http 의 표준 포트번호는 80임

즉, <http://www>.사이트주소.com:포트번호/

ex ) <http://www.naver.com/> -> <http://www.naver.com:80/> 임

그래서 원래는 특정사이트에 접속하고 싶으면 도매인, 포트번호를 알아야함.

서울시 빅데이터는 80포트번호를 안쓰기 때문에 접속할 때 포트번호를 입력해서 접속해야됨.

브라우저에서 요청방식을 확인할 수 있음

개발자도구 -> name에서 보낸 주소 클릭 -> header-general 확인

query문자열이나 form data가 서버로 전달될 때의 rule

1. name = value 형식을 갖춰야함
2. 여러 개의 name = value는 & 기호로 구분한다.
3. 영문 대소문자, 숫자, 일부 특수문자는 그대로 전달, 그 외 문자는 %16진수 코드값으로 전달됨(이 경우 utf-8로 전송됨)
4. 공백은 +기호로 전달되거나 %16진수 코드값으로 전달된다.

POST방식은 반드시 form tag를 사용하여 전송해야 됨.

rvest패키지는 get방식만 지원함

post방식을 크롤링해야될 때는 httr패키지의 post함수를 사용해야함

[추가로 전달하는 문자열]

get방식 : 요청 URL 문자열 뒤에 ? 기호 뒤에

post방식 : 요청 바디

[오픈API]

네이버 블로그, 뉴스

트위터

공공DB

네이버 오픈 api는 주로 url형식으로 지원해줌

URencode 함수는 기본적으로 euc-kr(ms-949)로 인식함, 그래서 utf8로 바꿔서 사용해야됨

Xpath 가 뭐지?

XmlValue하는 것도 알아보기

트위터는 자체 패키지를 제공함

처음 4줄을 실행하면 인증을 해주는데 인증해준 파일을 컴퓨터에 가지고 있으면 다시 시행할 필요가 없음.

기본적으로 트위터를 긁어오면 list형식으로 제공함

Xml이 조금더 엄격하기 때문에 html과 혼용할경우 문제가 될 수 있지만 반대의 경우는 괜찮음

2019-10-31 [13:00 ~ 14:50]

실습문제 10-31 2번 문제 풀고 확인

[15:00 ~ 15:50]

[바탕화면 새폴더 -> R데이터수집]

@ Open Api와 rest api특성 설명 (11page)

Open Api : R에 들어있는 트위터 함수 같은 것들 (함수 형식)

Rest Api : 웹을 사용해서 (URL문자 사용) 정보를 제공받는 것 (네이버, 서울시 오픈API)

하지만 요즘은 rest -> open 으로 포함해서 많이 부름

@ 인증키 받아오는법 설명 (12 ~ 25page)

@ 동적크롤링 하는 법 (30 ~ 39)

웹페이지가 HTML, CSS로만 이뤄져있으면 정적크롤링만으로 충분히 크롤링이 가능함

하지만 JavaScript가 포함되어있으면 웹브라우저가 스스로 작동해서 만들어낸 친구들이기

때문에 그정보를 받으려면 동적크롤링을 해야됨. 이 때 새폴더에 넣어놓은 selenium 이

라는 프로그램의 도움을 받아야함.

[ex <http://www.yes24.com/Product/goods/40936880>]

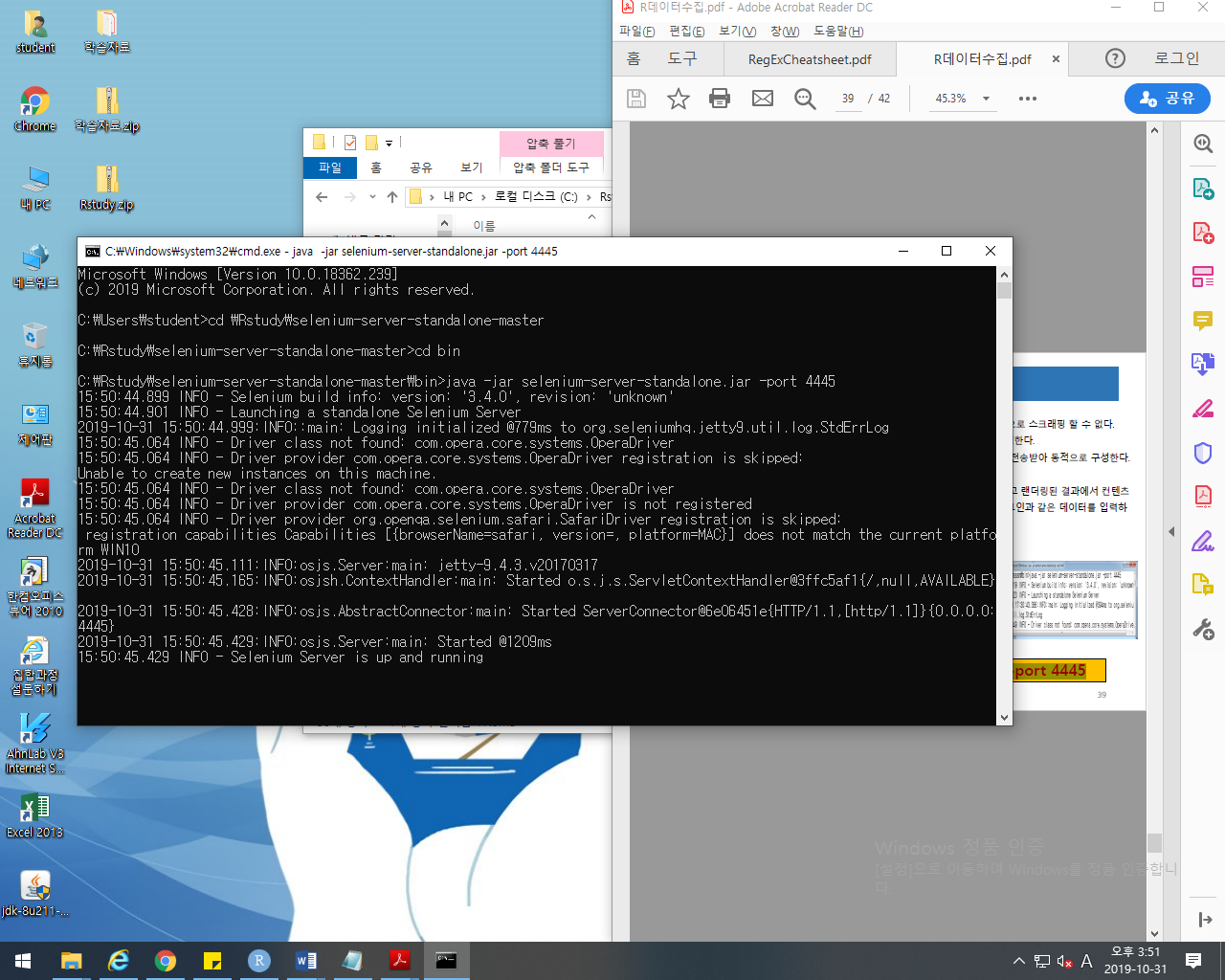
* 리뷰 긁어오려고 할 때 페이지를 넘겨도 URL이 변하지가 않음. 또 리뷰글 마지막에 …으로 생략되어있음. 펼쳐보기를 눌러야 제대로 읽어볼 수 있음.
* AJAX라는 기술을 사용하면 이런 현상이 나타남 AJAX라는 기술은 페이지에서 변해야 되는 부분만 바꿔주는 기술임 (AJAX를 사용하기 전에는 웹페이지에서 1가지만 변해도 전체페이지를 다바꿔야했음)
* 이 page에서 도구 > 개발자도구 > network창을 띄우고 다른페이지를 눌러본다.
* 그때마다 새로운 것이 생기는데 type이 xhr인 친구를 눌러본다. 그 다음에 preview를 눌러보면 이 항목이 리뷰를 가져오고 있는 것을 알 수 있다.
* AJAX를 사용하기 전을 web1.0, 사용 후 web2.0으로 구분한다. 그만큼 중요한 기술임
* 리뷰 약간을 복사하고, 페이지 소스보기 눌러서 검색해보면 그 리뷰내용을 찾을 수가 없음. 왜냐하면 처음 웹페이지를 로드해 올 때 가져온 내용이 아니기 때문에 (AJAX로 가져왔거나, JavaScript로 만들어진 내용) 찾을 수가 없는 것임. 화면에만 보여짐. (ex. 시간출력한 실습 생각하면됨. 직접 시간을 입력안하고 javaScript로 시간이 출력되도록 하는 경우 코드에 출력값이 나타나지 않는 것)
* 주의 할 점! 개발자도구의 element탭에서 보면 내용이 보이는데 이 내용은 이미 코드들을 랜더링해서 보여주는 것이기 때문에 서버에서 온건지 클라이언트에서 직접 계산돌린것인지 확인이 안되기 때문에 (즉 정적, 동적 컨탠츠인지 구분이 안됨), 우클릭 후 페이지 소스보기로 가서 소스에 그 값이 보여지는지 확인하는 것이 중요함.

31page와 32page를 비교하면 정적컨텐츠와 동적컨텐츠를 구분하는 법이 나와있음

33page부터는 스타벅스 매장찾는 법을 예시로 들어놓음 이 경우 클릭을 여러 번해야지 원하는 결과를 얻는데 이것들을 많이 반복해야되니까 자동화 할 필요가 있음. 이 경우 selenium이라는 프로그램을 활용한다.

selenium이라는 프로그램은 가상으로 브라우저를 돌리는 역할을 한다. 주로 성능테스트용도로 많이 씀. (39page를 보고 설치한다 – 설치完)

실행시키려면 윈도우 > 실행 > cmd실행 > cd \Rstudy\se 입력하고 tab키 누르면 나머지뜸 엔터 입력 후 다음 창에 cd bin 입력 엔터, 그 다음 39page에서 복사한 것을 붙여넣고 엔터



<이 화면 나오면 정상적으로 가동된 것임>

[16:00 ~ 16:50]

@ 동적크롤링 실습 (코드 작성)

#동적크롤링 실습

install.packages("RSelenium")

library(RSelenium)

remDr <- remoteDriver(remoteServerAddr = "localhost", port = 4445, browserName = "chrome") # 셀레늄 서버에 대해 접속하도록 설정한거임

remDr$open() #리모트 서버를 만들고 그 서버를 이용해서 브라우저를 열어줘 라는 의미임

remDr$navigate("http://www.google.com/ncr") # 그 브라우저를 이용해서 이 사이트를 열어줘

webElem <- remDr$findElement(using = "css", "[name = 'q']") #이름이 q인 엘리먼트를 찾아라

webElem$sendKeysToElement(list("JAVA",key = "enter")) #리턴받은 엘리먼트에 키를 입력해라 (list로 처음엔 입력하고 싶은 검색어, 그다음엔 어떤

키를 줄지 만들면됨)

@ 네이버접속 및 검색 실습해보기

[17:00 ~ 끝]

키 입력은 위에서 실습을 했으니 이제 클릭이벤트로 넘어감

Selenium을 활용하여 <http://comic.naver.com/comment/comment.nhn?titleId=570503&no=135>에 접속하기 -> 여기서는 best댓글만 보여줌 전체댓글을 보고싶을 때는 전체댓글이라는 로드를 클릭해주면 나옴 (이경우 URL은 변하지 않음) 즉, R로 전체댓글을 누르게 만들어야됨

#클릭이벤트 실습

remDr$navigate("http://comic.naver.com/comment/comment.nhn?titleId=570503&no=135")

방법1.

more <- remDr$findElements(using="css",'#cbox\_module > div > div.u\_cbox\_sort > div.u\_cbox\_sort\_option > div > ul > li:nth-child(2) > a > span.u\_cbox\_sort\_label')

sapply(more,function(x){x$clickElement()})

방법2.

more <- remDr$findElement(using="css",'#cbox\_module > div > div.u\_cbox\_sort > div.u\_cbox\_sort\_option > div > ul > li:nth-child(2) > a > span.u\_cbox\_sort\_label')

more$clickElement()

방법1은 복수형 : 복수형은 여러 요소를 찾아와서 모두 클릭한다 (sapply()가 반복문대체하는 거임 각 원소를 가지고 각각 함수시행해줌)

방법2는 단수형 : 한 개 요소 찾아서 하나 클릭하는거

전체댓글에서 2, 3페이지로 이동하도록 클릭이벤트 발생시키기

#2페이지

two <- remDr$findElements(using="css", "div > a:nth-child(4) > span")

sapply(two, function(x){x$clickElement()})

#3페이지

three <- remDr$findElement(using="css", " div > a:nth-child(5) > span")

three$clickElement()

규칙이 있으니까 자동으로 10페이지까지 이동하도록 for구문 사용하여 만들어보기

#자동으로 10페이지까지

for (i in 4:12) {

nextCss <- paste0("div > a:nth-child(",i,") > span")

nextPage <- remDr$findElement(using="css",nextCss)

nextPage$clickElement()

Sys.sleep(3)

}