R 언어의 구문 정리

Accessing the help files

?mean
Get help of a particular function.

help.search('weighted mean')
Search the help files for a word or phrase.

help(package = 'dplyr')
Find help for a package.

More about an object

str(iris)
Get a summary of an object's structure.

class(iris)
Find the class an object belongs to.

getwd()
Find the current working directory (where inputs are found and outputs are sent).

setwd('C://file/path')
Change the current working directory.

install.packages('dplyr')
Download and install a package from CRAN.

library(dplyr)
Load the package into the session, making all its functions available to use.

dplyr::select
Use a particular function from a package.

data(iris)
Load a built-in dataset into the environment.

Input	Ouput
`df <- read.table('file.txt')`	`write.table(df, 'file.txt')`
`df <- read.csv('file.csv')`	`write.csv(df, 'file.csv')`
`load('file.RData')`	`save(df, file = 'file.Rdata')`

R 언어의 구문 정리

Vectors

Creating Vectors

`c(2, 4, 6)`	`2 4 6`	Join elements into a vector
`2:6`	`2 3 4 5 6`	An integer sequence
`seq(2, 3, by=0.5)`	`2.0 2.5 3.0`	A complex sequence
`rep(1:2, times=3)`	`1 2 1 2 1 2`	Repeat a vector
`rep(1:2, each=3)`	`1 1 1 2 2 2`	Repeat elements of a vector

Vector Functions

`sort(x)`
Return x sorted.

`rev(x)`
Return x reversed.

`table(x)`
See counts of values.

`unique(x)`
See unique values.

`x[4]`	The fourth element.
`x[-4]`	All but the fourth.
`x[2:4]`	Elements two to four.
`x[-(2:4)]`	All elements except two to four.
`x[c(1, 5)]`	Elements one and five.

By Value

`x[x == 10]`	Elements which are equal to 10.
`x[x < 0]`	All elements less than zero.
`x[x %in% c(1, 2, 5)]`	Elements in the set 1, 2, 5.

Named Vectors

`x['apple']`	Element with name 'apple'.

R 언어의 구문 정리

Matrices

```
m <- matrix(x, nrow = 3, ncol = 3)
```
Create a matrix from x.

`m[2, ]` - Select a row

`m[ , 1]` - Select a column

`m[2, 3]` - Select an element

`t(m)`
Transpose

`m %*% n`
Matrix Multiplication

`solve(m, n)`
Find x in: m * x = n

Lists

```
l <- list(x = 1:5, y = c('a', 'b'))
```
A list is a collection of elements which can be of different types.

`l[[2]]`	`l[1]`	`l$x`	`l['y']`
Second element of l.	New list with only the first element.	Element named x.	New list with only element named y.

Also see the **dplyr** package.

Data Frames

```
df <- data.frame(x = 1:3, y = c('a', 'b', 'c'))
```
A special case of a list where all elements are the same length.

x	y
1	a
2	b
3	c

List subsetting

`df$x` 

`df[[2]]` 

Understanding a data frame

`View(df)` See the full data frame.

`head(df)` See the first 6 rows.

Matrix subsetting

`df[ , 2]` 

`df[2, ]` 

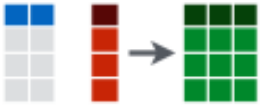
`df[2, 2]` 

`nrow(df)`
Number of rows.

`ncol(df)`
Number of columns.

`dim(df)`
Number of columns and rows.

cbind - Bind columns.



rbind - Bind rows.

R 언어의 구문 정리

Variable Assignment

```
> a <- 'apple'
> a
[1] 'apple'
```

The Environment

`ls()`	List all variables in the environment.
`rm(x)`	Remove x from the environment.
`rm(list = ls())`	Remove all variables from the environment.

`as.logical`	`TRUE, FALSE, TRUE`	Boolean values (TRUE or FALSE).
`as.numeric`	`1, 0, 1`	Integers or floating point numbers.
`as.character`	`'1', '0', '1'`	Character strings. Generally preferred to factors.
`as.factor`	`'1', '0', '1', levels: '1', '0'`	Character strings with preset levels. Needed for some statistical models.

Strings

```
paste(x, y, sep = ' ')
paste(x, collapse = ' ')
grep(pattern, x)
gsub(pattern, replace, x)
toupper(x)
tolower(x)
nchar(x)
```

Conditions

`a == b`	Are equal	`a > b`	Greater than	`a >= b`	Greater than or equal to	`is.na(a)`	Is missing
`a != b`	Not equal	`a < b`	Less than	`a <= b`	Less than or equal to	`is.null(a)`	Is null

R 언어의 구문 정리

For Loop

```
for (variable in sequence){
    Do something
}
```

Example

```
for (i in 1:4){
    j <- i + 10
    print(j)
}
```

While Loop

```
while (condition){
    Do something
}
```

Example

```
while (i < 5){
    print(i)
    i <- i + 1
}
```

If Statements

```
if (condition){
    Do something
} else {
    Do something different
}
```

Example

```
if (i > 3){
    print('Yes')
} else {
    print('No')
}
```

Functions

```
function_name <- function(var){
    Do something
    return(new_variable)
}
```

Example

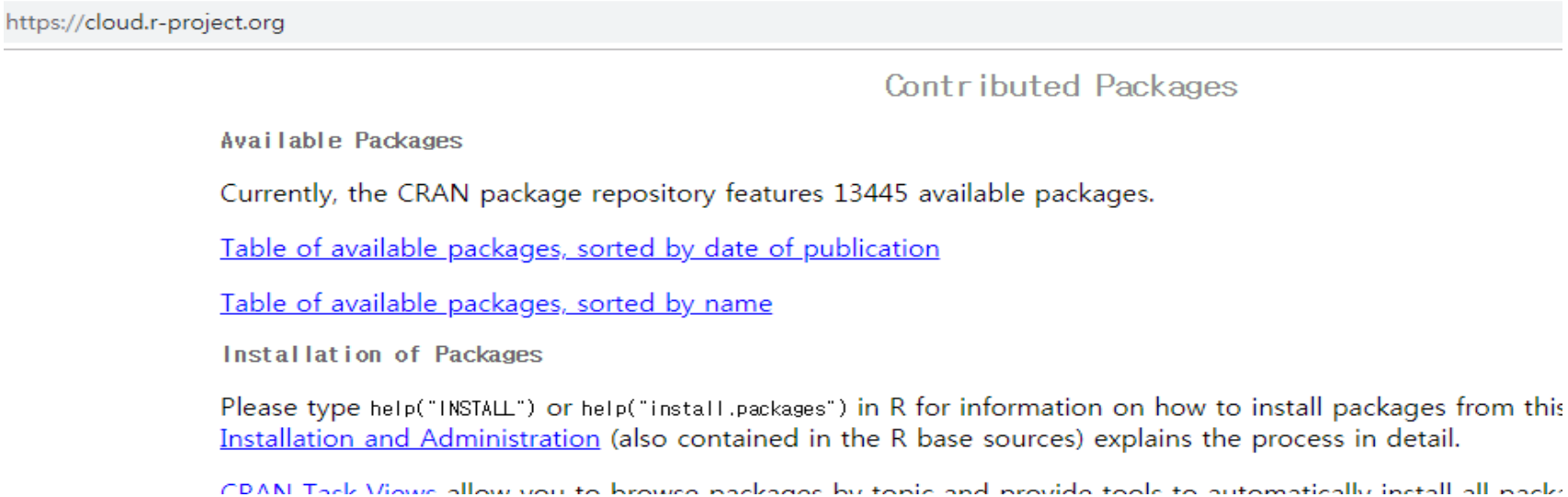
```
square <- function(x){

    squared <- x*x

    return(squared)
}
```

R 패키지

- R 패키지라는 것은 R를 가지고 할 수 있는 통계, 분석 그리고 시각화와 관련하여 기능을 정의한 함수들의 묶음이라 할 수 있다.
- R 패키지는 R을 설치할 때 함께 설치되는 기본 패키지가 있고 만약 찾는 기능이 없다면 원하는 기능을 처리해주는 패키지를 찾아서 추가로 설치 한 후 사용하면 된다.
- R 패키지는 CRAN(https://cran.r-project.org/) 사이트에서 모두 검색 가능하고 다운로드 받을 수 있다.
- R은 무료라는 장점 외에 일정 규칙에 맞춰 누구나 제작하고 배포할 수 있는 Package를 통해 기능 확장을 유연하게 할 수 있는 큰 장점을 갖고 있다.



R 패키지

- 새로운 R 패키지의 설치
  install.packages("패키지명")

- 이미 설치된 R 패키지 확인
  installed.packages()

- 설치된 패키지 삭제
  remove.packages("패키지명")

- 설치된 패키지의 버전 확인
  packageVersion("패키지명")

- 설치된 패키지 업데이트
  update.packages("패키지명")

- 설치된 패키지 로드
  library(패키지명)
  require(패키지명)

- 로드된 패키지 언로드
  detach("package:패키지명")

- 로드된 패키지 점검
  search()

- 설치된 패키지의 버전 점검
  packageVersion(패키지명)

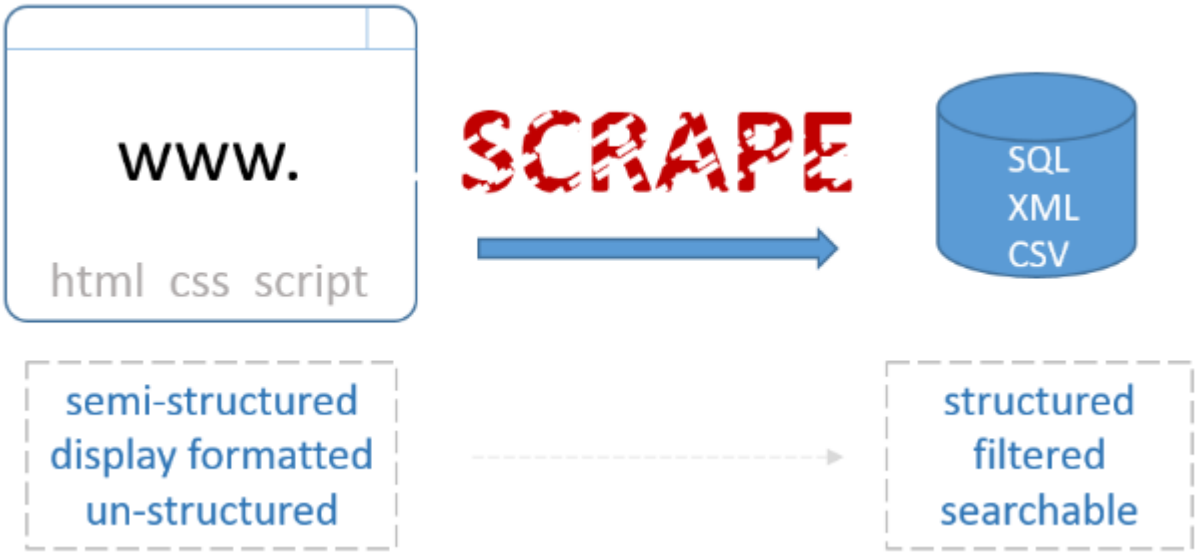
정적 스크래핑(크롤링)

[웹 스크래핑(web scraping)]

웹 사이트 상에서 원하는 부분에 위치한 정보를 컴퓨터로 하여금 자동으로 추출하여 수집하는 기술

[웹 크롤링(web crawling)]

자동화 봇(bot)인 웹 크롤러가 정해진 규칙에 따라 복수 개의 웹 페이지를 브라우징 하는 행위



정적 스크래핑(크롤링)

Selectors

Basics
#id
element
.class,
.class.class
*
selector1,
selector2

Hierarchy
ancestor
descendant
parent > child
prev + next
prev ~ siblings

Basic Filters
:first
:last
:not(selector)
:even
:odd
:eq(index)
:gt(index)
:lt(index)

Content Filters
:contains(text)
:empty
:has(selector)
:parent

Visibility Filters
:hidden
:visible

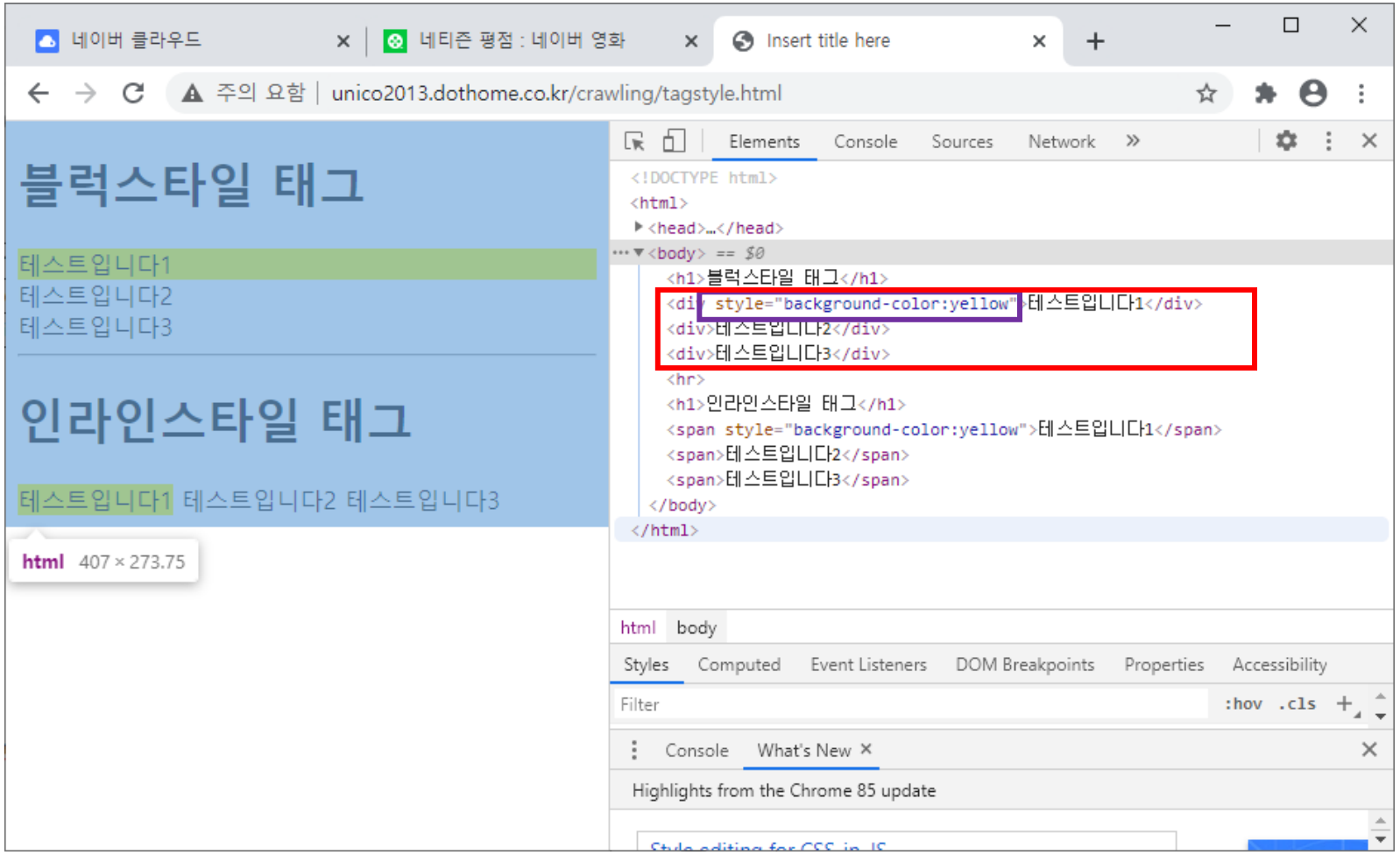
Child Filters
:nth-child(expr)
:first-child
:last-child
:only-child

Attribute Filters
[attribute]
[attribute=value]
[attribute!=value]
[attribute^=value]
[attribute$=value]
[attribute*=value]
[attribute|=value]
[attribute~=value]
[attribute]
[attribute2]

Forms
:input
:text
:password
:radio
:checkbox
:submit
:image
:reset
:button
:file

Form Filters
:enabled
:disabled
:checked
:selected

정적 스크래핑(크롤링)


네이버 클라우드
네티즌 평점 : 네이버 영화
Insert title here
주의 요함 | unico2013.dothome.co.kr/crawling/tagstyle.html
블럭스타일 태그
테스트입니다1
테스트입니다2
테스트입니다3
인라인스타일 태그
테스트입니다1 테스트입니다2 테스트입니다3
html 407 × 273.75
Elements
Console
Sources
Network
<!DOCTYPE html>
<html>
<head>…</head>
<body> == $0
<h1>블럭스타일 태그</h1>
테스트입니다1
테스트입니다2
테스트입니다3
<hr>
<h1>인라인스타일 태그</h1>
테스트입니다1
테스트입니다2
테스트입니다3
</body>
</html>
html
body
Styles
Computed
Event Listeners
DOM Breakpoints
Properties
Accessibility
Filter
:hov .cls
Console
What's New
Highlights from the Chrome 85 update

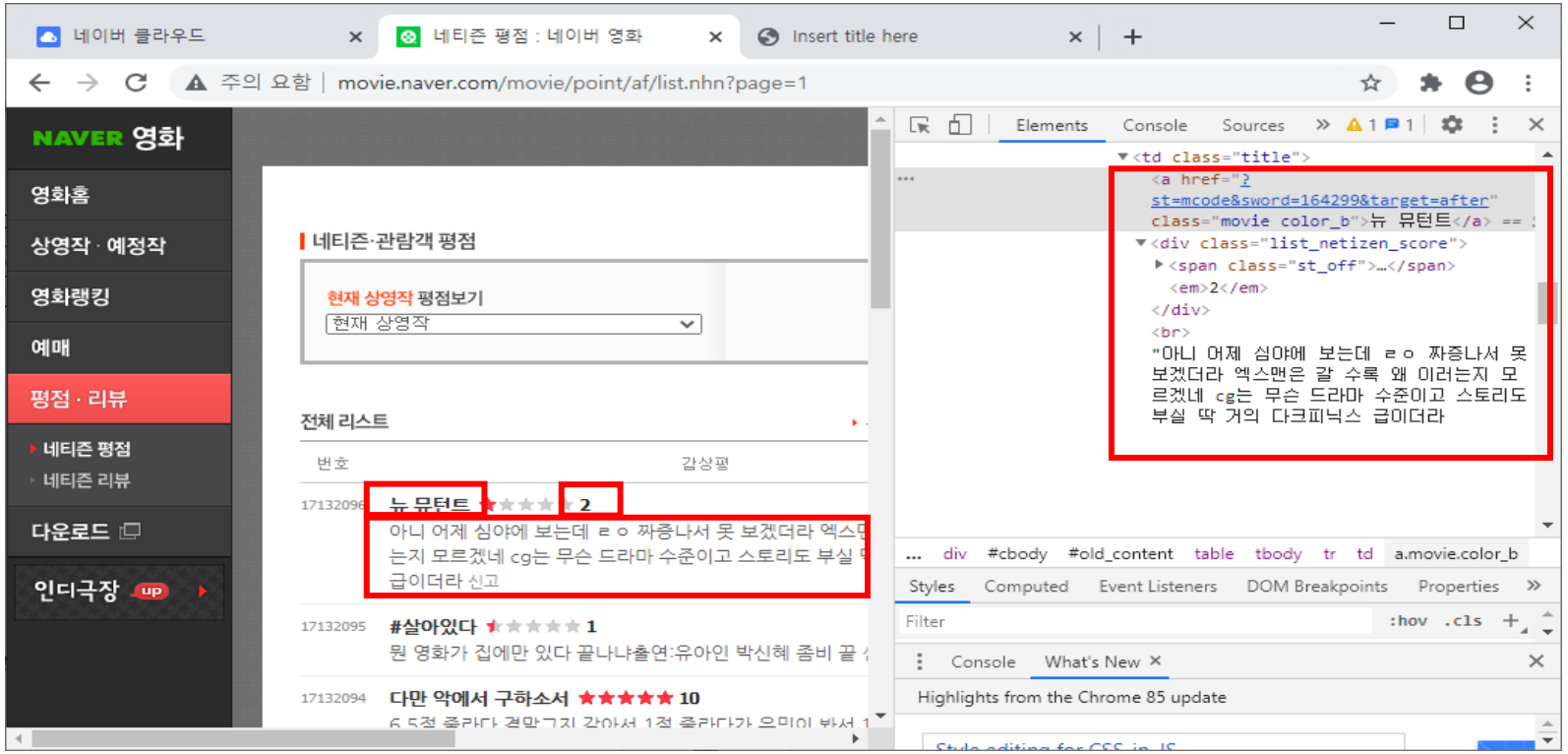
정적 스크래핑(크롤링)

(1) 네이버 영화 사이트 댓글정보 스크래핑

네이버 영화 사이트의 데이터 중 영화제목, 평점, 리뷰만을 추출하여 CSV 파일의 정형화된 형식으로 저장한다.

(1) 스크래핑하려는 웹페이지의 URL 구조와 문서 구조를 파악해야 한다.

- URL 구조 : http://movie.naver.com/movie/point/af/list.nhn?page=1



데이터 수집

정적 스크래핑(크롤링)

- 문서 구조
  영화 제목 class="movie"
  영화 평점 class=".title em"
  영화 리뷰 xpath=
  "//*[@id='old_content']/table/tbody/tr/td[2]/text()"

[rvest 패키지의 주요 함수]

html_nodes(x, css, xpath), html_node(x, css, xpath)
html_text(x, trim=FALSE)
html_attrs(x)
html_attr(x, name, default = "")

[1페이지 스크래핑]

```
install.packages("rvest");
library(rvest)
text<- NULL; title<-NULL; point<-NULL, review<-NULL; page=NULL
url<- "http://movie.naver.com/movie/point/af/list.nhn?page=1"
text <- read_html(url,  encoding="CP949"); text
# 영화제목
nodes <- html_nodes(text, ".movie")
title <- html_text(nodes); title
# 영화평점
nodes <- html_nodes(text, ".title em")
point <- html_text(nodes); point
# 영화리뷰
nodes <- html_nodes(text,
xpath="//*[@id='old_content']/table/tbody/tr/td[2]/text()")
nodes <- html_text(nodes, trim=TRUE)
review <- nodes[nchar(nodes) > 0]
review
page <- data.frame(title, point, review)
write.csv(page, "movie_reviews.csv")
```

데이터 수집

정적 스크래핑(크롤링)

[여러 페이지 스크래핑]

```
site<- "http://movie.naver.com/movie/point/af/list.nhn?page="
text <- NULL; movie.review <- NULL
for(i in 1: 100) {
  url <- paste(site, i, sep="")
  text <- read_html(url,  encoding="CP949")
  nodes <- html_nodes(text, ".movie")
  title <- html_text(nodes)
  nodes <- html_nodes(text, ".title em")
  point <- html_text(nodes)
  nodes <- html_nodes(text, xpath="//*[@id='old_content']/table/tbody/tr/td[2]/text()")
  imsi <- html_text(nodes, trim=TRUE)
  review <- imsi[nchar(imsi) > 0]
  if(length(review) == 10) {
    page <- cbind(title, point)
    page <- cbind(page, review)
    page <- data.frame(title, point, review)
    movie.review <- rbind(movie.review, page)
  } else
    cat(paste(i," 페이지에는 리뷰글이 생략된 데이터가 있어서 수집하지 않습니다.ㅜㅜ₩n"))
}
write.csv(movie.review, "movie_reviews2.csv")
```

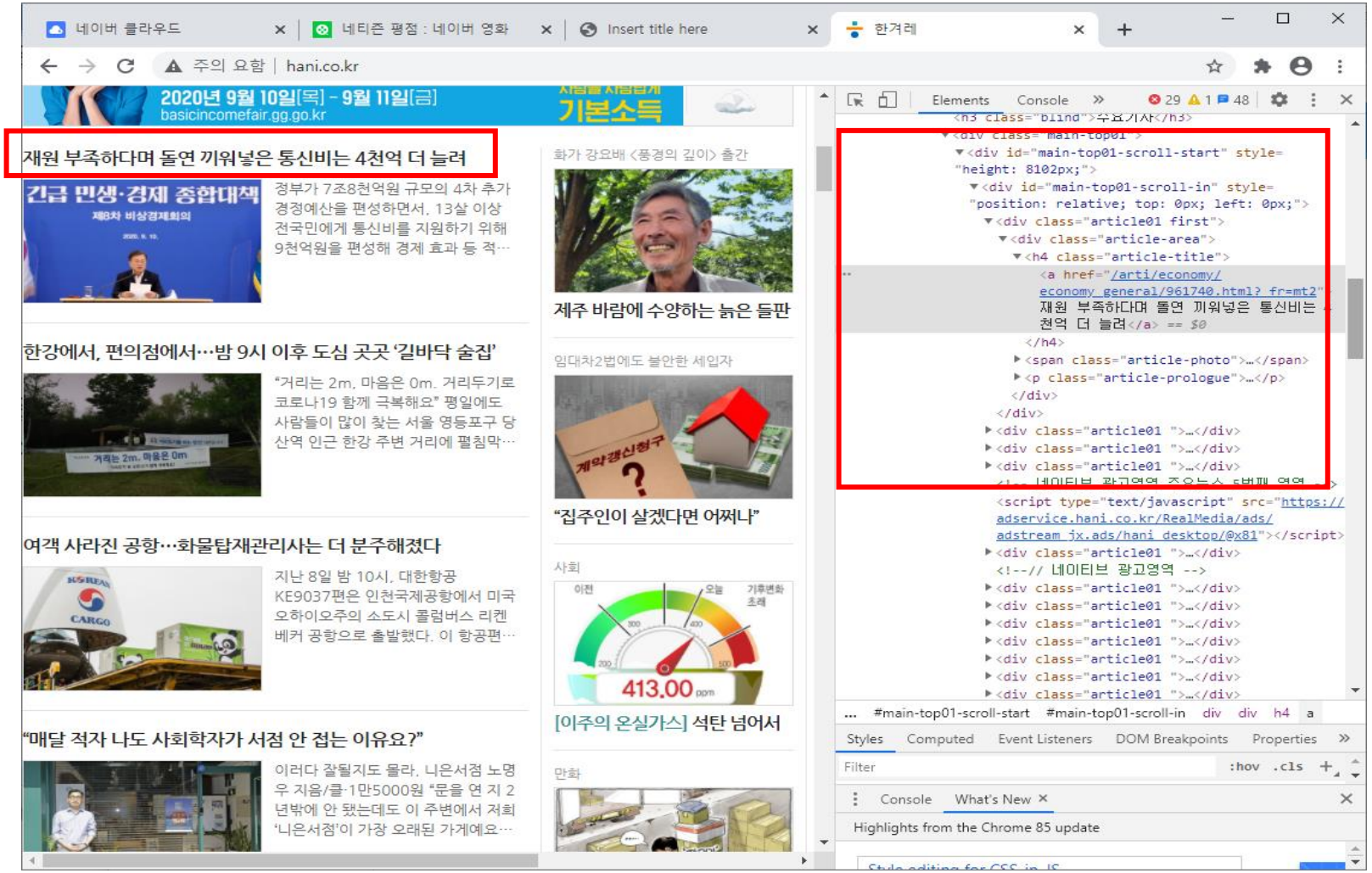
데이터 수집

정적 스크래핑(크롤링)

2. 한겨레신문

기사 제목

스크래핑



정적 스크래핑(크롤링)

[XML 패키지의 주요 함수]

htmlParse (file, encoding="...")

xpathSApply(doc, path, fun)

fun : **xmlValue, xmlGetAttr, xmlAttrs**

```
install.packages("XML")
library(XML)
imsi <- read_html("http://www.hani.co.kr/")
t <- htmlParse(imsi)
content<- xpathSApply(t,
'//*[@id="main-top01-scroll-in"]/div/div/h4/a', xmlValue);
content
content <- gsub("[[:punct:]]", "", content)
content
content
```

정적 스크래핑(크롤링)

그 외의 웹 스크래핑시 알고 있으면 도움되는 내용들

[R에서 GET으로 사이트 내용 가져오기 : httr 패키지 사용]

```
library(httr)
http.standard <- GET('http://www.worg/Protocols/rfc2616/rfc261html')
title2 = html_nodes(read_html(http.standard), 'div.toc h2')
title2 = html_text(title2)
```

[R에서 POST로 사이트 내용 가져오기 : httr 패키지 사용]

```
library(httr)

download_url = 'http://file.krx.co.kr/download.jspx'
data = POST(download_url, query = list(code = my_otp),
            add_headers(referer = otp_url))
```

정적 스크래핑(크롤링)

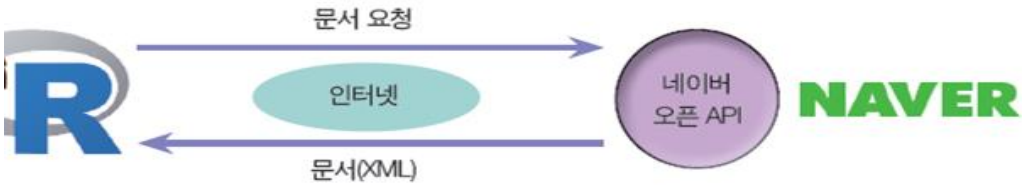
[뉴스, 게시판 등 글 목록에서 글의 URL만 뽑아내기]

```
res = GET('https://news.naver.com/main/list.nhn?mode=LSD&mid=sec&sid1=001')
htxt = read_html(res)
link = html_nodes(htxt, 'div.list_body a')
article.href = unique(html_attr(link, 'href'))
```

[이미지, 첨부파일 다운 받기]

```
# pdf
res = GET('http://cran.r-project.org/web/packages/httr/httr.pdf')
writeBin(content(res, 'raw'), 'c:/Temp/httr.pdf')
# jpg
h = read_html('http://unico201dothome.co.kr/productlog.html')
imgs = html_nodes(h, 'img')
img.src = html_attr(imgs, 'src')
for(i in 1:length(img.src)){
  res = GET(paste('http://unico201dothome.co.kr/',img.src[i], sep=""))
  writeBin(content(res, 'raw'), paste('c:/Temp/', img.src[i], sep=""))
}
```

2 네이버의 뉴스와 블로그 글 읽어오기



https://developers.naver.com/docs/search/blog/ 에서 내용 검토

API	요청	출력 포맷
뉴스	https://openapi.naver.com/v1/search/news.xml	XML
블로그	https://openapi.naver.com/v1/search/blog.xml	XML

요청 변수	값	설명
query	문자(필수)	검색을 원하는 질의, UTF-8 인코딩
display	정수: 기본값 10, 최대 100	검색 결과의 출력 건수(최대 100까지 가능)
start	정수: 기본값 1, 최대 1000	검색의 시작 위치(최대 1000까지 가능)
sort	문자: date(기본값), sim	정렬 옵션 • date : 날짜순(기본값) • sim : 유사도순

2 네이버의 뉴스와 블로그 글 읽어오기

네이버 블로그에서 "여름추천요리"를 검색하여 블로그에 올려진 데이터를 수집해본다.

[네이버 블로그 연동 : Rcurl 패키지 사용]

```
install.packages("RCurl")
library(RCurl)
library(XML)
searchUrl<- "https://openapi.naver.com/v1/search/blog.xml"
Client_ID <- "……"
Client_Secret <- "……"
query <- URLencode(iconv("여름추천요리","euc-kr","UTF-8"))
url<- paste(searchUrl, "?query=", query, "&display=20", sep="")
doc<- getURL(url, httpheader = c('Content-Type' = "application/xml",
                        'X-Naver-Client-Id' = Client_ID,'X-Naver-Client-Secret' = Client_Secret))
# 블로그 내용에 대한 리스트 만들기
doc2 <- htmlParse(doc, encoding="UTF-8")
text<- xpathSApply(doc2, "//item/description", xmlValue);
text
```

2 네이버의 뉴스와 블로그 글 읽어오기

네이버 뉴스에서 "미세먼지"로 검색하여 뉴스에 올려진 데이터를 수집해본다.

[네이버 블로그 연동 : Rcurl 패키지 사용]

```
install.packages("RCurl")
library(RCurl)
library(XML)
searchUrl<- "https://openapi.naver.com/v1/search/news.xml"
Client_ID <- "izGsqP2exeThwwEUVU3x"
Client_Secret <- "WrwbQ1l6ZI"
query <- URLencode(iconv("미세먼지","euc-kr","UTF-8"))
url<- paste(searchUrl, "?query=", query, "&display=20", sep="")
doc<- getURL(url, httpheader = c('Content-Type' = "application/xml",
                        'X-Naver-Client-Id' = Client_ID,'X-Naver-Client-Secret' = Client_Secret))
# 블로그 내용에 대한 리스트 만들기
doc2 <- htmlParse(doc, encoding="UTF-8")
text<- xpathSApply(doc2, "//item/description", xmlValue);
text
```

3 트위터 글 읽어오기

트위터에서는 twitteR 이라는 패키지를 제공하여 트위터에 올려진 글을 수집하는데 도움을 준다.

```
install.packages("twitteR")
library(twitteR)
api_key <- "....."
api_secret <- "....."
access_token <- "....."
access_token_secret <- "....."
setup_twitter_oauth(api_key,api_secret, access_token,access_token_secret)
key <- "취업"
key <- enc2utf8(key)
result <- searchTwitter(key, n=100)
DF <- twListToDF(result)
content <- DF$text
content <- gsub("[[:lower:][:upper:][:digit:][:punct:][:cntrl:]]", "", content);
content
```

setup_twitter_oauth(api_key,api_secret , access_token,access_token_secret)	현재의 R세션에 인증키를 내려받는 기능
result<- searchTwitter(key, n=100)	key 에 해당되는 트위터 글 읽어 오기
DF <- twListToDF(result)	응답 내용을 데이터 프레임으로 변환