20190701

**Chapter 01 크롤링, 스크래핑**

**스크래핑(scraping)**

-웹사이트의 특정 정보를 추출하는 기술

-공개된 정보는 대부분 HTML형식으로 되어있기 때문에 이들 중 필요한 데이터를 저장하기 위해 데이터 가공이 필요

-최근에는 절차가 까다로워져 로그인 이후 필요한 웹페이지 접근 기술 필요

ex. HTML

**크롤링 (crawling)**

-프로그램이 웹사이트 정보를 정기적으로 추출하는 기술

🡪 크롤링을 하는 프로그램 : 크롤러 / 스파이더

**웹컴포넌트 : HTML과 HTTP:HTML**

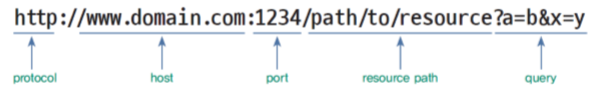
- markup language

-태그 : < >로 둘러싸여 있고, 그 안에 정보에 대한 의미를 작성. 의미가 끝나는 부분에 /사용하여 종료

-HTTP (Hypertext Transaction Protocol) : 인터넷에서 컴퓨터 간 정보를 주고받을 때 사용하는 일종의 약속 🡪 프로토콜(protocol)

**웹의 동작 순서**

- 웹브라우저를 시작하고 주소정보를 입력. URL(Uniform Resource Locator)



- 일반적으로 컴퓨터는 인터넷 프로토콜 주소(Internet Protocol address, IP address)를 가짐

- 도메인 네임 서버(Domain Name Server, DNS)를 운영해 도메인 네임과 IP주소를 연결

**웹상의 정보를 추출하는 방법**

- urllib 라이브러리를 사용 🡪 HTTP / FTP를 이용해 데이터를 다운로드

- urllib는 URL을 다루는 모듈을 모아 놓은 패키지

🡪 특히 urllib.request는 웹 사이트에 있는 데이터에 접근하는 기능을 제공

**\* 설치 : pip install request, pip install beautifulsoup4**

|  |  |
| --- | --- |
| 함수 | 설명 |
| urlretrieve(url, name) | URL주소의 파일을 다운로드 |
| urlopen() | 곧바로 파일을 저장하지 않고 메모리상에 load |

1. urlopen() 은 binary 형태로 저장됨 (b’로 시작)
2. .read() 메서드 통해서 읽어줘야함
3. .decode(‘utf-8’)을 통해 문자열로 바꿀 수 있음

cf.

<a class = >

a태그가 보통 링크 주소 href= 속성 주소로 이동

-20190701 스크래핑, 크롤링.ipynb

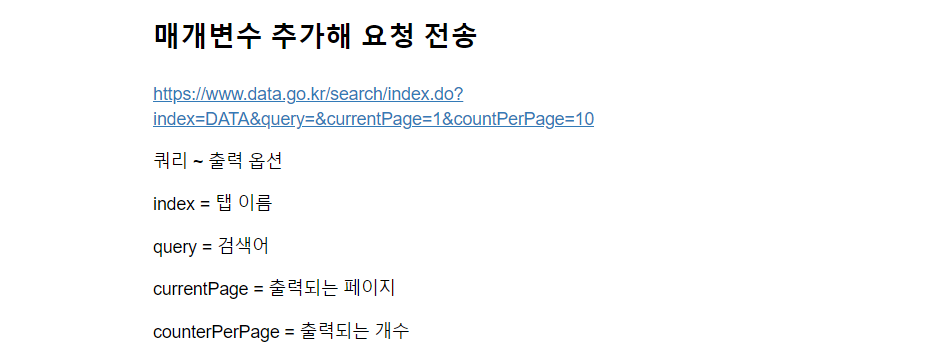


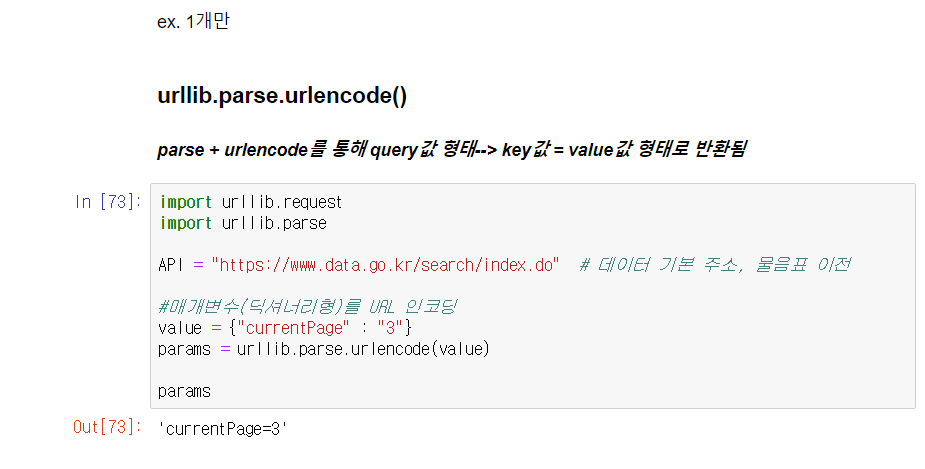


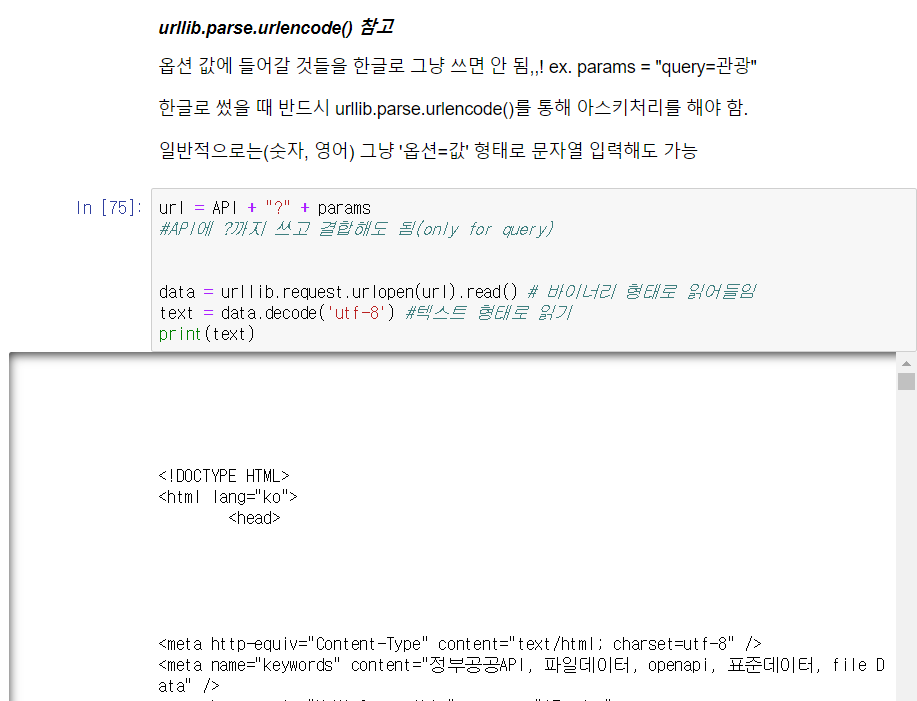


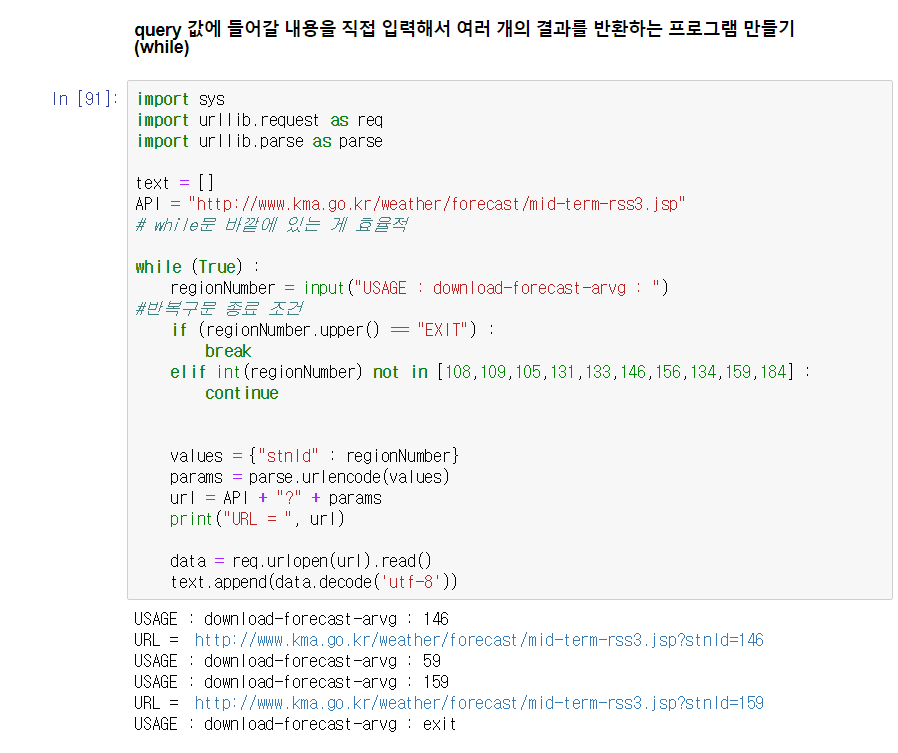


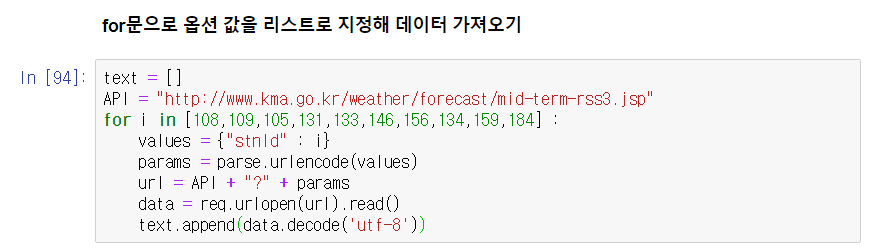












**BeautifulSoup로 스크래핑하기**

-스크래핑 : 웹사이트에서 데이터를 추출하여 원하는 정보를 얻어내는 것

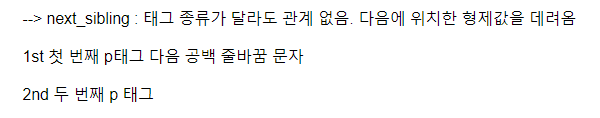
-HTML과 XML 분석을 해주는 라이브러리

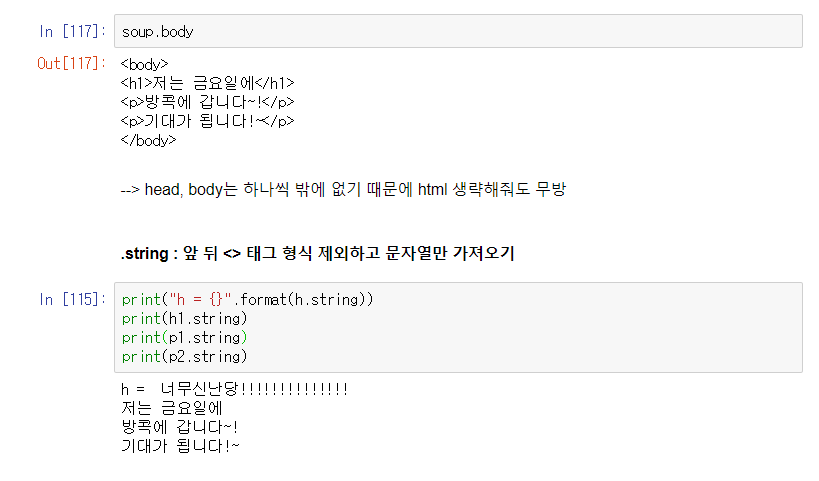
-자체로 다운로드 기능은 없음

|  |  |
| --- | --- |
| markup parser | 설명 |
| html.parser | 기본옵션으로 빠르지만 유연하지 못함  (단순한 html문서에서 사용) |
| lxml | 매우 빠르고 유연 |
| xml | xml 파일에만 사용 |
| html5lib | 매우 느리지만 유연  (구조가 복잡한 html 문서에 사용) |



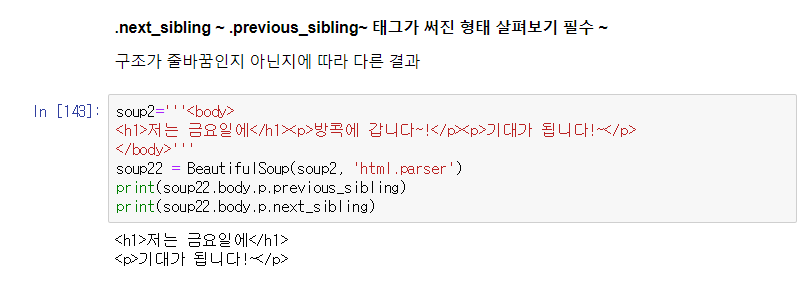






추가

.next\_sibling , .previous\_sibling

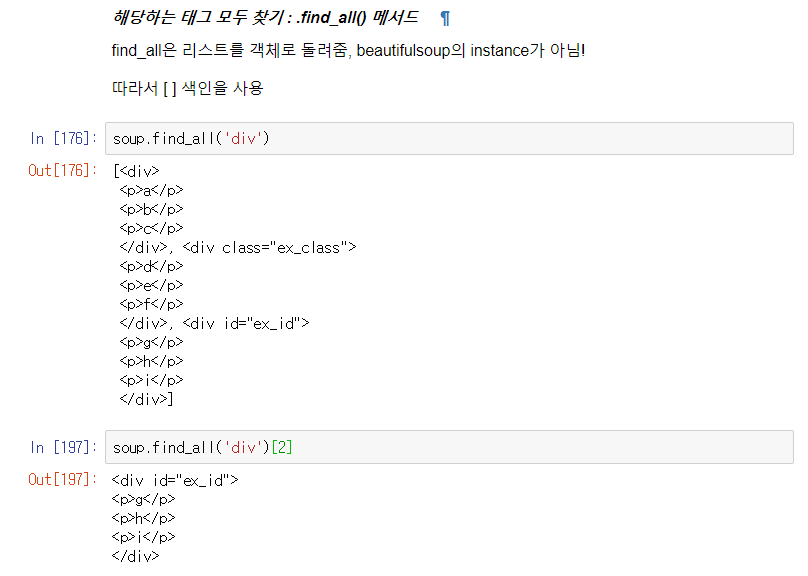


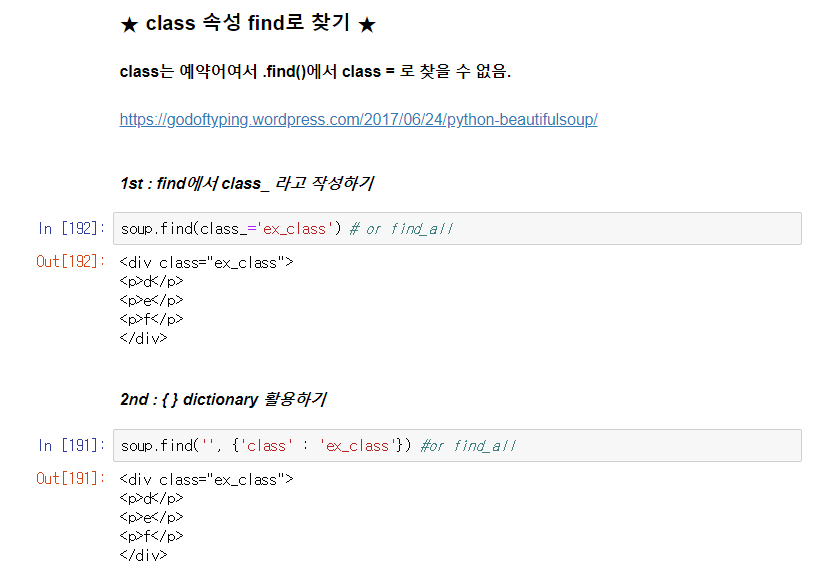
띄어쓰기 안한 경우 한 번 next\_ or previous\_sibling 은 \n이 아닌 그 전 값!

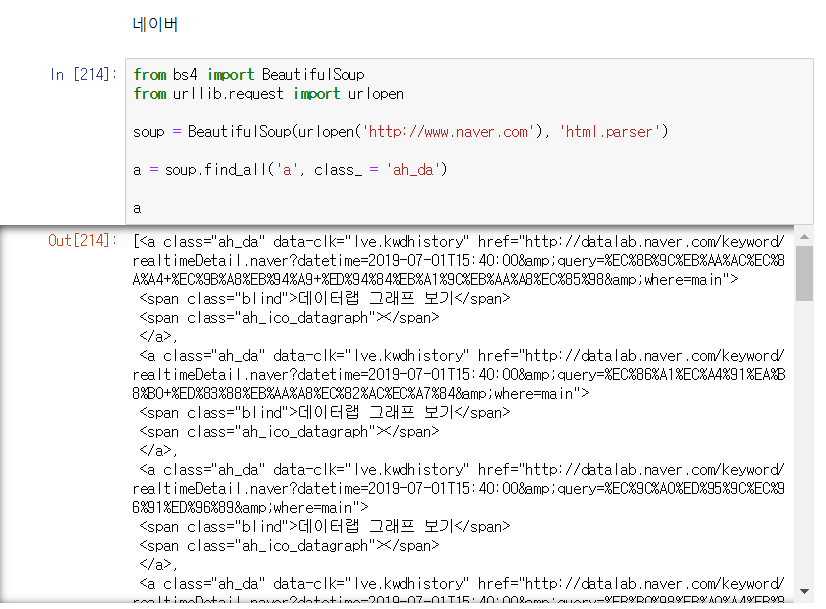














**CSS 선택자 (cascading style sheets)**

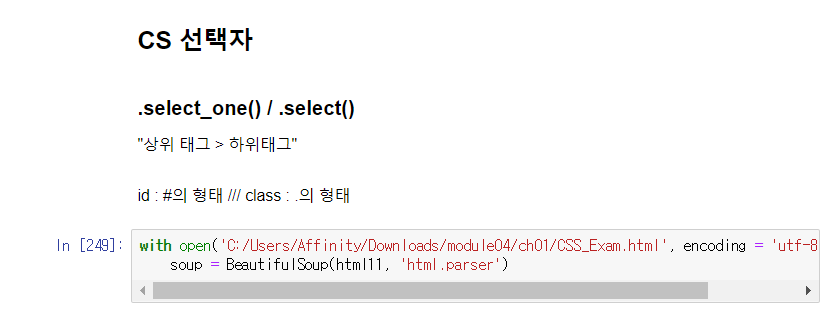
-CSS는 웹문서의 전반적인 스타일을 미리 저장해 둔 스타일 시트로, 문서 전체의 일관성을 유지할 수 있고 세세한 스타일 지정의 필요를 줄어들게 함

-CSS선택자를 지정해서 원하는 요소를 추출

- copy selector를 통해 형식 가져오기

-클래스 / id의 경우 띄어쓰기가 있는 속성은 공백 앞부분만 써줘도 검색 가능하다.

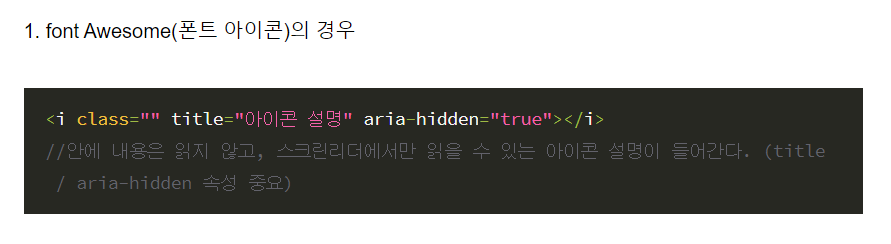
|  |  |
| --- | --- |
| 메소드 | 설명 |
| soup.select\_one(<선택자>) | CSS선택자로 요소 하나를 추출 |
| soup.select(<선택자>) | CSS 선택자로 요소 여러 개를 리스트로 추출 |



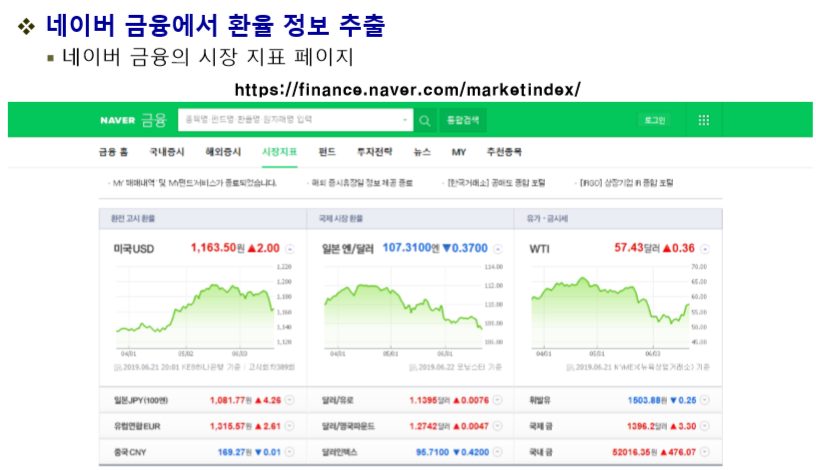




cf. <i> </i>



<i> </i> + aria-hidden 속성 🡪 숨기기

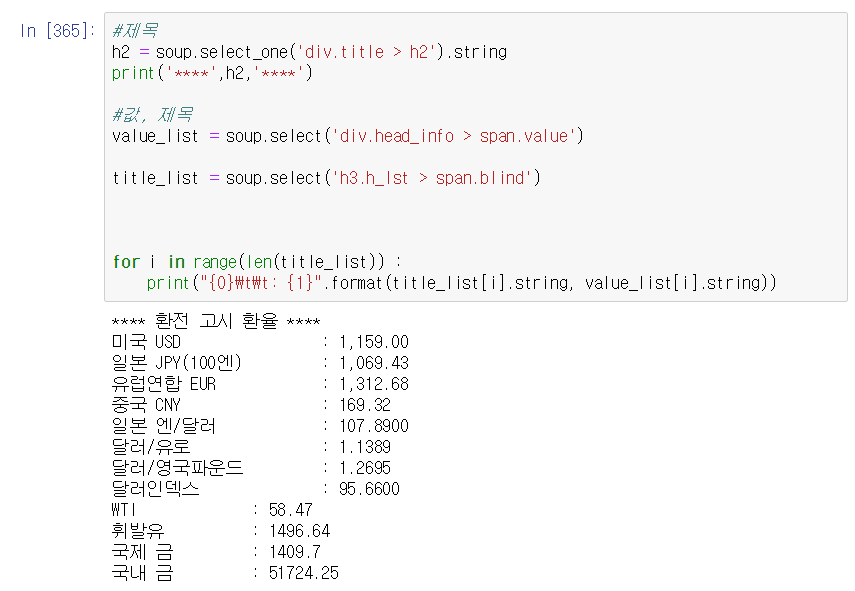


# 개발자 도구-> 오른쪽 마우스 -> copy selector를 선택한 후 공통 분모 묶기

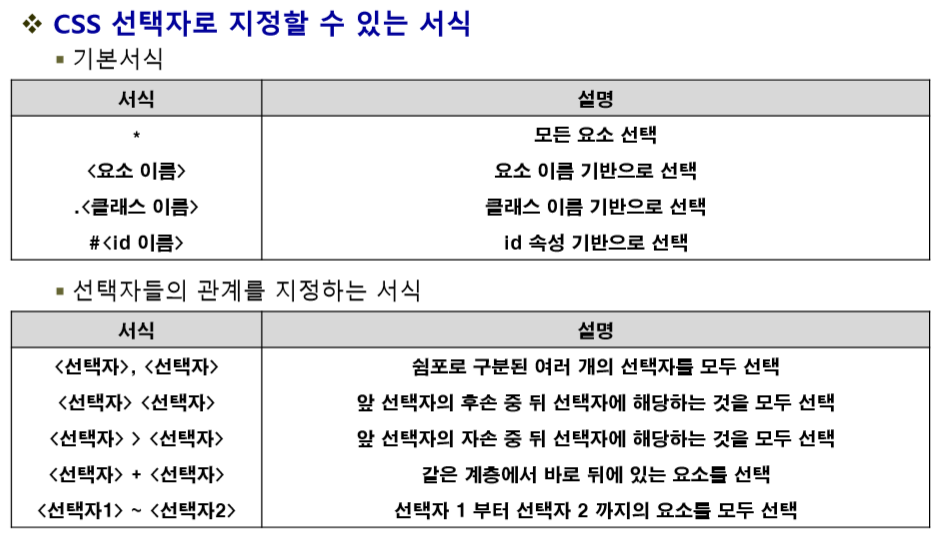
# > 는 바로 직계! 하위!(자식) 넉넉하게 검색할 때에는 띄어쓰기만 사용





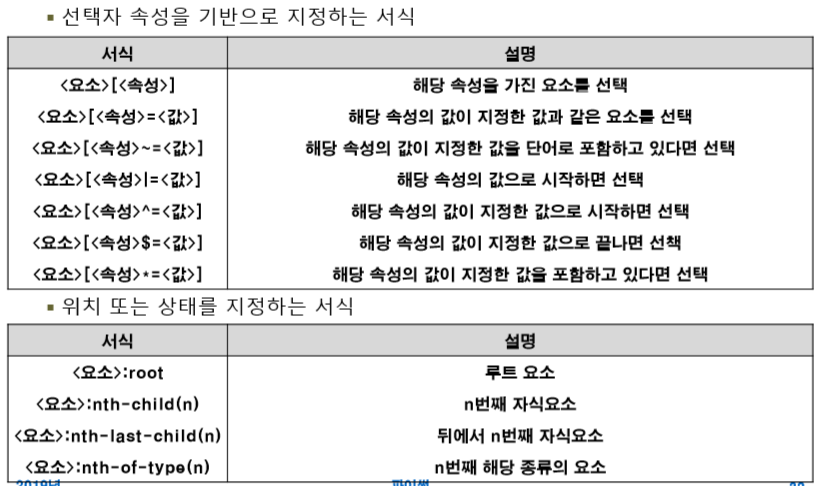


190702 화



> : 직계(자식)--명시적으로 바로 하위태그임을 표시

: 후손,자손-- 하위태그의 하위태그도 포함 (공백)



**# 빨간네모는 패턴**

**#nth-child(n) : 전체 태그 중 n번째 자식**

**#nth-of-type(n) : 해당 타입 태그의 n번째 -- 중간에 다른 데이터가 섞여있으면 nth-of-type과 다른 값**

**#이 때 n은 1부터 count (파이썬과 다름)**

