

[네이버 Open API Key 를 발급받는 새로운 방법]

(2016-04-01 기준입니다)

새로운 네이버 개발자 센터 OPEN API를 이용하는 방법입니다.

키 발급 및 기존의 API에서 변경된 사항들에 대한 간략한 설명과 변경된 코드를 제공해 드리도록 하겠습니다.

1. 네이버 개발자 센터로 접속합니다. (<https://developers.naver.com/main>)



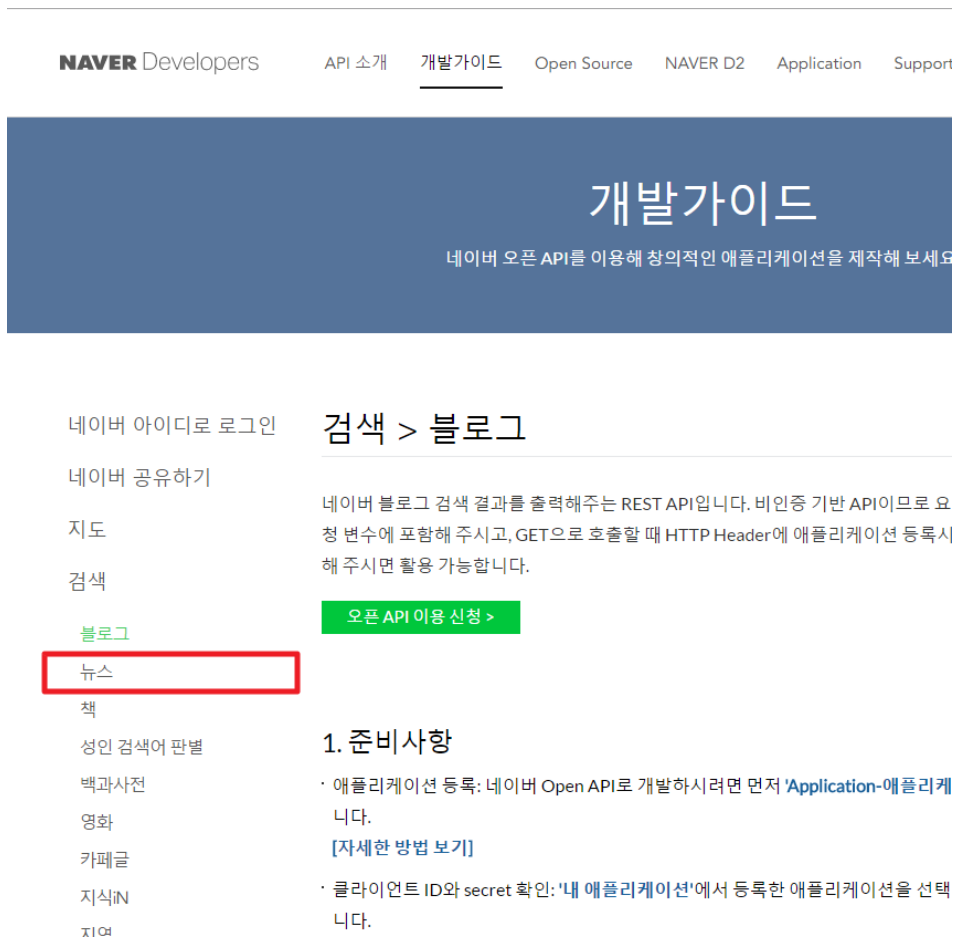
2. 로그인을 합니다.



3. 개발가이드 -> 검색 메뉴로 이동합니다.

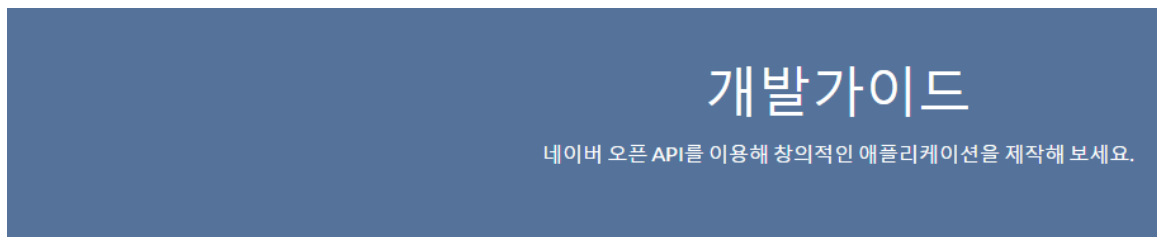


4. 3번에서 검색 메뉴로 이동하신 후, 좌측의 메뉴 리스트에서 뉴스 항목으로 이동합니다.



왕초보! 파이썬 배워 크롤러 DIY 하다! - 서진수,최경우 공저 - 도서출판 더알음

5. '오픈 API 이용 신청' 버튼을 클릭합니다. 애플리케이션 등록 화면으로 이동하시면 등록하기 위해 입력해야하는 여러가지 항목들이 있습니다. 각각의 항목들을 알맞게 채워줍니다.



네이버 아이디로 로그인

네이버 공유하기

지도

검색

블로그

뉴스

책

성인 검색어 판별

백과사전

영화

카페글

지식iN

타임

검색 > 뉴스

네이버 뉴스 검색 결과를 출력해주는 REST API입니다. 비인증 기반 API이므로 요청 URL 변수에 포함해 주시고, GET으로 호출할 때 HTTP Header에 애플리케이션 등록시 발급된 키를 주시면 활용 가능합니다.

[오픈 API 이용 신청 >](#)

좌측의 리스트에서 뉴스를
선택하신 후
오픈 API 이용 신청 버튼을
클릭합니다.

1. 준비사항

- 애플리케이션 등록: 네이버 Open API로 개발하시려면 먼저 **'Application-애플리케이션'**을 등록합니다.
[\[자세한 방법 보기\]](#)
- 클라이언트 ID와 secret 확인: **'내 애플리케이션'**에서 등록된 애플리케이션을 선택하면 확인할 수 있습니다.

위 항목을 모두 입력한 화면들입니다.

애플리케이션 등록 (API 이용신청)

애플리케이션의 기본 정보를 등록하면, 좌측 메뉴에 그 애플리케이션 이름으로 서브 메뉴가 만들어집니다. Open API를 이용하시려면 애플리케이션 이름을 클릭해서 나오는 'API 권한관리' 탭메뉴에서 각 API들을 선택하시면 됩니다.

애플리케이션 이름	<input type="text" value="Naver New API"/> 애플리케이션 이름은 자유롭게 생성해 주시면 됩니다.
카테고리	<input type="text" value="기타"/> 애플리케이션의 카테고리를 지정해 줍니다.
이용 목적 ?	<input type="checkbox"/> 로그인 오픈API (네이버 아이디로 로그인) ? <input checked="" type="checkbox"/> 비로그인 오픈API ?
비로그인 오픈API 서비스 환경	<div><input checked="" type="checkbox"/> 오픈API 이용 권한 신청 ? <input type="checkbox"/> 지도 <input type="checkbox"/> 음성인식 <input checked="" type="checkbox"/> 검색 <input type="checkbox"/> 음성합성 <input type="checkbox"/> 기계번역 <input type="checkbox"/> 캡차 <input type="checkbox"/> 단축URL</div> <div><p>웹 서비스 URL</p><p>애플리케이션의 서비스 환경을 알기 위한 항목입니다.</p><input type="text" value="http://concierge-pr.com/"/></div> <div><p>안드로이드 앱 Package명</p><input type="text" value="안드로이드 앱의 Package명을 입력하세요 : 예)com.nhn.android.sampleapp"/></div> <div><p>iOS Bundle ID</p><input type="text"/></div>

	<div> <div> <div>iOS Bundle ID</div> <div> <div>iOS 앱의 Bundle ID를 입력하세요 : 예)com.nhncorp.NaverMap</div> </div> </div> <div> <div>최소 1개 이상의 사용환경 정보를 입력해주세요.</div> </div> </div> <div> <div> <div>1 사용환경에 따른 필요정보 입력에 대한 안내</div> <div>▼</div> </div> </div>
담당자 연락처	<div> <div> <div>휴대폰 +82 10-4***-****8</div> <div>수정</div> </div> <div> <div>이메일 wo****@n*****.com</div> <div>수정</div> </div> </div> <div> <div>현재 로그인된 계정의 정보가 기본적으로 저장되어 있습니다.</div> <div>연락처 인증은 최초 애플리케이션 등록시 1회만 하며, 휴대폰 인증만 하시면 됩니다.</div> </div>
이용약관	<div> <div> <div>2.1.4. "애플리케이션 등"이라 함은 "API 서비스"가 적용된 "이용자"의 애플리케이션 또는 웹사이트를 말합니다.</div> <div>2.1.5. "인증 Key"라 함은 관련 시스템이 "API 서비스" 이용 승낙을 받은 "이용자"임을 식별할 수 있도록 "회사"가 "이용자" 별로 할당하는 고유한 값을 말합니다.</div> <div>2.1.6. "네이버 로그인 서비스"라 함은 "이용자"의 고객이 본인의 네이버 아이디와 비밀번호를 입력하여 "이용자"의 "애플리케이션 등"에 로그인할 수 있고, 또한 네이버 서비스에서 사용하는 별명, 성별, 프로필 사진 등을 "이용자"의 "애플리케이션 등"에서 사용할 수 있도록 "회사"가 제공하는 제반 서비스를 말합니다. 다만, "약관" 규정상 "이용자"의 고객이 본인의 네이버 아이디와 비밀번호를 입력하여 "이</div> </div> <div> <div><input checked="" type="checkbox"/> 해당 약관의 내용을 숙지하였으며, 이에 동의합니다.</div> <div><input checked="" type="checkbox"/> 유해, 불법적인 내용과 관련되거나, 신청하지 않은 애플리케이션 등에 무단으로 사용하지 않겠습니다.</div> </div> </div>

등록하기

취소

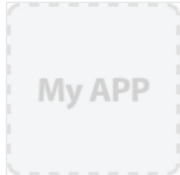
위의 항목을 빠짐없이 기재하셨다면

등록하기 버튼을 클릭합니다.

애플리케이션 등록을 완료하시면 아래와 같은 화면이 나타납니다.

Naver New API

개요	설정	API 권한관리	멤버관리	로그인 통계
----	----	----------	------	--------



Naver New API

개발중 ?

Client ID

6lCHJBh0LOJQoYoRJ1cx

Client Secret

amGLdLV1vd

재발급

OpenAPI 통계

어제의 로그인 사용자수	0
최근 7일간 로그인 사용자수	0

저희가 OPEN API를 사용하는데 필요한 정보는 Client ID와 Client Secret입니다. 이 정보들을 HTTP 헤더에 삽입하여 요청을 하여야 합니다.

새로운 API의 가이드 입니다.

1. 요청 URL

기존 API	새로운 API
http://openapi.naver.com/search	https://openapi.naver.com/v1/search/news.xml

2. 요청 변수

기존 API	새로운 API
key, target, query, display, start, sort	query, display, start, sort

아래의 표는 새로운 API에서 사용하게 되는 요청 변수의 설명입니다. 기존 API에서 key와 target 변수만 사라졌을 뿐 기본값 및 설명은 동일합니다.

요청변수	타입	필수여부	기본값	설명
query	string	Y	-	검색을 원하는 문자열로서 UTF-8 로 인코딩한다.
display	integer	N	10(기본값),100(최대)	검색 결과 출력 건수 지정
start	integer	N	1(기본값), 1000(최대)	검색 시작 위치로 최대 1000 까지 가능
sort	string	N	sim(기본값), date	정렬 옵션: sim (유사도순), date (날짜순)

3. 출력 결과 - 출력 결과에서 나오는 모든 태그는 동일하게 출력 되므로 어떠한 태그들이 나오는지 정도만 표로 정리해두도록 하겠습니다.

필드	타입	설명
rss	-	디버그를 쉽게 하고 RSS 리더기만으로 이용할 수 있게 하기 위해 만든 RSS 포맷의 컨테이너이며 그 외의 특별한 의미는 없다.

필드	타입	설명
channel	-	검색 결과를 포함하는 컨테이너이다. 이 안에 있는 title, link, description 등의 항목은 참고용으로 무시해도 무방하다.
lastBuildDate	datetime	검색 결과를 생성한 시간이다.
total	integer	검색 결과 문서의 총 개수를 의미한다.
start	integer	검색 결과 문서 중, 문서의 시작점을 의미한다.
display	integer	검색된 검색 결과의 개수이다.
item	-	개별 검색 결과이며 title, originallink, link, description, pubDate 를 포함한다.
title	string	개별 검색 결과이며, title, originallink, link, description, pubDate 를 포함한다.
originallink	string	검색 결과 문서의 제공 언론사 하이퍼텍스트 link 를 나타낸다.
link	string	검색 결과 문서의 제공 네이버 하이퍼텍스트 link 를 나타낸다.
description	string	검색 결과 문서의 내용을 요약한 패시지 정보이다. 문서 전체의 내용은 link 를 따라가면, 읽을 수 있다. 패시지에서 검색어와 일치하는 부분은 태그로 감싸져 있다.
pubDate	datetime	검색 결과 문서가 네이버에 제공된 시간이다.

4. 에러 코드 - 에러 코드는 기존 API에서 메시지 형태로만 출력되었지만, 새로운 API에서는 조금 더 구체화 되어지고 명확하게 명시되어 있습니다. 어떤 에러코드들이 존재하는지 표로 정리해 두었습니다.

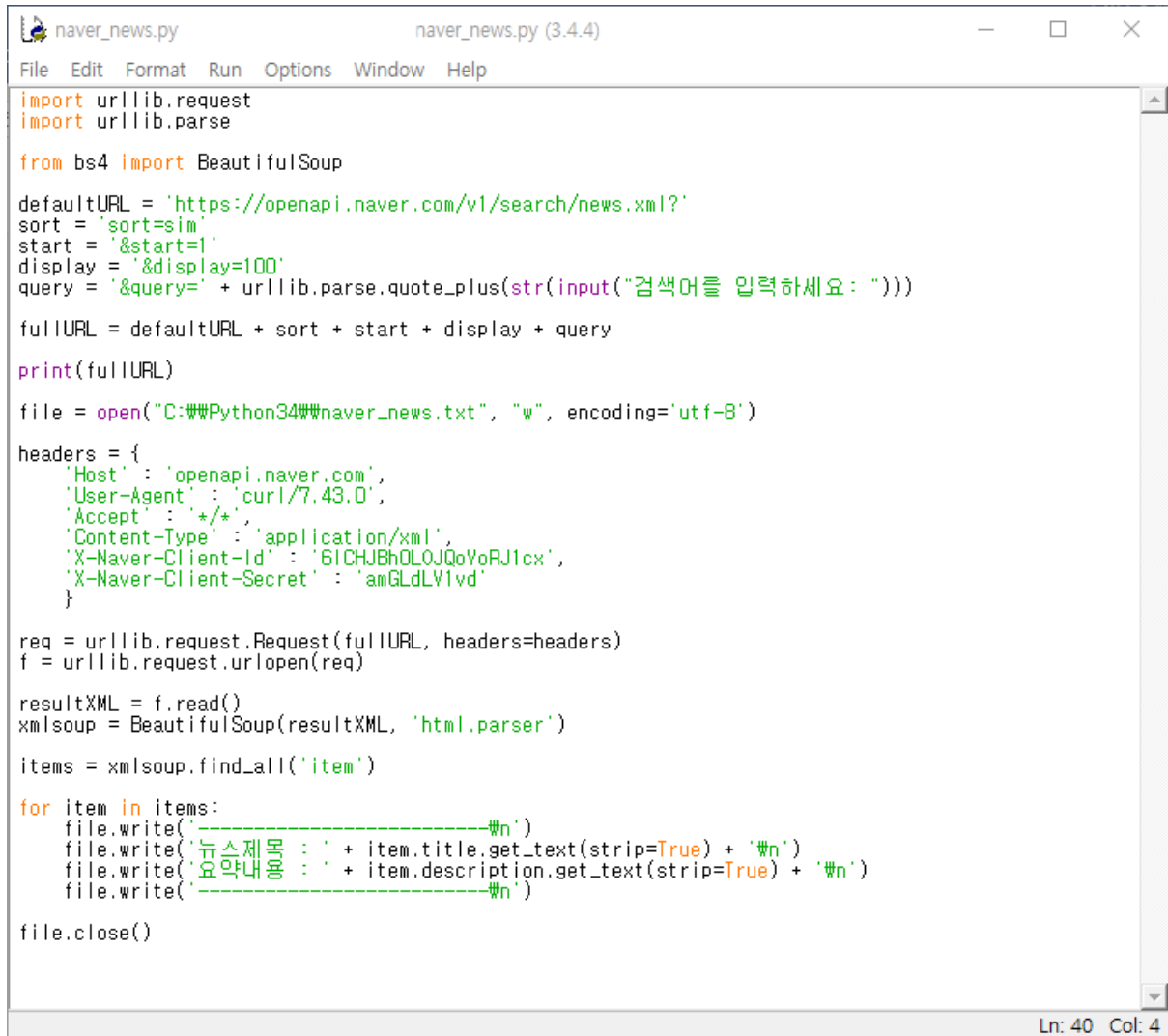
HTTP 코드	에러코드	에러 메시지	조치방안
400	SE01	Incorrect query request (잘못된 쿼리요청입니다.)	검색 API 요청에 오류가 있습니다. 요청 URL, 필수 요청 변수가 정확한지 확인 바랍니다.

HTTP 코드	에러코드	에러 메시지	조치방안
400	SE02	Invalid display value (부적절한 display 값입니다.)	display 요청 변수값이 허용범위(1~100)인지 확인해 보세요.
400	SE03	Invalid start value (부적절한 start 값입니다.)	start 요청 변수값이 허용범위(1~1000)인지 확인해 보세요.
400	SE04	Invalid sort value (부적절한 sort 값입니다.)	sort 요청 변수 값에 오타가 없는지 확인해 보세요.
404	SE05	Invalid search api (존재하지 않는 검색 api 입니다.)	검색 API 대상에 오타가 없는지 확인해 보세요.
500	SE99	System Error (시스템 에러)	서버 내부 에러가 발생하였습니다. 포럼에 올려주시면 신속히 조치하겠습니다.

여기까지 새로운 Open API Key 를 발급 받았습니다.

이제 이 Key 를 활용하는 방법을 설명 드리겠습니다.

[새로운 Open API Key 를 활용한 소스코드]



```
naver_news.py naver_news.py (3.4.4)
File Edit Format Run Options Window Help

import urllib.request
import urllib.parse

from bs4 import BeautifulSoup

defaultURL = 'https://openapi.naver.com/v1/search/news.xml?'
sort = 'sort=sim'
start = '&start='
display = '&display=100'
query = '&query=' + urllib.parse.quote_plus(str(input("검색어를 입력하세요: ")))

fullURL = defaultURL + sort + start + display + query

print(fullURL)

file = open("C:\\Python34\\naver_news.txt", "w", encoding='utf-8')

headers = {
    'Host' : 'openapi.naver.com',
    'User-Agent' : 'curl/7.43.0',
    'Accept' : '*/.*',
    'Content-Type' : 'application/xml',
    'X-Naver-Client-Id' : '6lCHJBhOLOJQoYoRJ1cx',
    'X-Naver-Client-Secret' : 'amGLdLV1vd'
}

req = urllib.request.Request(fullURL, headers=headers)
f = urllib.request.urlopen(req)

resultXML = f.read()
xmlsoup = BeautifulSoup(resultXML, 'html.parser')

items = xmlsoup.find_all('item')

for item in items:
    file.write('-----#n')
    file.write('뉴스제목 : ' + item.title.get_text(strip=True) + '#n')
    file.write('요약내용 : ' + item.description.get_text(strip=True) + '#n')
    file.write('-----#n')

file.close()
```

Ln: 40 Col: 4

코드를 파헤치기 전에 HTTP 헤더에 대해서 모르시는 분들을 위하여 간단하게 설명을 해드리도록 하겠습니다.

HTTP에서 사용하는 헤더란, 서버에게 지금 요청을 하는 사용자에 대한 정보가 담겨 있는 데이터들입니다. 예를 들어서 입국 심사처럼 나의 국적은 어디인지, 어느 나라를 통해서 왔는지, 이 나라에 무슨 목적으로 왔는지 등 이런 모든 정보가 기재되어있는 하나의 종이라고 생각하시면 됩니다. 이와 같이 헤더를 서버에 전달하여 브라우저의 종류는 무엇인지, 어떤 유형의 데이터를 받는 지, 어디에서 페이지에 접속하게 되었는지 등을 서버에 전달하는 역할을 합니다.

그렇기 때문에 입국 심사에서 통과하지 못했을 때 입국을 거절 당하는 것처럼, 서버에서 요구하는 특정 헤더의 정보가 잘못 되었거나 존재하지 않을 경우에는 서버쪽에서 사용자의 데이터 요청을 거절할 수 있습니다. 하지만 헤더의 정보를 따로 전달해주지 않았을 경우에는 어떻게 될까요?

예를 들어서 입국 심사의 절차에서 아무것도 묻지도 따지지도 않는다고 하면 입국이 되지 않을 이유는 없겠죠? 이와 같은 경우입니다. 서버 쪽에서도 반드시 필요로 하는 헤더의 정보 같은 필수 헤더 정보가 따로 존재하지 않다면 사용자의 요청에 언제나 응답해주는 서버가 되는 것입니다.

현재 네이버에서 요구하는 헤더의 정보는 애플리케이션을 등록하고 발급받은 Client id와 Client secret 정보가 됩니다. 이 정보가 누락되었거나 잘못된 정보가 전달 되었을 경우에는 네이버 쪽 서버에서 데이터 요청을 거절 또는 거부할 수 있습니다.

```
#1. import urllib.request
#2. import urllib.parse
#3. from bs4 import BeautifulSoup
#4. defaultURL = 'https://openapi.naver.com/v1/search/news.xml?'
#5. sort = 'sort=sim'
#6. start = '&start=1'
#7. display = '&display=100'
#8. query = '&query=' + urllib.parse.quote_plus(str(input("검색어를 입력하세요: ")))
#9. fullURL = defaultURL + sort + start + display + query
#10. print(fullURL)
#11. file = open("C:\\Python34\\naver_news.txt", "w", encoding='utf-8')
#12. headers = {
    'Host' : 'openapi.naver.com',
    'User-Agent' : 'curl/7.43.0',
    'Accept' : '*/*',
    'Content-Type' : 'application/xml',
    'X-Naver-Client-Id' : '6lCHJBh0LOJQoYoRJ1cx',
    'X-Naver-Client-Secret' : 'amGLdLV1vd'
}
#13. req = urllib.request.Request(fullURL, headers=headers)
#14. f = urllib.request.urlopen(req)
#15. resultXML = f.read()
#16. xmlsoup = BeautifulSoup(resultXML, 'html.parser')
#17. items = xmlsoup.find_all('item')
#18. for item in items:
    file.write('-----\n')
    file.write('뉴스제목 : ' + item.title.get_text(strip=True) + '\n')
    file.write('요약내용 : ' + item.description.get_text(strip=True) + '\n')
    file.write('-----\n')
#19. file.close()
```

[소스코드 설명].

#1, #2, #3 - HTTP 요청을 하는데 필요한 모듈을 import 합니다.

#4 - 새로운 API에서 사용하는 요청 URL을 변수에 담아둡니다.

#5, #6, #7 - 새로운 API에서 사용하는 요청 변수를 각각의 변수에 담아둡니다.

#8 - 사용자에게 검색어를 입력받아 quote_plus함수로 UTF-8타입에 맞도록 변환시켜 줍니다.

#9 - 요청 URL을 요청 변수들과 함께 조합하여 완성된 URL을 만들어 변수에 저장합니다.

#10 - 요청 URL이 제대로 조합되었는지 확인하기 위해서 출력합니다.

#11 - API를 통해서 얻어온 데이터를 저장할 텍스트 파일을 지정된 경로에 생성합니다.

#12(중요) - HTTP 요청을 할 때 서버에 애플리케이션 정보에 대한 정보를 넘겨주기 위한 변수입니다. 이곳에서 가장 중요한 부분은 X-Naver-Client-Id와 X-Naver-Client-Secret입니다. 각각의 변수에는 애플리케이션 정보에서 확인할 수 있는 Client ID와 Client Secret을 넣어주면 되겠습니다. 나머지 헤더 정보는 API요청 예시를 확인하면 알 수 있습니다.

#13(중요) - HTTP 요청을 하기 전에 헤더 정보를 이용해 request객체를 생성합니다. 저희가 사용하는 urllib 모듈에서 헤더 정보를 서버에 전달할 때 사용하는 대표적인 방법입니다.

#14(중요) - 생성된 request객체를 urlopen함수의 인수에 전달합니다. 이렇게 되면 헤더 정보를 포함하여 서버에게 HTTP 요청을 하게 됩니다.

#15 - 서버로부터 받은 데이터를 읽어 들입니다.

#16 - 읽어 들인 xml데이터를 이용해 원하는 데이터를 가져오기 위한 뷰티풀 수프 객체를 생성합니다.

#17 - 뷰티풀 수프 객체에서 우리가 필요한 데이터들이 존재하는 item태그를 모두 가져와서 저장합니다.

#18 - for문을 이용하여 title, description을 모두 공백을 제거한 후에 텍스트 파일에 write시킵니다. 구분하기 쉽도록 '-----\n' 라인을 추가로 write 해주었습니다.

#19 - 텍스트 파일이 저장 후 종료 됩니다.