

0517

연합세션#2 :Crawling

LIKE LION

EWHA x SOGANG

5th 엄서영

Crawling = Web Scraping

Web scraping, web harvesting, or web data extraction is data scraping used for extracting data from websites.

쉽게 말해, 웹 사이트에서 원하는 데이터를 추출하는 것!



0517



Nokogiri gem

Why Nokogiri?

1. 루비는 Nokogiri를 가장 많이 사용!
2. XML/HTML 파일을 굉장히 빠른 속도로 검색/파싱
3. Css셀렉터와 Xpath기능 지원

Parsing?

파싱은 말 그대로 분석하는 것.

웹 페이지를 분석하여 원하는 정보들을 추출할 수 있도록 도와줌.

원하는 데이터를 뽑기 전, url에 해당하는 html문서를 가져오는 것.

0517



메일 카페 블로그 지식IN 쇼핑 Pay TV 사전 뉴스 증권 부동산 지도 영화 뮤직 책 웹툰 더보기

1 민유라

1시간만! 프리미엄피자, 오늘 예약주문 건까지!
 35,900원 → **17,232원**
 LAJON 기준 방문포장 최대 40% 할인, 주유비 최대 20% 할인 등 모든 혜택 반영 시

연합뉴스 > 北 이르면 열흘후 종계리 핵실험장 폭파...생중계 안할듯 네이버뉴스 연예 스포츠 경제 랭킹

뉴스스탠드 > 전체 언론사 MY 뉴스

| | | | | | |
|-------|----------------|--------|-----------|-----------|--------------|
| 스포츠동아 | JOONGANG DAILY | 아이뉴스24 | mydaily | 에브리데이경제 | 매일경제 |
| 스포츠조선 | 조선일보 | 뉴시스 | Net Korea | KBS WORLD | sportalkorea |

아이디 로그인 P보안 ON

비밀번호 일회용 로그인

☐ 로그인 상태 유지 ☐ 회원가입 ☐ 아이디/비밀번호 찾기

05.13. (일) 영어회화 > 4/5 < >

아, 찾았어요! 여깁어요!
 Oh, I found it! Here it is!
 오, 아이 파운드 잇! 히어 잇 이즈!

5월엔 말로 주문하고
 최대 1만원
 네 ~ 주문했어요

```
<html lang="ko" class="svgless">
<head>
<meta charset="utf-8">
<meta name="Referrer" content="origin">
<meta http-equiv="Content-Script-Type" content="text/javascript">
<meta http-equiv="Content-Style-Type" content="text/css">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=1100">
<meta name="apple-mobile-web-app-title" content="NAVER" />
<meta name="robots" content="index,nofollow"/>
<meta name="description" content="네이버 메인에서 다양한 정보와 유용한 콘텐츠를 만나 보세요"/>
<meta property="og:title" content="네이버">
<meta property="og:url" content="http://www.naver.com/">
<meta property="og:image" content="https://s.pstatic.net/static/www/mobile/edit/2016/0705/mobile_212852414260.png">
<meta property="og:description" content="네이버 메인에서 다양한 정보와 유용한 콘텐츠를 만나 보세요"/>
<meta name="twitter:card" content="summary">
<meta name="twitter:title" content="">
<meta name="twitter:url" content="http://www.naver.com/">
```




Nokogiri gem

1.css selector 사용

```
puts "### Search for nodes by css"
doc.css('nav ul.menu li a', 'article h2').each do |link|
  puts link.content
end
```

2.Xpath 사용

```
puts "### Search for nodes by xpath"
doc.xpath('//nav//ul//li/a', '//article//h2').each do |link|
  puts link.content
end
```

3.섞어서 둘 다 사용

```
puts "### Or mix and match."
doc.search('nav ul.menu li a', '//article//h2').each do |link|
  puts link.content
end
```

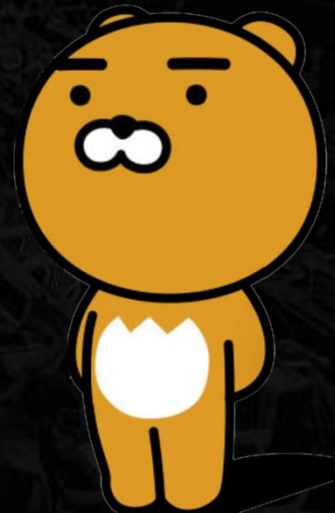
오늘은 세가지 방법 중 css selector 사용!

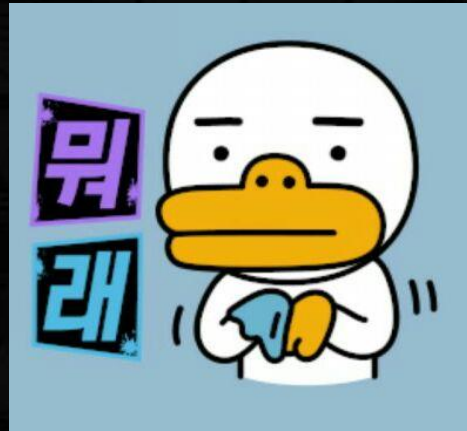
Css Selector?

내가 정보를 빼 오려는 block을 찾도록 도와주는 역할

```
h1.special
```

special 클래스를 가진
요소들 중에서,
h1 태그들이 와봐~





무슨 말인지 잘 모르겠으니
더 자세히 알아보러 가 봅시다!

1단계 : gem 설치

<Gemfile>

```
[M] /README.md  x  Gemfile  x
1  source 'https://rubygems.org'
2
3  gem 'nokogiri'
4
5  # Bundle edge Rails instead: gem 'rails'
6  gem 'rails', '4.2.5'
7  # Use sqlite3 as the database for Active
8  gem 'sqlite3'
9  # Use SCSS for stylesheets
10 gem 'sass-rails', '~> 5.0'
11 # Use Uglifier as compressor for JavaScr
```

Gemfile에
gem 'nokogiri' 입력 후

Bash창에
Bundle install

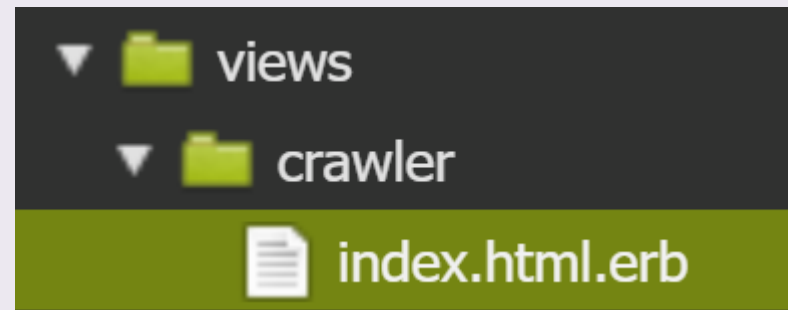
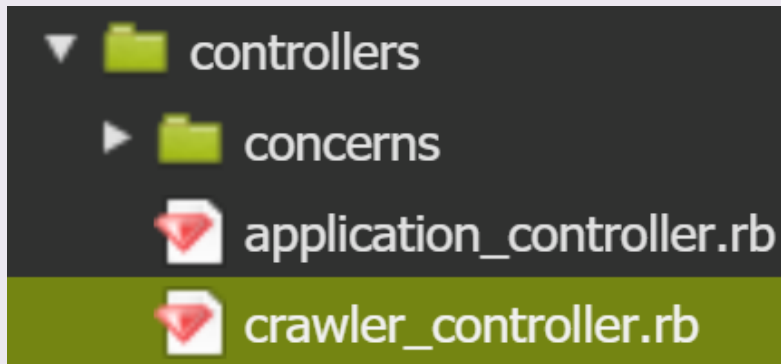
```
tjdud0123:~/workspace $ bundle install
```

```
Bundle complete!
```

2단계 : controller 생성(with index view파일)

```
~/workspace $ rails g controller crawler index
```

Crawler라는 이름의 컨트롤러 와 index라는 이름의 view파일 생성



3단계 : 라우팅, index페이지 홈으로 설정

<config/routes.rb>

```
rails application.routes.draw do
  root 'crawler#index'
  get 'crawler/index'
```

4단계 : crawler 컨트롤러작성 - nokogiri 쓸 준비

<crawler_controller.rb>

```
require 'nokogiri' #굳이 안써도 되긴 함
require 'open-uri'

class CrawlerController < ApplicationController
  def index
  end
end
```

`require 'nokogiri'` //nokogiri를 쓸 거야!
`require 'open-uri'` //open-uri를 쓸 거야!

Cf. [OpenURI](#) is an easy-to-use wrapper for `Net::HTTP`, `Net::HTTPS` and `Net::FTP`.

5단계 : 크롤링 하고 싶은 페이지 찾고 url복사!



NAVER MUSIC

유지 홈 마이 뮤직 오늘의 뮤직 뮤지션 리그 온스테이지 이용권 구매 New HD 음원 라디오 NEW 네이버뮤

음악플레이어

음악감상

- TOP 100
 - 종합
 - 국내
 - 해외
 - 뮤지션 리그
 - 차트 히스토리
- 최신앨범
- 최신곡
- 국내음악
- 해외음악
- 기타장르
- 뮤직비디오

추천음악

- 테마채널 NEW
- 명예의 전당
- 블로그 DJ
- Musician's Choice
- 지구촌 팝뉴스

TOP 100 | 종합

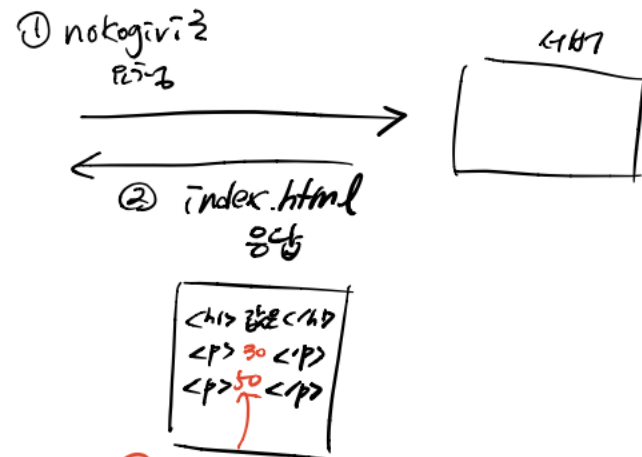
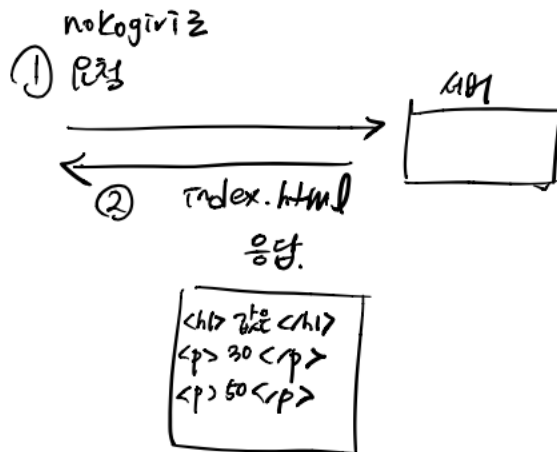
▶ 전체듣기 ▶ 들기 + 재생목록에 추가 다운로드 음악리스트 선물 내 리스트에 담기 보내기

| 순위 | 곡명 |
|------|---------------------------------|
| 1 -0 | 밤 (Time for the moon night) |
| 2 -0 | 주지마 밤 (Time for the moon night) |
| 3 -0 | What is Love? |
| 4 -0 | 사랑을 했다 (LOVE SCENARIO) |
| 5 -0 | You |
| 6 -0 | 뽕뽕 |
| 7 -0 | 소나기 (Feat. 10cm) |
| 8 +1 | 별이 빛나는 밤 |

music.naver.com/listen/top100.nhn?domain=TOTAL

NAVER MUSIC

크롤링 할 페이지 찾을 때 주의!

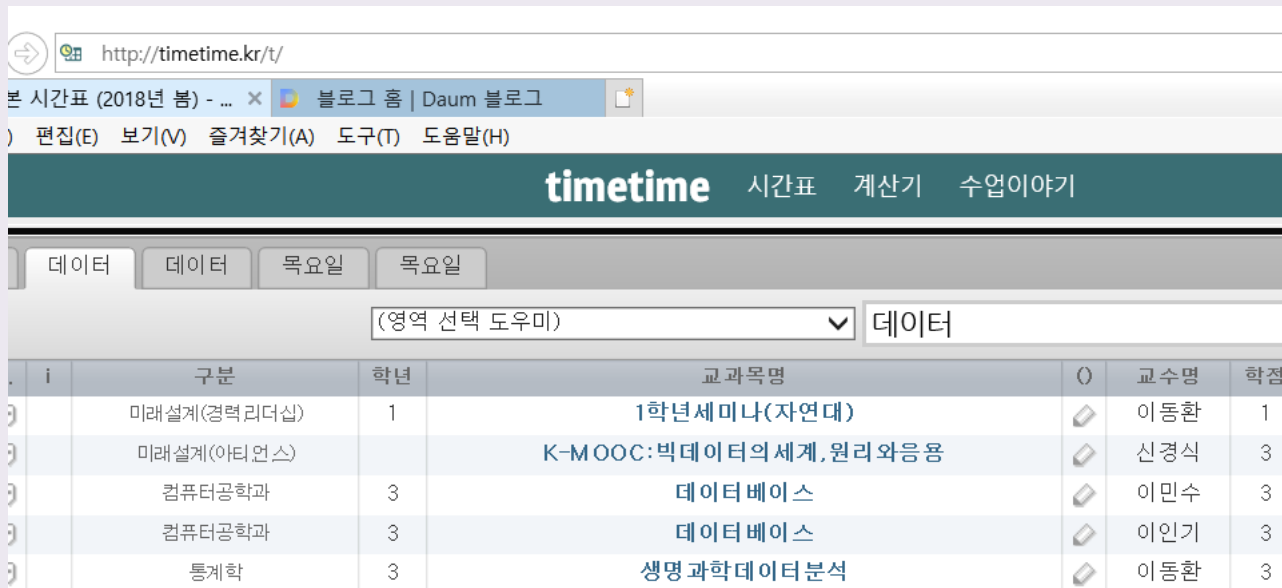


③ 데이터 JS로 채워넣기
∴ nokogiri로 가져왔을 때,
index.html에는 값이 바뀌었다

크롤링 할 페이지 찾을 때 주의!

서버에 요청해서 html문서를 받아 올 때,
html껍데기만 존재하고 그 안의 요소들이
Js등으로 작성되어 있다면 값이 비어있어서
데이터를 뽑아 올 수 없음!

크롤링 할 페이지 찾을 때 주의!



The screenshot shows a web browser window with the URL <http://timetime.kr/t/>. The page has a header with the 'timetime' logo and navigation links: '시간표' (Timetable), '계산기' (Calculator), and '수업이야기' (Class Story). Below the header, there are tabs for '데이터' (Data) and '목요일' (Friday). A dropdown menu is set to '(영역 선택 도우미)' (Help select area) and the '데이터' (Data) tab is selected. The main content is a table with the following data:

| i | 구분 | 학년 | 교과목명 | | 교수명 | 학점 |
|---|-------------|----|----------------------|--|-----|----|
| | 미래설계(경력리더십) | 1 | 1학년세미나(자연대) | | 이동환 | 1 |
| | 미래설계(아티언스) | | K-MOOC:빅데이터의세계,원리와응용 | | 신경식 | 3 |
| | 컴퓨터공학과 | 3 | 데이터베이스 | | 이민수 | 3 |
| | 컴퓨터공학과 | 3 | 데이터베이스 | | 이인기 | 3 |
| | 통계학 | 3 | 생명과학데이터분석 | | 이동환 | 3 |

Url주소가 변경되지 않는다면

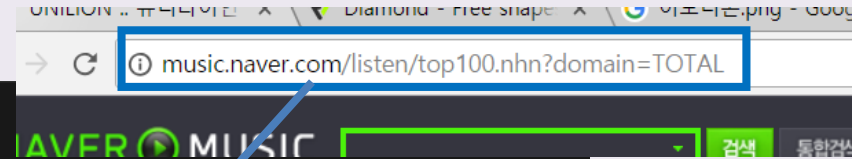
데이터를 뽑아 올 수 없음!

6단계 : crawler 컨트롤러작성 - HTML문서 가져오기

<crawler_controller.rb>

```
require 'nokogiri' #굳이 안써도 되긴 함
require 'open-uri'

class CrawlerController < ApplicationController
  def index
    url = "http://music.naver.com/listen/top100.nhn?domain=TOTAL"
    doc = Nokogiri::HTML(open(url))
  end
end
```



Nokogiri 를 이용해서 HTML파일을 읽을 것.
그 HTML파일은 변수 안에 담긴 url주소로 **오픈** 한 파일임.
그 파일을 doc변수에 저장!

콘솔에 컨트롤러에 작성한 코드 입력해보기!

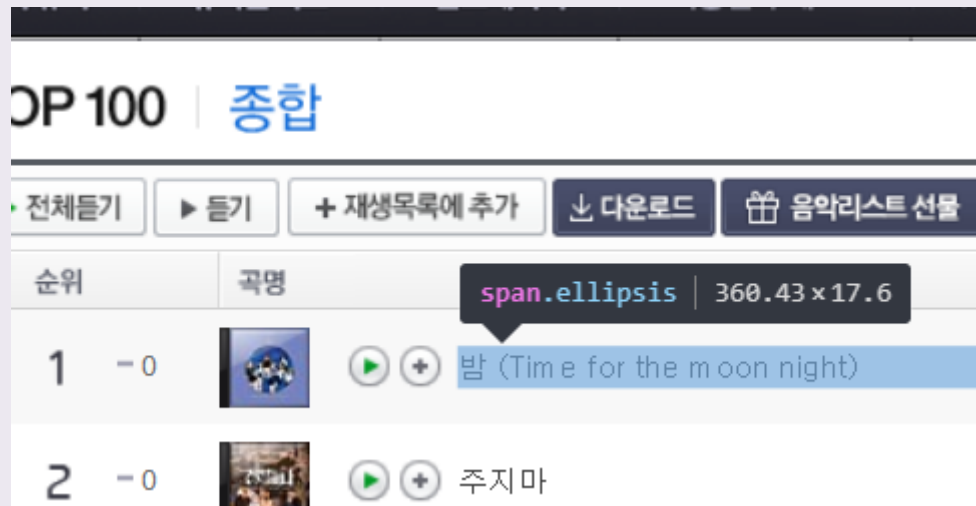
2018.5.17

7단계 : crawler 컨트롤러작성 - 원하는 데이터 추출

<crawler_controller.rb>

```
url = "http://music.naver.com/listen/top100.nhn?domain=TOTAL"  
doc = Nokogiri::HTML(open(url))  
music = doc.css('span.ellipsis')
```

개발자도구 켜고 원하는 데이터
Css selector 확인!

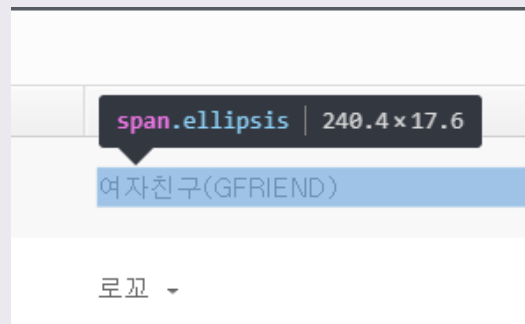


콘솔로 확인해보면서 합시다!

```
music = doc.css('span.ellipsis')
```

```
=> [#<Nokogiri::XML::Element:0x1d41ec0 name="span" attributes=[#<Nokogiri::XML::Attr:0x1d3ff58 name="class" value="ellipsis">] children=[#<Nokogiri::XML::Text:0x1d3f580 "밤 (Time for the moon night)">]>, #<Nokogiri::XML::Element:0x1d3de38 name="span" attributes=[#<Nokogiri::XML::Attr:0x1d3ddd4 name="class" value="ellipsis">] children=[#<Nokogiri::XML::Text:0x1d3d8e8 "\r\n\t\t\t\r\n\t\t\t\r\n\t\t\t여자친구(GFRIEND)\r\n\t\t\t">]>, #<Nokogiri::XML::Element:0x1cef97c name="span" attributes=[#<Nokogiri::XML::Attr:0x1cef864 name="class" value="ellipsis">] children=[#<Nokogiri::XML::Text:0x1cef134 "주지마">]>, #<Nokogiri::XML::Element:0x1c9f24c name="span" attributes=[#<Nokogiri::XML::Attr:0x1c9f1e8 name="class" value="ellipsis">] children=[#<Nokogiri::XML::Text:0x1c9ecfc "What is Love?">]>, #<Nokogiri::XML::Element:0x1c97
```

※동일한 css selector를 가지고 있다면 원하지 않는 것까지 모두 긁어옴.



콘솔로 확인해보면서 합시다!

```
music = doc.css('span.ellipsis')
```

```
2.3.4 :005 > music.count  
=> 96
```

변수이름.count -> 개수

곡 제목만 필요했는데...
어떻게 해야 하지..?

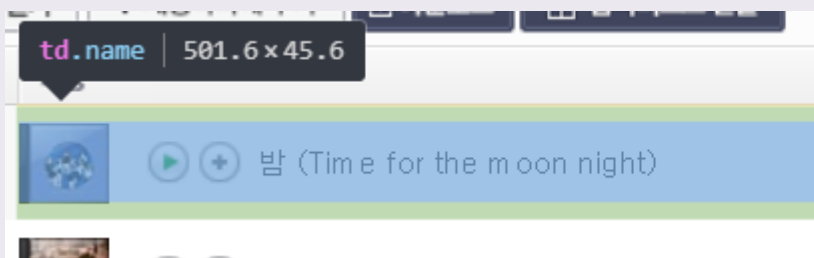


콘솔로 확인해보면서 합시다!

```
music = doc.css('span.ellipsis')
```

```
2.3.4 :005 > music.count  
=> 96
```

변수이름.count -> 개수



td.name 자식요소로 들어가있음.

```
<td class="name"> == $0  
▶ <a href="/album/index.nhn?albumId=2450804" class="thumb pht36  
NPI=a:image,r:1,i:2450804">...</a>  
▶ <a href="#" class="_play_ico ico_listen  
NPI=a:play,r:1,i:21293002">...</a>  
▶ <a href="#" class="_add_ico ico_add  
NPI=a:plus,r:1,i:21293002">...</a>  
▼ <a href="#21293002" class="_title title  
NPI=a:track,r:1,i:21293002" title="밤  
(Time for the moon night)">  
  <span class="ellipsis">밤 (Time for  
the moon night)</span>
```



But, 가수이름은 그렇지 않음. 그렇다면?



Tip! - 헛갈린다!

Chaining Selector

VS

Nested Element

```
h1.special {  
  }  
}
```

```
<h1 class="special">
```

Chaining Selector

special 클래스를 가진
요소들 중에서 h1 태그를 선택

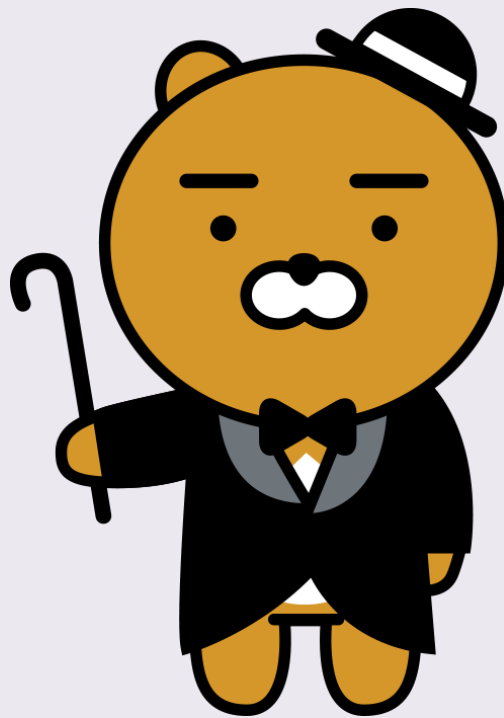
```
.main-list li {  
  }  
}
```

※띄어쓰기

```
<ul class='main-list'>  
  <li> ... </li>  
  <li> ... </li>  
  <li> ... </li>  
</ul>
```

Nested Element

main-list 클래스 안에 있는 요소
(자식요소) 중에 li 태그를 선택



결론 : 더 구체적으로 골라내자!

7단계 : crawler 컨트롤러작성 - 원하는 데이터 추출

<crawler_controller.rb>

```
class CrawlerController < ApplicationController
  def index
    url = "http://music.naver.com/listen/top100.nhn?domain=TOTAL"
    doc = Nokogiri::HTML(open(url))
    music = doc.css('td.name span.ellipsis')
  end
end
```



```
music = doc.css('span.ellipsis')
```



```
music = doc.css('td.name span.ellipsis')
```

```
=> [#<Nokogiri::XML::Element:0x1d41ec0 name="span" attributes=[#<Nokogiri::XML::Attr:0x1d41e9c value="ellipsis">] children=[#<Nokogiri::XML::Text:0x1d3f580 "밤 (Time for the moon night)">], #<Nokogiri::XML::Element:0x1d41ec0 name="span" attributes=[#<Nokogiri::XML::Attr:0x1cef864 name="class" value="ellipsis">] children=[#<Nokogiri::XML::Text:0x1cef134 "주지마">], #<Nokogiri::XML::Element:0x1c9f24c name="span" attributes=[#<Nokogiri::XML::Attr:0x1cecfcc name="class" value="ellipsis">] children=[#<Nokogiri::XML::Text:0x1c9ecfc "What is Love?">], #<Nokogiri::XML::Element:0x1c3b350 name="span" attributes=[#<Nokogiri::XML::Attr:0x1c3b2c4 name="class" value="ellipsis">] children=[#<Nokogiri::XML::Text:0x1c3b2c4 "What is Love?">]]
```

```
2.3.4 :007 > music.count
=> 50
```



콘솔로 확인해보면서 합시다!

※태그자체를 가져오기 때문에 가져오고 싶었던 데이터 값만 뜨지 않음.

```
=> [#<Nokogiri::XML::Element:0xd41ec0 name="span" attributes=[#<Nokogiri::XML::Attr:0xd1d  
lipsis">] children=[#<Nokogiri::XML::Text:0xd3f580 "밤 (Time for the moon night)">], #<N  
97c name="span" attributes=[#<Nokogiri::XML::Attr:0x1cef864 name="class" value="ellipsis">  
:Text:0x1cef134 "주지마">], #<Nokogiri::XML::Element:0x1c9f24c name="span" attributes=[#<N  
8 name="class" value="ellipsis">] children=[#<Nokogiri::XML::Text:0x1c9ecfc "What is Love?  
nt:0x1c3b350 name="span" attributes=[#<Nokogiri::XML::Attr:0x1c3b2c4 name="class" value="e
```


```
2.3.4 :008 > music[0].text  
=> "밤 (Time for the moon night)"
```

.text ->안에 text요소만 가져옴.

7단계 : crawler 컨트롤러작성 - 원하는 데이터 추출

<crawler_controller.rb>

```
def index
  url = "http://music.naver.com/listen/top100.nhn?d"
  doc = Nokogiri::HTML(open(url))
  music = doc.css('td.name span.ellipsis')
  music_text = music.map{|mus| mus.text}
end
```



```
arr.map { |a| 2*a }  #=> [2, 4, 6, 8, 10]
arr                #=> [1, 2, 3, 4, 5]
```

.map method

music array에 있는 요소 하나하나를 **.text**로 텍스트만 뽑아서
.map을 이용해 music_text라는 array에 새롭게 저장

8단계 : view파일에 출력해보기

<crawler_controller.rb>

```
@music_text = music.map{|mus| mus.text}
```

인스턴스 변수로

<index.html.erb>

```
<h1>네이버 뮤직 TOP 100</h1>

<%@music_text.each_with_index do |music,index|%>
<p><%= (index+1) %>. <%= music %></p>
<%end%>
```

.each_with_index -> 각 요소와 함께 index도 사용가능

Index사용없이 찍고 싶다면 그냥 each do 사용해서 출력!

크롤링된 데이터 출력화면입니다

1. 밤 (Time for the moon night)
2. 주지마
3. You
4. What is Love?
5. 별, 그대 (The Only Star)
6. 소나기 (Feat. 10cm)
7. 붕붕 (Feat. 식케이) (Prod. GroovyRoom)
8. 지나오다
9. 사랑을 했다 (LOVE SCENARIO)
10. EVERYDAY
11. 별이 빛나는 밤
12. 그때 헤어지면 돼
13. 너에게 (To You)
14. 뽀뽀
15. 그날처럼
16. 그날의 너



옆 사람이랑 같이 어떤 데이터 크롤링 해볼지 상의해보고

1. 네이버 뮤직
2. 좋아하는 네이버 웹툰의 에피소드 제목들
3. 네이버 영화에서 현재 상영작 제목들
4. 네이버 TV Top 100 프로그램 제목들
ex)뮤직뱅크...



Crawling 실습시작!!

Workspace name

crawling0517

Description

Make a short description of your workspace

[Hosted workspace](#)
[Clone workspace](#)
[Remote SSH workspace](#)
[Salesforce](#)

☐
Private
 This is a workspace for your eyes only

☐
Public
 This will create a workspace for everybody to see

Clone from Git or Mercurial URL (optional)

e.g. ajaxorg/ace or git@github.com:ajaxorg/ace.git

Choose a template

HTML5

Node.js

PHP, Apache & ...

Python

Django

Ruby

C++

Wordpress

Rails Tutorial

Blank

Harvard's CS50

Create workspace

EWHAxSOGANG LIKELION

2018.5.17

Crawling 8단계!

1단계 : gem 설치

2단계 : controller 생성(with index view파일)

3단계 : 라우팅, index페이지 홈으로 설정

4단계 : crawler 컨트롤러작성 - nokogiri 쓸 준비

5단계 : 크롤링 하고 싶은 페이지 찾고 url복사!

6단계 : crawler 컨트롤러작성 - HTML문서 가져오기

7단계 : crawler 컨트롤러작성 - 원하는 데이터 추출

8단계 : view파일에 출력해보기

모두
수고하셨습니다!

