데이터 수집

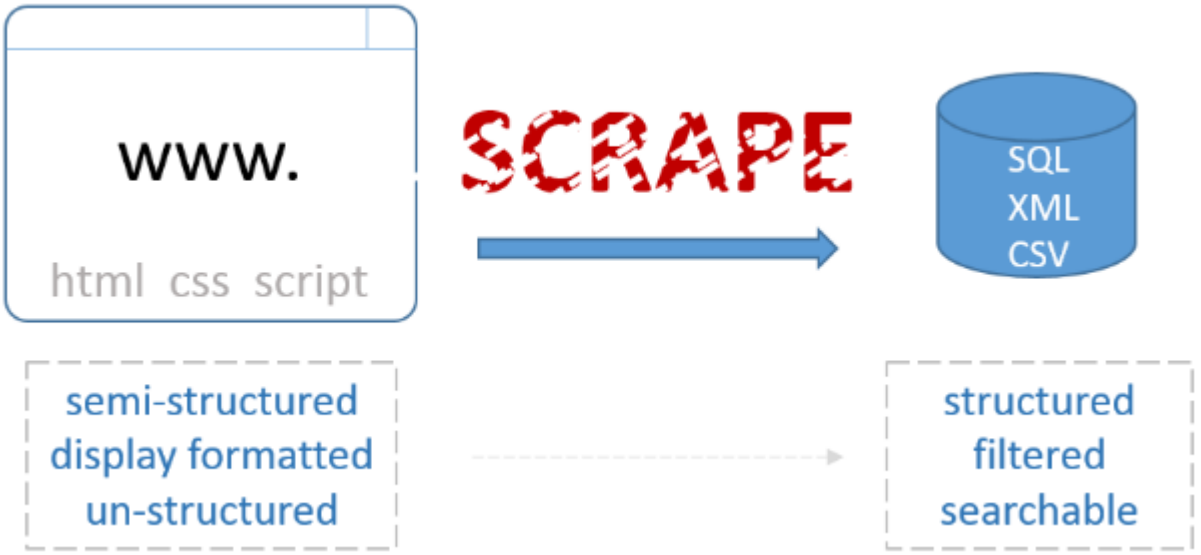
1. 정적 스크래핑(크롤링)

[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



1. 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

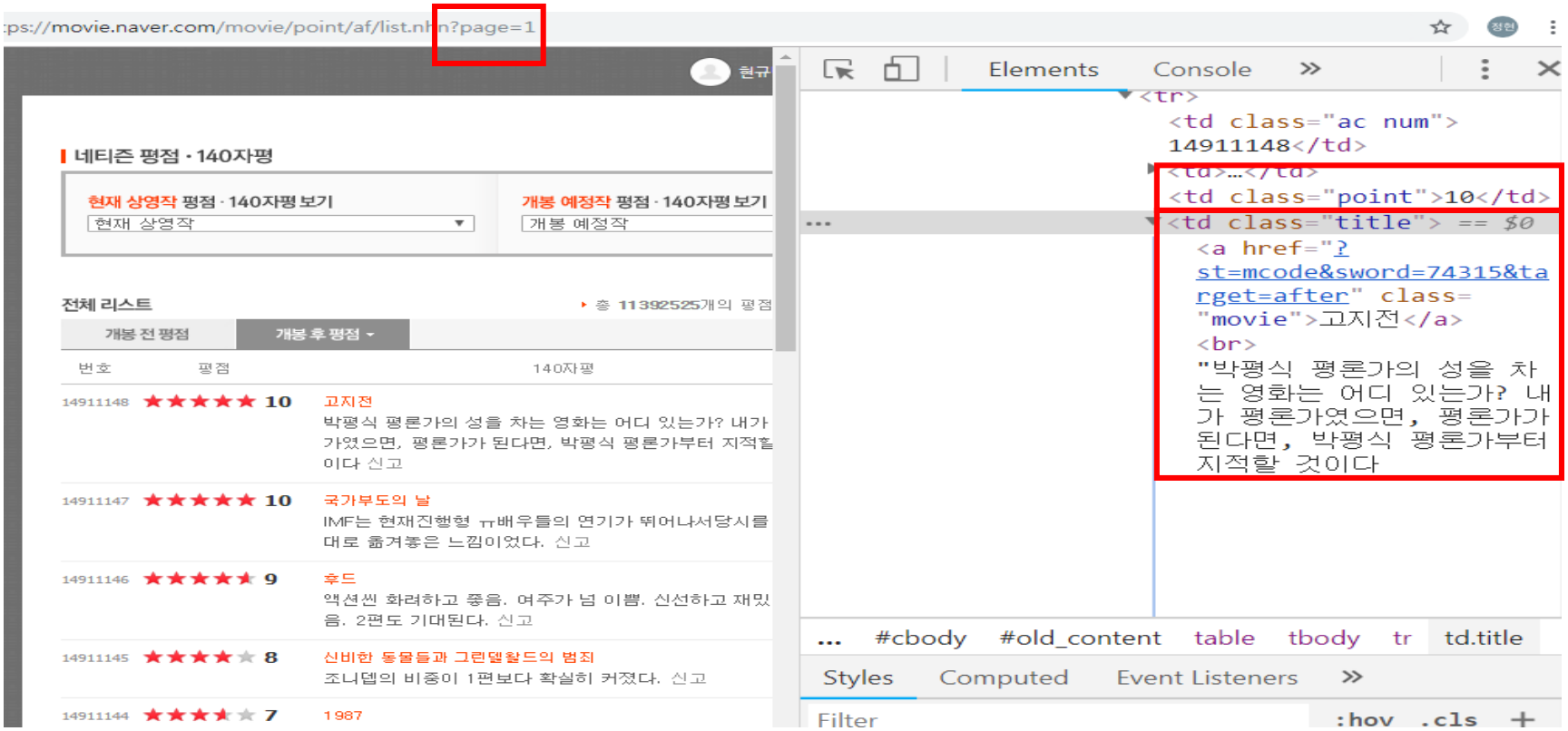
데이터 수집

1. 정적 스크래핑(크롤링)

(1) 네이버 영화 사이트 댓글정보 스크래핑

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

- 스크래핑하려는 웹페이지의 URL 구조와 문서 구조를 파악해야 한다.
  - URL 구조 : http://movie.naver.com/movie/point/af/list.nhn?page=1



데이터 수집

1. 정적 스크래핑(크롤링)

- 문서 구조
  영화 제목 class="movie"
  영화 평점 class="point"
  영화 리뷰 class="title"

[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)
url<- "http://movie.naver.com/movie/point/af/list.nhn?page=1"
text <- read_html(url,  encoding="CP949")
# 영화제목
nodes <- html_nodes(text, ".movie")
title <- html_text(nodes))
# 영화평점
nodes <- html_nodes(text, ".point")
point <- html_text(nodes)
# 영화리뷰
nodes <- html_nodes(text, ".title")
review <- html_text(nodes, trim=TRUE); review
review <- gsub("₩t", "", review)
review <- gsub("₩r₩n", "", review)
review <- gsub("신고", "", review); review
page <- cbind(title, point)
page <- cbind(page, review)
write.csv(page, "movie_reviews.csv")
```

데이터 수집

1. 정적 스크래핑(크롤링)

[여러 페이지 스크래핑]

```
site<- "http://movie.naver.com/movie/point/af/list.nhn?page="
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, ".point")
    point <- html_text(nodes)
    nodes <- html_nodes(text, ".title")
    review <- html_text(nodes, trim=TRUE)
    review <- gsub("₩t", "", review);  review <- gsub("₩r₩n", "", review)
    review <- gsub("신고", "", review)
    page <- cbind(title, point)
    page <- cbind(page, review)
    movie.review <- rbind(movie.review, page)
}
write.csv(movie.review, "movie_reviews2.csv")
```

데이터 수집

1. 정적 스크래핑(크롤링)

한국일보 헤드라인
기사 스크래핑

1. 정적 스크래핑(크롤링)

[XML 패키지의 주요 함수]

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

xpathSApply(doc, path, fun)

fun : **xmlValue, xmlGetAttr, xmlAttrs**

```
library(XML)
t <- htmlParse("http://hankookilbo.com")
content<- xpathSApply(t,"//p[@class='title']", xmlValue);
Content
content <- gsub("[[:punct:][:cntrl:]]", "", content)
content
content <- trimws(content)
content
```

1. 정적 스크래핑(크롤링)

[Xpath]

XPath란?

XPath란 XML Path Language를 의미한다.

XPath는 XML 문서의 특정 요소나 속성에 접근하기 위한 경로를 지정하는 언어이다.

/bookstore	**최상위 엘리먼트로서의 bookstore 태그를 찾는다.**
/bookstore/book	**최상위 엘리턴트 bookstore 태그를 찾은 후에 book 이라는 자식 태그를 찾는다.**
/bookstore//book	**최상위 엘리턴트 bookstore 태그를 찾은 후에 book 이라는 자손 태그를 찾는다.**
//book	**조상이 누구든 book이라는 태그를 찾는다.**
//@lang	**lang이라는 속성을 갖는 태그를 찾는다.**
//title[@lang='en']	**lang이라는 속성의 값이 'en' 인 title 태그를 찾는다.**

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
<book>
  <title lang="en">Harry Potter</title>
  <price>29.99</price>
</book>
<book>
  <title lang="en">Learning XML</title>
  <price>39.95</price>
</book>
</bookstore>
```

1. 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)
# POST 함수를 이용해 모바일 게임 랭킹 10월 29일 주  모바일 게임 랭킹을 찾는다
#(http://www.gevolution.co.kr/score/gamescore.asp?t=3&m=0&d=week)
game = POST('http://www.gevolution.co.kr/score/gamescore.asp?t=3&m=0&d=week',
            encode = 'form', body=list(txtPeriodW = '2018-10-29'))
title2 = html_nodes(read_html(game), 'a.tracktitle')
title2 = html_text(title2)
title2[1:10]
```

1. 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=""))
}
```