElement(list(key='shift',key='home',key='delete'))
start_date_button$sendKeysToElement(list("20100101"))

end_date_button<-remDr$findElement(using="css selector",value='#search_end_date')
end_date_button$sendKeysToElement(list(key='shift',key='home',key='delete'))
end_date_button$sendKeysToElement(list("20120101"))

find_button<-remDr$findElement(using="css selector",value='#periodSearch')
find_button$clickElement()
find_button$clickElement();Sys.sleep(2)
```

# 예시

이제 검색이 되었으니 영화리뷰에서 했던 것처럼 크롤링 할건데 read\_html만 조금 달라요.

```
html<-read_html(remDr$getPageSource()[[1]])
```

처음 창 열때 지정한 객체 remDr를 이용해 html의 정보를 가져옴.

그 뒤는 영화 리뷰 한 것과 똑같습니다.



01

02

03

## 예시

```
html<-read_html(remDr$getPageSource()[[1]]);Sys.sleep(1)
content<-html_nodes(html, ".search_number")
num<-html_text(content)
num
```

01

02

03

**이제 반복문만 사용하면 완성입니다.**

Baf

웹 데이터 크롤링

Thank you

비어플 3기 | 김영석