Jsoup

Jsoup 개요
Jsoup 요소
URL 접속해 결과 얻어오기
문서 파싱

1. Jsoup 개요

- 자바 HTML 파서(Java HTML Parser)
  - https://jsoup.org/
  - 오픈 소스 프로젝트로 제공
- jsoup 주요 기능
  - URL, 파일, 문자열을 소스로 하여 HTML 파싱
  - DOM 구조를 추적하거나 익숙한 CSS 선택자를 사용하여 데이터를 찾아 추출
  - 문서내의 HTML 요소, 속성, 텍스트를 조작
  - 사용자가 입력한 데이터로부터 XSS(Cross-Site Script) 공격을 방지하기 위해서 안전한 화이트 리스트 방식으로 지정된 태그만 남기고 나머지는 제거
  - 깔끔한 형태의 html을 출력

2. Jsoup의 주요 요소

- jsoup 요소

클래스명	설명
Document	Jsoup 얻어온 결과 HTML 전체 문서
Element	Document의 HTML 요소
Elements	Element가 모인 자료형. for나 while 등 반복문 사용이 가능하다.
Connection	Jsoup의 connect 혹은 설정 메소드들을 이용해 만들어지는 객체, 연결을 하기 위한 정보를 담고 있다.
Response	Jsoup가 URL에 접속해 얻어온 결과. Document와 다르게 status 코드, status 메시지나 charset같은 헤더 메시지와 쿠키등을 가지고 있다.

3. URL 접속해 결과(Document) 얻어 오기

- Document를 얻어내기
  - GET,POST

```
Document google1 = Jsoup.connect("http://www.google.com").get();
Document google2 = Jsoup.connect("http://www.google.com").post();
```

  - Response

```
Connection.Response response = Jsoup.connect("http://www.google.com")
                    .method(Connection.Method.GET)
                    .execute();
Document google3 = response.parse();
```

3. URL 접속해 결과(Document) 얻어 오기

- 얻어낸 Document 두가지 방법으로 출력
  - .html(), .text() 사용

```
Connection.Response response = Jsoup.connect("http://www.google.com")
            .method(Connection.Method.GET)
            .execute();
Document document = response.parse();

String html = document.html();
String text = document.text();
```

3. URL 접속해 결과(Document) 얻어 오기

- **얻어온 결과에서 특정 값 뽑아내기**
  - **.select("css query") 메소드 사용**

```
Connection.Response response = Jsoup.connect("http://www.google.com")
            .method(Connection.Method.GET)
            .execute();
Document googleDocument = response.parse();
Element btnK = googleDocument.select("input[name=btnK]").first();
String btnKValue = btnK.attr("value");

System.out.println(btnKValue); // Google 검색
```

3. URL 접속해 결과(Document) 얻어 오기

- 문서의 내부 - 요소 찾기
  - getElementById(String id) :
    - Element 객체를 반환합니다. 하나를 반환. 없으면 null 을 반환.
  - getElementsByTag(String tag)
    - Elements 객체를 반환. 없으면 size() 가 0 임.
  - getElementsByClass(String className) :
    - Elements 객체를 반환. 없으면 size() 가 0 임.

3. URL 접속해 결과(Document) 얻어 오기

- **Element 객체가 할 수 있는 작업**
  - **attr(String key)** 로 속성의 값을 얻기
  - **attr(String key, String value)**로 속성의 각을 설정.
  - **id(), className()** 은 id와 class속성의 값을 가져 옴. class는 여러개 지정되면 하나의 문자열로 반환. 예로 요소가 ** ** 라면 **className()** 은 **"center red"** 를 반환합니다. 하나씩 구하기 위해서는 **classNames()** 메소드를 사용합니다. **Set<String>** 타입으로 반환합니다.
  - **text()**로 순수 텍스트만 구함.
  - **text(String value)**로 요소의 텍스트를 설정.
  - **html()**로 html 문자열을 구함
  - **html(String value)** 메소드로 **inner HTML** 을 설정.
  - **outerHtml()** 요소의 outer html을 반환.

- inner HTML 은 요소가 포함하는 html을 나타내고, **outer html** 은 요소 자체 태그까지 포함하는 것.

- <img>태그 요소 중 src속성 값 구하기

```
Elements imgs = doc.getElementByTag("img");
if(imgs.size() > 0) {
        String src = imgs.get(0).attr("src');
 }
```

```
Element img = doc.getElementByTag("img").first();
if(img != null) {
        String src = img.attr("src");
 }
```

HTML 과 text 조작하기

- **append(String html), prepend(String html)** : 선택된 요소의 뒤(append)와 앞(prepend)에 html 을 추가.
- **appendText(String text), prependText(String text)** : 선택된 요소의 뒤(append)와 앞(prepend)에 text를 추가.
- **appendElement(String tagName), prependElement(String tagName)** : 선택된 요소의 뒤(append)와 앞(prepend)에 Element를 추가.
- **html(String value)** : 선택된 요소에 inner html 을 설정.
- **remove()** : 선택된 요소를 삭제.

```
Elements tables = doc.select("table");
for(Element table : tables) {
        table.remove();
}
```

CSS 스타일로 요소를 선택하기

doc.select("a") : <a> 요소를 모두 선택
doc.select("#logo") : id="logo" 인 요소를 선택
doc.select(".head") : class="head"인 요소들을 선택
doc.select("[href]") : href 속성을 가진 요소들을 선택
doc.select("[width=500]") : width 속성의 값이 500인 모든 요소들을 선택

-
doc.select("div").select(".head").select("[width=500]");

- div 요소들중에서 class가 "logo" 가 아닌것들 을 선택
Elements divs = doc.select("div").not(".logo");