

Project 06. Naver API 사용하기



네이버 API 사용 등록

- 네이버, 페이스북, 구글 등 IT의 거대 기업들부터 많은 S/W 관련 회사들이
- 자사 제품을 사용하는 것에 대해 API를 제공한다
- 특히 요즘처럼 웹크롤링에 대한 방어가 점점 심해지는 이때
- 해당 회사의 API를 사용하는 것이 오히려 더 바람직할 수 있다
- 점점 API의 사용에 익숙해질 필요 또한 분명하다

Naver API 등록

zero-base /

NAVER Developers

Products

Documents

Application

NAVER D2

Support

Forum

API 상태

Search Here

NAVER D²NDTI로 알아보는
나의 맞춤 채용 제안

N Pay 최대 30만원 이벤트

간단 이력서만 남겨주세요!

기술 유형을 분석해 최적의 네이버 채용 소식을 전합니다.



- <https://developers.naver.com/main/>
- 네이버 개발자 센터 
- 네이버 로그인  CLOVA
- 서비스 API 선택



네이버 아이디로 로그인



papago



서비스 API



네이버 클라우드 플랫폼

NAVER Developers

Products

Documents

Application

NAVER D2

Support

Forum

API 상태

Search Here



데이터랩

검색

단축URL

캡차

네이버 공유하기

모바일앱 연동

네이버 오픈메인

데이터랩

통합검색어 트렌드

통합검색어 트렌드는 네이버 통합검색에서 발생하는 검색어를 연령별, 성별, 기기별(PC, 모바일)로 세분화해서 조회할 수 있는 API입니다.

분석하고 싶은 주제군을 설정합니다

궁금한 주제어를 설정하고, 하위 주제어에 해당하는 검색어를 쉼표(,)로 구분해 입력합니다. 입력한 단어의 검색 추이를 하나로 합산해 해당 주제가 네이버에서 얼마나 검색되는지 조회할 수 있습니다.

예) 주제어 캠핑 : 캠핑, Camping, 캠핑용품, 겨울캠핑, 캠핑장, 글램핑, 오토캠핑, 캠핑카, 텐트, 캠핑요리

상대적 값으로만 제공됩니다

검색어 트렌드는 요청된 기간 중 검색 횟수가 가장 높은 시점을 100으로 두고 나머지는 상대적 값으로 제공하고 있습니다. 검색 횟수의 절댓값 제공은 아직 고려하고 있지 않습니다. 상대적 값만으로도 특정 주제, 검색어의 트렌드를 파악하는 데는 충분할 것입니다.



[오픈 API 이용 신청](#)

[개발 가이드 보기](#)

쇼핑인사이트

쇼핑인사이트는 네이버 통합검색의 쇼핑 영역과 네이버쇼핑 서비스의 검색 결과에서 사용자가 클릭한 데이터를 조회할 수 있는 API로, 소상공인에게 도움을 주기 위해 개발됐습니다. 쇼핑인사이트에서는 쇼핑 분야별, 연령별, 성별, 기기별(PC, 모바일)로 세분화된 사용자의 검색 클릭 트렌드를 조회할 수 있습니다.

- 오픈 API 이용 신청

[내 애플리케이션](#)[애플리케이션 등록](#)[API 제휴 신청](#)[계정 설정](#)

애플리케이션 등록 (API 이용신청)

애플리케이션의 기본 정보를 등록하면, 좌측 [내 애플리케이션](#) 메뉴의 서브 메뉴에 등록하신 애플리케이션 이름으로 서브 메뉴가 만들어집니다.

The screenshot shows the 'Application Name' input field containing 'ds_study' with a red arrow pointing to it. Below it is a dropdown menu titled '선택하세요.' with a red arrow pointing to it. The dropdown contains four options: '검색', '데이터랩 (검색어트렌드)', '데이터랩 (쇼핑인사이트)', and 'Clova Face Recognition'. Each option has a green checkmark icon to its right.

애플리케이션 이름 ↗

ds_study

- 네이버 아이디로 로그인할 때 사용자에게 표시되는 이름이므로 서비스 브랜드를 대표할 수 있는 이름으로 가급적 10자 이내로 간결하게 설정해주세요.
- 40자 이내의 영문, 한글, 숫자, 공백문자, "-", "_"만 입력 가능합니다.

선택하세요. ▼ ✓

검색

데이터랩 (검색어트렌드)

데이터랩 (쇼핑인사이트)

Clova Face Recognition

- 애플리케이션 이름 설정
- 사용자 API
 - 검색
 - 데이터랩(검색어트렌드)
 - 데이터랩(쇼핑인사이트)
 - Clova Face Recognition

환경 추가 ▼

WEB 설정 X ^

웹 서비스 URL (최대 10개)

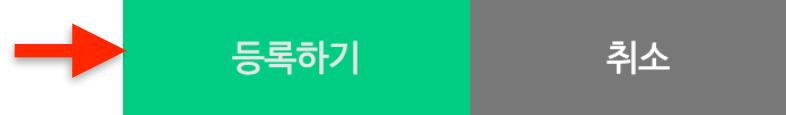
비로그인 오픈 API
서비스 환경

http://localhost + ✓

- 텍스트 폼 우측 끝의 '+' 버튼을 누르면 행이 추가되며, '-' 버튼을 누르면 행이 삭제됩니다.
- http와 https는 구분하지 않습니다.
- www는 빼고 입력해 주세요. 예) http://naver.com
- 서브 도메인이 있으면 대표 도메인명만 입력해 주세요. (예: http://naver.com)
- 하이브리드 앱은 location.href 객체 출력 값을 입력하면 됩니다. (예: file:///로컬 URI)

- 환경추가
- WEB설정
- http://localhost 입력

- 텍스트 품 우측 끝의 '+' 버튼을 누르면 행이 추가되며, '-' 버튼을 누르면 행이 삭제됩니다.
- http와 https는 구분하지 않습니다.
- www는 빼고 입력해 주세요. 예) http://naver.com
- 서브 도메인이 있으면 대표 도메인명만 입력해 주세요. (예: http://naver.com)
- 하이브리드 앱은 location.href 객체 출력 값을 입력하면 됩니다. (예: file:///로컬 URI)



- 등록하기 선택

내 애플리케이션

ds_study

애플리케이션 등록

API 제휴 신청

계정 설정

ds_study

개요

API 설정

멤버관리

로그인 통계

API 통계

Playground (Beta)

애플리케이션 정보

Client ID

sqqPisXslZSkUEB5xQXT

Client Secret

보기

- Client ID
- Client Secret
- 복사해두기

The screenshot shows a browser window with the URL `developers.naver.com/apps/#/myapps/sqqPisXslZSkUEB5xQXT/overview` highlighted by a red box. The page title is "Naver API 등록". The top navigation bar includes links for NAVER Developers, Products, Documents, Application, NAVER D2, Support, Forum, and a green "API 상태" button. A search bar and user profile icon are also present. On the left, there's a sidebar with "내 애플리케이션" and a list containing "ds_study". The main content area shows the application "ds_study" with tabs for "개요", "API 설정", "멤버관리", "로그인 통계", "API 통계", and "Playground(Beta)". The "개요" tab is selected, displaying fields for "Client ID" (value: sqqPisXslZSkUEB5xQXT) and "Client Secret" (value: masked). A "보기" button is next to the Client Secret field.

- 위 주소를 즐겨찾기에 두면 나중에 찾기 편리하다

ds_study

애플리케이션 등록

API 제휴 신청

계정 설정

API 호출 안내

지도 API 인증실패나 네이버 로그인 이용 제한이 걸렸다면 [API 설정] 탭에서 URL 관련 설정을 수정하시면 정상 이용 가능합니다 !!!

비로그인 오픈 API 당일 사용량

API호출량/일일허용량

검색

0/25000

데이터랩 (검색어트렌드)

0/1000

데이터랩 (쇼핑인사이트)

0/1000

Clova Face Recognition

0/1000

- 일일 사용량이 정해져 있다

Naver API 등록

zero-base /

NAVER Developers

Products → Documents

Application

NAVER D2

Support

Forum

API 상태

Search Here



API 공통 가이드

CLOVA

네이버 아이디로 로그인

파파고

서비스 API

개요

Papago 번역

데이터랩

검수 가이드

언어 감지

검색

개발 가이드

한글인명-로마자 변환

단축 URL

API 명세

이미지캡차

튜토리얼

음성캡차

SDK 다운로드

네이버 공유하기

모바일 앱 연동

네이버 오픈메인

• 개발 가이드 보기

• Documents 선택

• 서비스 API 선택

• 검색 선택

지식iN

지역

오타변환

웹문서

이미지

쇼핑

전문자료

오픈 API 이용 신청 >

0.API 호출 예제

예제 실행 전에 아래 1.준비사항 항목들을 꼭 체크하시길 바랍니다.

Java

네이버 검색 API 예제는 블로그를 비롯 전문자료까지 호출방법이 동일하므로 blog검색만 대표로 예제를 올렸습니다.

```
// 네이버 검색 API 예제 - blog 검색
import java.io.*;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLEncoder;
import java.util.HashMap;
import java.util.Map;
```

Naver API 등록

zero-base /

NAVER Developers

Products

Documents

Application

NAVER D2

Support

Forum

API 상태

Search Here



Documents

네이버 오픈 API를 이용해 창의적인 애플리케이션을 제작해 보세요.

Documents > 서비스 API > 검색

블로그

블로그

개발 가이드에 맞춰 먼저 간단히 테스트 해보기

카페글

지식iN

지역

오타변환

웹문서

혹시 발급받은 Client ID와 Client Secret 값을 같이 전송해 주시면 활용 가능합니다.

[오픈 API 이용 신청 >](#)

O.API 호출 예제

예제 실행 전에 아래 [1.준비사항](#) 항목들을 꼭 체크하시길 바랍니다.

네이버 검색 API 사용하기



지식iN

지역

오타변환

웹문서

이미지

쇼핑

전문자료

Python

```
# 네이버 검색 API예제는 블로그를 비롯 전문자료까지 호출방법이 동일하므로 blog검색만 대표로 예제를 올렸습니다.
# 네이버 검색 Open API 예제 - 블로그 검색
import os
import sys
import urllib.request
client_id = "YOUR_CLIENT_ID"
client_secret = "YOUR_CLIENT_SECRET"
encText = urllib.parse.quote("검색할 단어")
url = "https://openapi.naver.com/v1/search/blog?query=" + encText # json 결과
# url = "https://openapi.naver.com/v1/search/blog.xml?query=" + encText # xml 결과
request = urllib.request.Request(url)
request.add_header("X-Naver-Client-Id",client_id)
request.add_header("X-Naver-Client-Secret",client_secret)
response = urllib.request.urlopen(request)
rescode = response.getcode()
if(rescode==200):
    response_body = response.read()
    print(response_body.decode('utf-8'))
else:
    print("Error Code:" + rescode)
```

- API에 대한 Python 예제

뉴스
책

성인 검색어 판별

백과사전

영화

카페글

지식iN

지역

오타변환

웹문서

이미지

쇼핑

전문자료

1. 준비사항

- 애플리케이션 등록: 네이버 오픈 API로 개발하시려면 먼저 '[Application-애플리케이션 등록](#)' 메뉴에서 애플리케이션을 등록하셔야 합니다.
[\[자세한 방법 보기\]](#)
- 클라이언트 ID와 secret 확인: '[내 애플리케이션](#)'에서 등록한 애플리케이션을 선택하면 Client ID와 Client Secret 값을 확인할 수 있습니다.
- API 권한 설정: '[내 애플리케이션](#)'의 'API 권한관리' 탭에서 사용하려는 API가 체크되어 있는지 확인합니다. 체크되어 있지 않을 경우 403 에러(API 권한 없음)가 발생하니 주의하시기 바랍니다.

2. API 기본 정보

메서드	인증	요청 URL	출력 포맷
GET	-	https://openapi.naver.com/v1/search/blog.xml	XML
GET	-	https://openapi.naver.com/v1/search/blog.json	JSON

- 그 중에 API 기본 정보 항목에 보면 요청 URL이 있다
- 웹 API는 이 경로를 이용한다

ds_study

개요	API 설정	멤버관리	로그인 통계
애플리케이션 정보			
Client ID	sqqPisXslZSkUEB5xQXT		
Client Secret	보기	

```
1 # 네이버 검색 API 예제는 블로그를 비롯 전문자료까지 호출방법
2 # 네이버 검색 Open API 예제 - 블로그 검색
3 import os
4 import sys
5 import urllib.request
6 client_id = "sqqPisXslZSkUEB5xQXT" ← Client ID
7 client_secret = "_rxReBpug1" ← Client Secret
8 encText = urllib.parse.quote("파이썬")
9 url = "https://openapi.naver.com/v1/search/blog?query=" +
10 # url = "https://openapi.naver.com/v1/search/blog.xml?quer
```

- 예제 코드를 Jupyter Notebook에 복사
- client_id, client_secret 변수에 발급받은 내용을 넣어준다

```
6 client_id = "sqqPisXs1ZSkUEB5xQXT"
7 client_secret = "_rxReBpug1"
8 encText = urllib.parse.quote("파이썬")
9 url = "https://openapi.naver.com/v1/search/blog?query=" + encText # json 결과
10 # url = "https://openapi.naver.com/v1/search/blog.xml?query=" + encText # xml 결과
11 request = urllib.request.Request(url)
12 request.add_header("X-Naver-Client-Id",client_id)
13 request.add_header("X-Naver-Client-Secret",client_secret)
14 response = urllib.request.urlopen(request)
```

- **urllib**: http 프로토콜에 따라서 서버의 요청/응답을 처리하기 위한 모듈
- **urllib.request**: 클라이언트의 요청을 처리하는 모듈
- **urllib.parse**: url 주소에 대한 분석

```
6 client_id = "sqqPisXs1ZSkUEB5xQXT"
7 client_secret = "_rxReBpug1"
8 encText = urllib.parse.quote("파이썬")
9 url = "https://openapi.naver.com/v1/search/blog?query=" + encText # json 결과
# url = "https://openapi.naver.com/v1/search/blog.xml?query=" + encText # xml 결과
10 request = urllib.request.Request(url)
11 request.add_header("X-Naver-Client-Id",client_id)
12 request.add_header("X-Naver-Client-Secret",client_secret)
13 response = urllib.request.urlopen(request)
```



- 이 주소가 네이버 블로그 검색 결과를 가져오는 주소이다

In [2]: 1 response

Out[2]: <http.client.HTTPResponse at 0x12b4a1610>

In [3]: 1 response.getcode()

Out[3]: 200

- 응답 결과를 **response**에 저장
- 결과를 보기 위해서는 약간의 작업이 더 필요하다
- **http.client.HTTPResponse** 클래스는 **getcode()** 함수를 이용해서 응답의 상태를 확인할 수 있다
- 200은 정상

블로그

뉴스

책

성인 검색어 판별

백과사전

영화

카페글

지식iN

지역

오타변환

웹문서

이미지

쇼핑

전문자료

5. 에러 코드

공통 에러 코드는 [여기](#)를 참조하세요.

HTTP 코드	에러 코드	에러 메시지	조치 방안
400	SE01	Incorrect query request (잘못된 쿼리요청입니다.)	검색 API 요청에 오류가 있습니다. 요청 URL, 필수 요청 변수가 정확한지 확인 바랍니다.
400	SE02	Invalid display value (부적절한 display 값입니다.)	display 요청 변수값이 허용 범위(1~100)인지 확인해 보세요.
400	SE03	Invalid start value (부적절한 start 값입니다.)	start 요청 변수값이 허용 범위(1~1000)인지 확인해 보세요.
400	SE04	Invalid sort value (부적절한 sort 값입니다.)	sort 요청 변수 값에 오타가 없는지 확인해 보세요.
400	SE06	Malformed encoding (잘못된 형식의 인코딩입니다.)	검색어를 UTF-8로 인코딩하세요.
404	SE05	Invalid search api (존재하지 않는 검색 api입니다.)	검색 API 대상에 오타가 없는지 확인해 보세요.
500	SE99	System Error (시스템 에러)	서버 내부 에러가 발생하였습니다. 포럼에 올려주시면 신속히 조치하겠습니다.

- 사용하는 서비스마다 에러코드가 조금씩 다르므로 잘 확인하자

```
In [16]: print(response.read().decode('utf-8'))
```

```
{
  "lastBuildDate": "Sun, 05 May 2019 15:13:01 +0900",
  "total": 97347,
  "start": 1,
  "display": 10,
  "items": [
    {
      "title": "<b>파이썬</b>학원 목표를 달성하려면",
      "link": "https://blog.naver.com/jjkbw1234?Redirect=Log&logNo=221497979172",
      "description": "왜 이이 이야기를 하냐고요~ 바로 제가 정보보안전문가가 되기 위해서 지금 <b>파이썬</b>학원을 다니고... 기본적으로 <b>파이썬</b>이라는 것은 좀 쉽게 개발자들이 쉽게 활용을 할 수 있는 프로그램으로서 R기반으로... ",
      "bloggername": "달콤한, 뼈빠롱스타킹!",
      "bloggerlink": "https://blog.naver.com/jjkbw1234",
```

- 글자로 읽을 때 `decode`로 `utf-8`을 설정해줘야 한다.

```
8 encText = urllib.parse.quote("파이썬")
9 url = "https://openapi.naver.com/v1/search/book?query=" + encText # json 결과
# url = "https://openapi.naver.com/v1/search/blog.xml?query=" + encText # xml 결과
10 request = urllib.request.Request(url)
11 request.add_header("X-Naver-Client-Id",client_id)
12 request.add_header("X-Naver-Client-Secret",client_secret)
13 response = urllib.request.urlopen(request)
```

- 이 주소는 네이버 책 검색 결과를 가져오는 주소이다

```
{  
    "lastBuildDate": "Tue, 28 Sep 2021 14:46:19 +0900",  
    "total": 1375,  
    "start": 1,  
    "display": 10,  
    "items": [  
        {  
            "title": "이것이 취업을 위한 코딩 테스트다 with <b>파이썬</b> (취업과 이직을 결정하는 알고리즘 인터뷰 완벽 가이드, C/C++, 자바 코드 제공)",  
            "link": "http://book.naver.com/bookdb/book_detail.php?bid=16439154",  
            "image": "https://bookthumb-phinf.pstatic.net/cover/164/391/16439154.jpg?type=m1&update=20200928",  
            "author": "나동빈",  
            "price": "34000",  
            "discount": "30600",  
            "publisher": "한빛미디어",  
            "pubdate": "20200805",  
            "isbn": "1162243074 9791162243077",  
            "description": "8가지 핵심 알고리즘 이론을 쉽게 설명하고, 관련 실전 문제를 풀이했다. 출제 유형 분석, 이론 설명, 기출문제 풀이까지! 어떤 코딩 테스트도 이 책 한 권으로 대비할 수 있을 것이다. 코딩 테스트에서 주로 선택하는 <b>"  
        }  
    ]  
}
```

```
8 encText = urllib.parse.quote("파이썬")
9 url = "https://openapi.naver.com/v1/search/movie?query=" + encText # json 결과
# url = "https://openapi.naver.com/v1/search/blog.xml?query=" + encText # xml 결과
10
11 request = urllib.request.Request(url)
12 request.add_header("X-Naver-Client-Id",client_id)
13 request.add_header("X-Naver-Client-Secret",client_secret)
14 response = urllib.request.urlopen(request)
15 rescode = response.getcode()
```



- 이 주소는 네이버 영화 검색 결과를 가져오는 주소이다

```
{  
    "lastBuildDate": "Tue, 28 Sep 2021 14:47:35 +0900",  
    "total": 1,  
    "start": 1,  
    "display": 1,  
    "items": [  
        {  
            "title": "<b>파이썬</b> 앤 가드",  
            "link": "https://movie.naver.com/movie/bi/mi/basic.nhn?code=152070",  
            "image": "https://ssl.pstatic.net/imgmovie/mdb/m110/1520/152070_P01_145336.jpg",  
            "subtitle": "PYTHON AND GUARD",  
            "pubDate": "2015",  
            "director": "안톤 디아코프!",  
            "actor": "",  
            "userRating": "0.00"  
        }  
    ]  
}
```

- 파이썬이라는 단어가 들어가있는 제목의 영화가 있을 줄은..

```
8 encText = urllib.parse.quote("파이썬")
9 url = "https://openapi.naver.com/v1/search/cafearticle?query=" + encText # json 결과
10 # url = "https://openapi.naver.com/v1/search/blog.xml?query=" + encText # xml 결과
11 request = urllib.request.Request(url)
12 request.add_header("X-Naver-Client-Id",client_id)
13 request.add_header("X-Naver-Client-Secret",client_secret)
14 response = urllib.request.urlopen(request)
15 rescode = response.getcode()
```



- 이 주소는 공개된 네이버 카페 검색 결과를 가져오는 주소이다

```
{  
    "lastBuildDate": "Tue, 28 Sep 2021 14:48:13 +0900",  
    "total": 120243,  
    "start": 1,  
    "display": 10,  
    "items": [  
        {  
            "title": "<b>파이썬</b> 강의 프로그래밍 수업 무료로 듣자!",  
            "link": "http://cafe.naver.com/anjun/885594",  
            "description": "※이 글은 산준모카페 관리자 승인 하에 작성된 글입니다 <b>파이썬</b> 강의 프로그래밍 수업 무료로 듣자! <b>파이썬</b> 강의 프로그래밍 수업 무료로 듣자! 안녕하세요 IT전문 학습멘토 동자쌤입니다 프로그래밍을 배우기로...",  
            "cafename": "산준모★산업안전기사 | 위험물산업기사...",  
            "cafeurl": "https://cafe.naver.com/anjun"
```

```
8 encText = urllib.parse.quote("파이썬")
9 url = "https://openapi.naver.com/v1/search/shop?query=" + encText # json 결과
10 # url = "https://openapi.naver.com/v1/search/blog.xml?query=" + encText # xml 결과
11 request = urllib.request.Request(url)
12 request.add_header("X-Naver-Client-Id",client_id)
13 request.add_header("X-Naver-Client-Secret",client_secret)
14 response = urllib.request.urlopen(request)
15 rescode = response.getcode()
```



- 이 주소는 공개된 네이버 카페 검색 결과를 가져오는 주소이다

```
{  
    "lastBuildDate": "Tue, 28 Sep 2021 14:48:36 +0900",  
    "total": 343989,  
    "start": 1,  
    "display": 10,  
    "items": [  
        {  
            "title": "잘모이 최신상 리얼 <b>파이톤</b> 셀리나 스마트백",  
            "link": "https://search.shopping.naver.com/gate.nhn?id=26517760078",  
            "image": "https://shopping-phinf.pstatic.net/main_2651776/26517760078.20210326040741.jpg",  
            "lprice": "89180",  
            "hprice": "",  
            "mallName": "네이버",  
            "productId": "26517760078",  
            "productType": "1",  
            "brand": "잘모이",  
            "maker": "",  
            "category1": "패션잡화",  
            "category2": "여성가방",  
        }  
    ]  
}
```

- 파이썬(파이톤)이 들어가는 제품이 있는 듯 하다

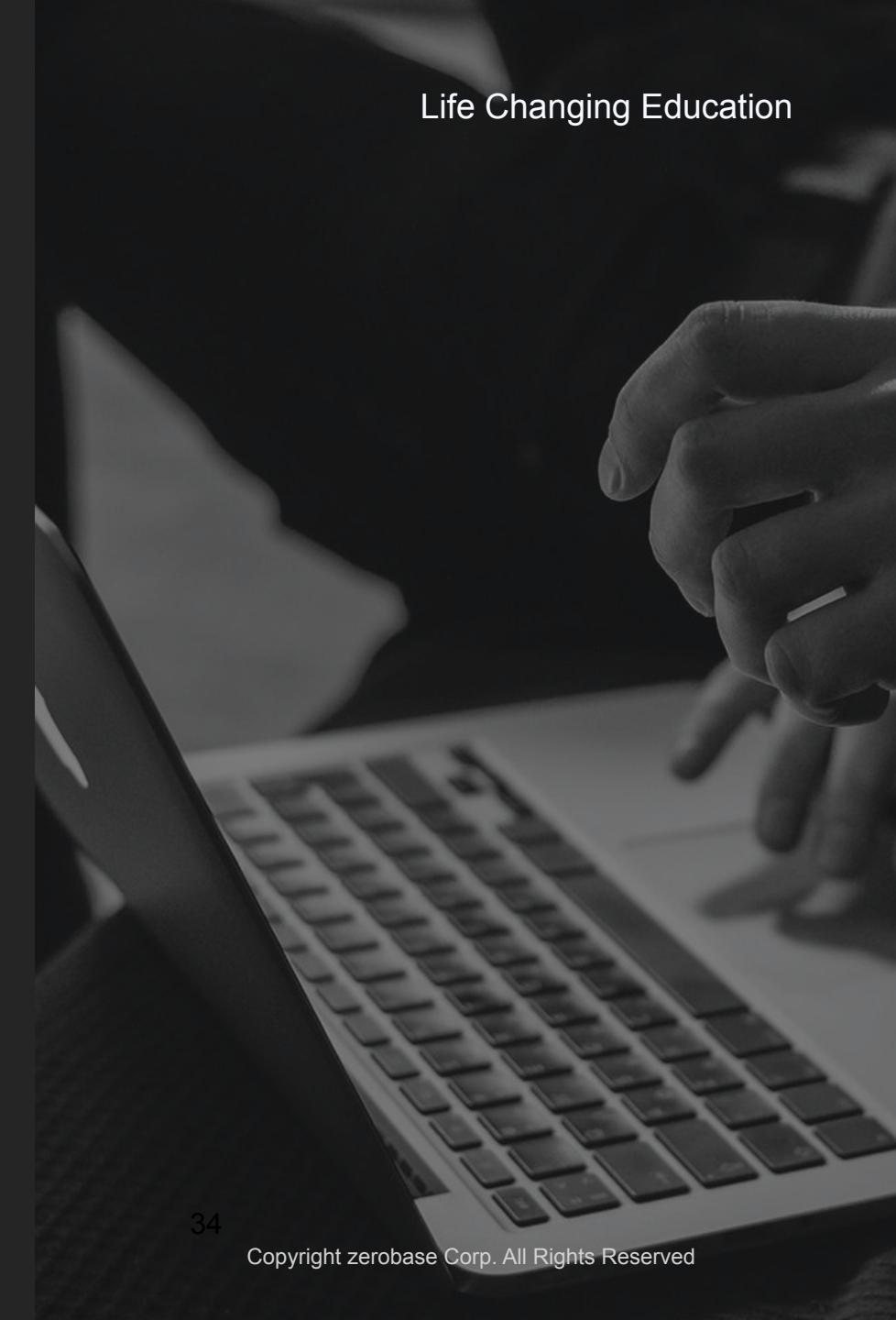
```
8 encText = urllib.parse.quote("파이썬")
9 url = "https://openapi.naver.com/v1/search/encyc?query=" + encText # json 결과
# url = "https://openapi.naver.com/v1/search/blog.xml?query=" + encText # xml 결과
10
11 request = urllib.request.Request(url)
12 request.add_header("X-Naver-Client-Id",client_id)
13 request.add_header("X-Naver-Client-Secret",client_secret)
14 response = urllib.request.urlopen(request)
15 rescode = response.getcode()
```



- 이 주소는 공개된 네이버 카페 검색 결과를 가져오는 주소이다

```
{  
    "lastBuildDate": "Tue, 28 Sep 2021 14:49:15 +0900",  
    "total": 442,  
    "start": 1,  
    "display": 10,  
    "items": [  
        {  
            "title": "<b>파이썬</b>",  
            "link": "https://terms.naver.com/entry.naver?docId=3580815&cid=59088&categoryId=59096",  
            "description": "'<b>파이썬</b>'이다. 간결한 문법으로 입문자가 이해하기 쉽고, 다양한 분야에 활용할 수 있기 때문이다. 이 외에도 <b>파이썬</b>은 머신러닝, 그래픽, 웹 개발 등 여러 업계에서 선호하는 언어로 꾸준히...",  
            "thumbnail": "http://openapi-dbsctrhumb.phinf.naver.net/4749_000_1/20170118193349632_0CHSSS5Y6.png/01_16.png?type=m160_160"  
        }  
    ]  
}
```

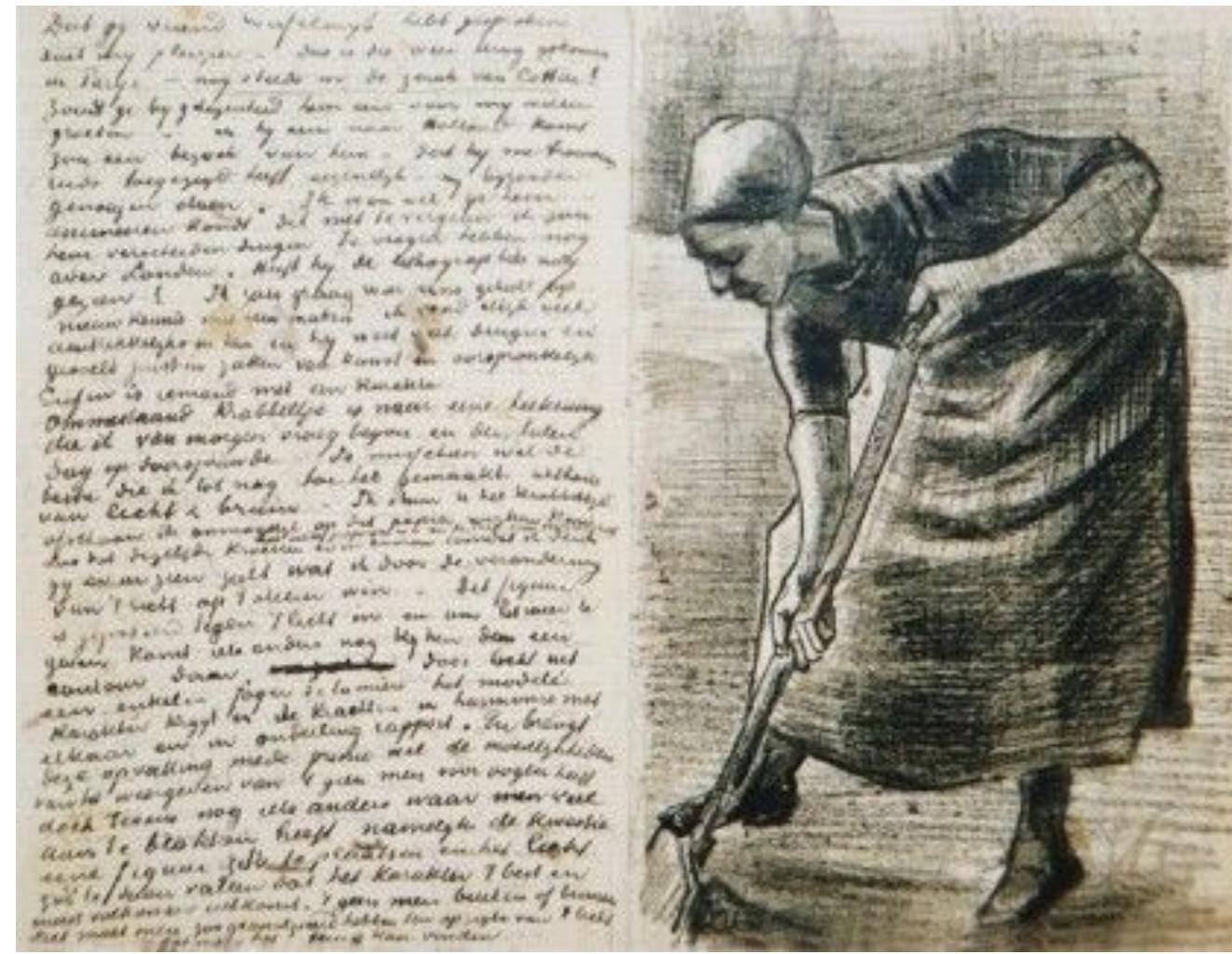
상품 검색(feat. 몰스킨)



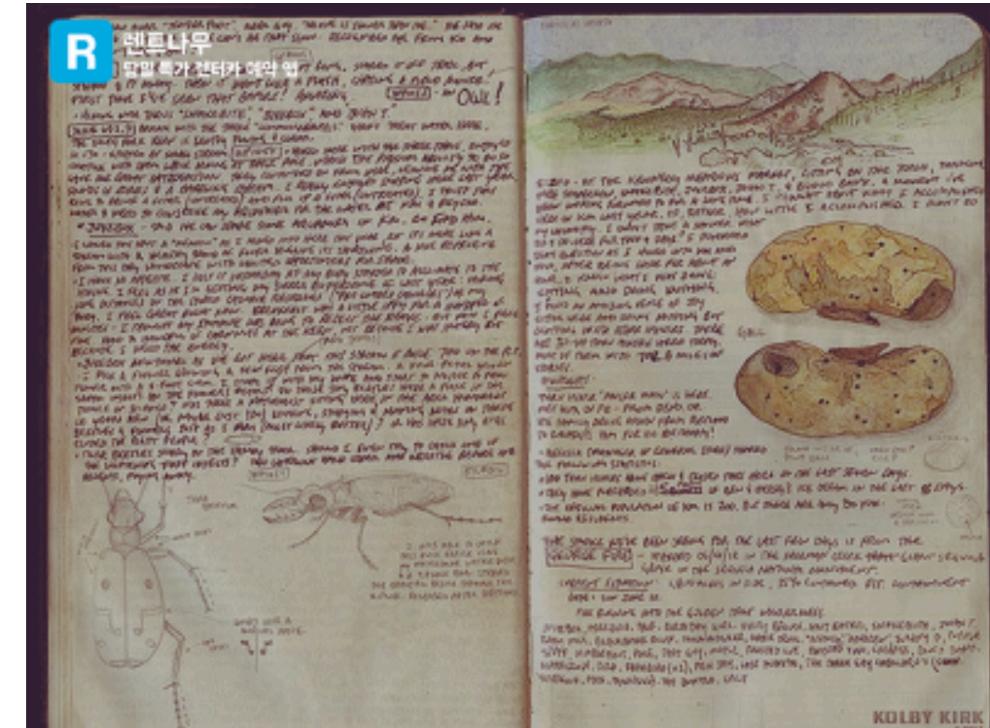
```
8 encText = urllib.parse.quote("몰스킨")
9 url = "https://openapi.naver.com/v1/search/shop?query=" + encText # json 결과
# url = "https://openapi.naver.com/v1/search/blog.xml?query=" + encText # xml 결과
11 request = urllib.request.Request(url)
12 request.add_header("X-Naver-Client-Id",client_id)
13 request.add_header("X-Naver-Client-Secret",client_secret)
14 response = urllib.request.urlopen(request)
15 rescode = response.getcode()
16 if(rescode==200):
17     response_body = response.read()
18     print(response_body.decode('utf-8'))
19 else:
20     print("Error Code:" + rescode)
```

- 몰스킨이라는 이름으로 검색

```
{  
    "lastBuildDate": "Tue, 28 Sep 2021 14:50:02 +0900",  
    "total": 28081,  
    "start": 1,  
    "display": 10,  
    "items": [  
        {  
            "title": "한정판 <b>몰스킨</b> 2022 해리포터 데일리 다이어리 포켓",  
            "link": "https://search.shopping.naver.com/gate.nhn?id=28719894754",  
            "image": "https://shopping-phinf.pstatic.net/main_2871989/28719894754.20210906035419.jpg",  
            "lprice": "31840",  
            "hprice": "",  
            "mallName": "네이버",  
            "productId": "28719894754",  
            "productType": "1",  
            "brand": "몰스킨",  
            "maker": "몰스킨",  
            "category1": "생활/건강",  
            "category2": "문구/사무용품",  
        }]
```



• 반고호가 쓴 몰스킨 노트의 메모



- 몰스킨은 마케팅으로 성공을 이끈 대표적인 아날로그 감성 제품
- 창조적인 일을 하는 사람들이, 창조성을 기록하는 책…의 이미지를 가짐

3. 몰스킨은 눈에 보이는 것 이상을 판매한다!

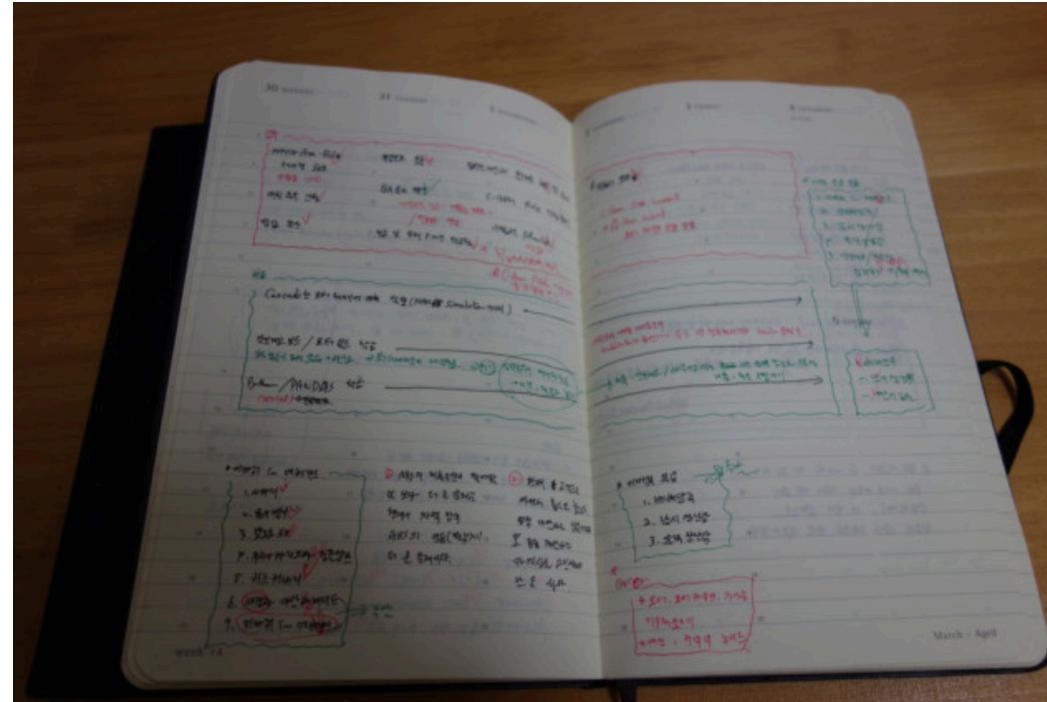


악마는 프라다를 입는다 - 몰스킨에 메모하는 앤 해서웨이

몰스킨은 '보이지 않는' 제품 디자인과 기획력을 가지고 있다. 마영범 소갤러리 대표는 "최근 디자인의 영역은 외형에만 머무르지 않고 창조적인 소비자들이 민감하게 여기는 '보이지 않는 가치'로 영역을 넓히고 있는데, 몰스킨은 그 대표적인 사례"라고 말하고 있다.

- <https://brunch.co.kr/@univbrand/7>
- 블로그 - <우리가 몰랐던 몰스킨의 5가지 비밀>에서 발췌

상품검색



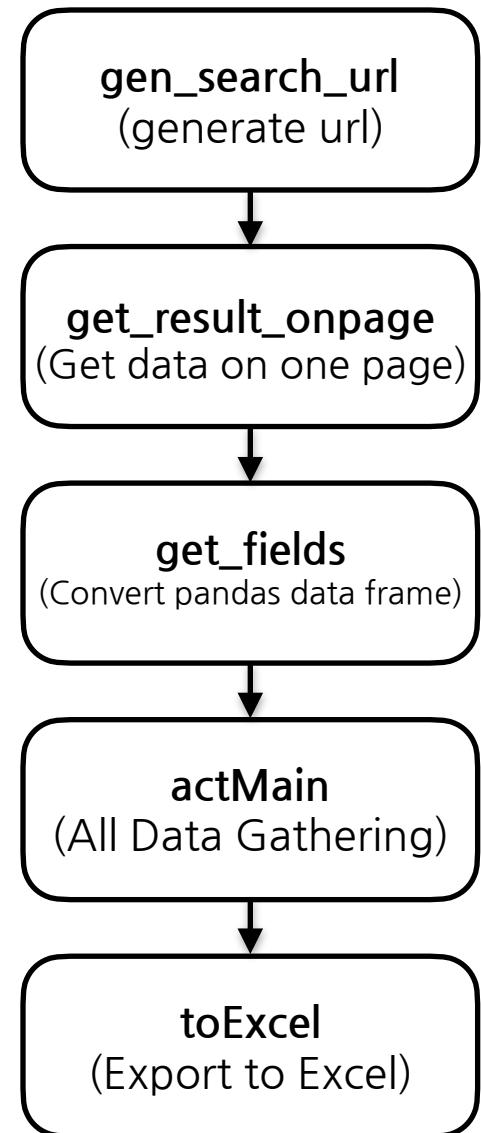
- 일상을 적고 기록하는 것은 참으로 중요한듯…



이런 역사를 가진 몰스킨에 대해

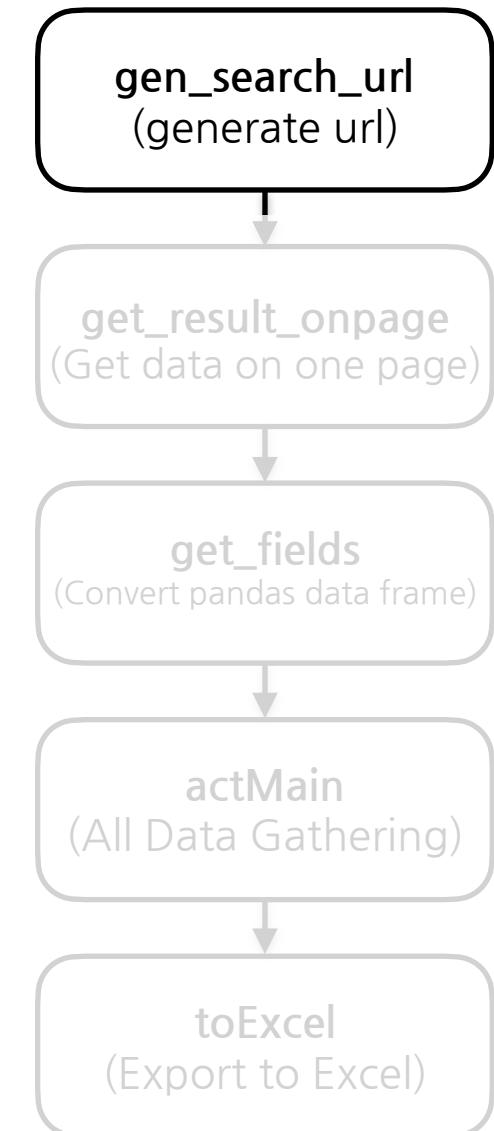
쇼핑 정보를 취득하는 과정을 작은 주제로 실습해 보겠습니다.

- 옆에 보이는 순서로 함수를 만들어 갈 겁니다.
- 이 순서는 단지 오늘 수업의 순서일 뿐이지만,
- 이제 이런 순서도가 없으면 코드의 흐름을 잡기 어려울 수 있습니다.
- 더 중요한 것은 이건 SW 개발과는 관련이 없이 수업을 위한 것입니다.
- 좋은 SW는 보다 더 체계적으로 개발됩니다.
- 우리는 단지 우리 수준에 맞춰서~^^



gen_search_url

zero-base /



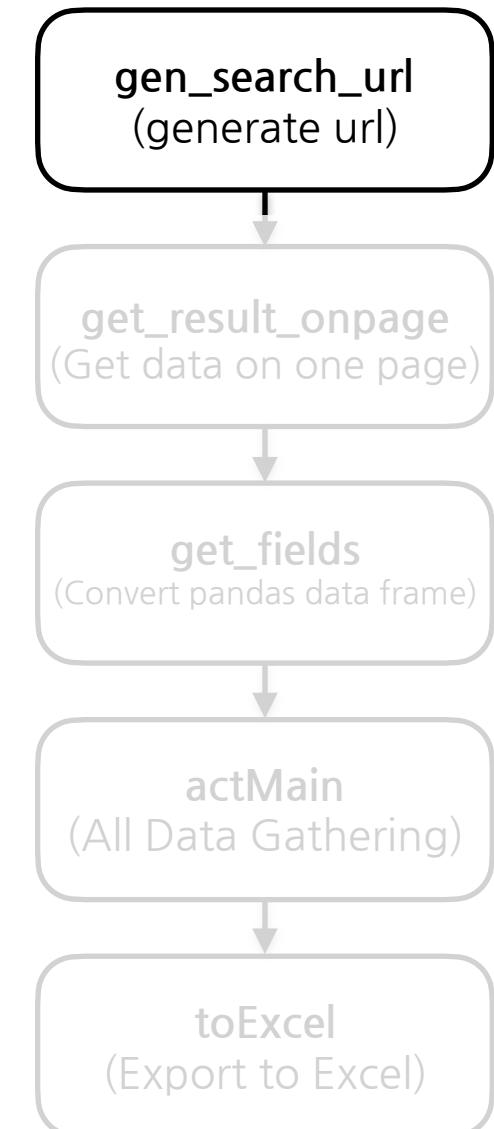
- 먼저 검색해야할 URL 만들기
- 앞선 내용으로 검색할 URL과 그 URL에는 많은 옵션이 있는데,
- 해당 옵션을 사용해서 URL을 만들어야 한다

3. 요청 변수

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

- 위 요청 변수들을 URL에 포함시켜야 한다

gen_search_url



In [5]:

```

1 def gen_search_url(api_node, search_text, start_num, disp_num):
2     base = "https://openapi.naver.com/v1/search"
3     node = "/" + api_node + ".json"
4     param_query = "?query=" + urllib.parse.quote(search_text)
5     param_start = "&start=" + str(start_num)
6     param_disp = "&display=" + str(disp_num)
7
8     return base + node + param_query + param_start + param_disp
  
```

- 일단 URL을 만드는 함수를 만들고 테스트

In [47]:

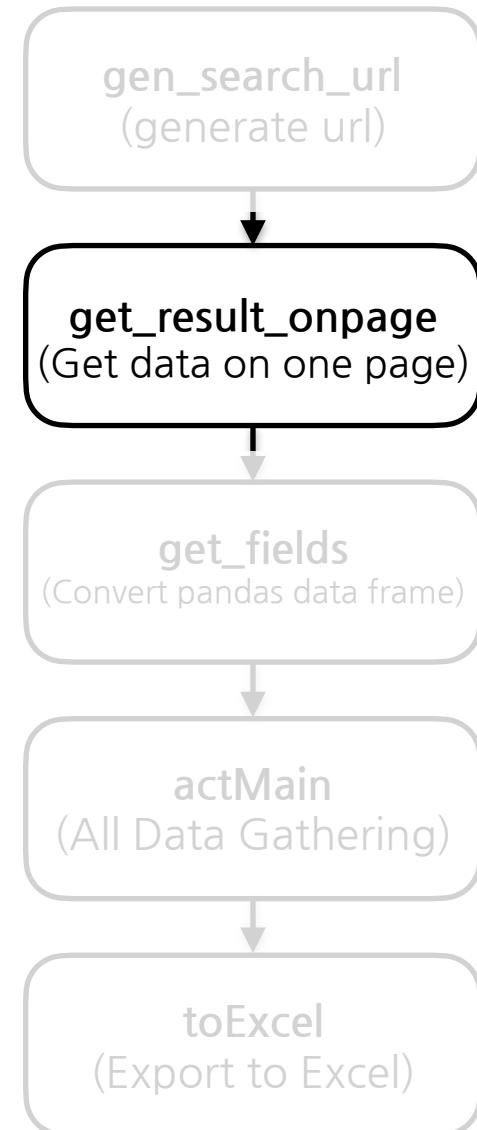
```
1 gen_search_url("shop", "TEST", 10, 3)
```

```
Out[47]: 'https://openapi.naver.com/v1/search/shop.json?query=TEST&display=3'
```

- 결과가 잘 나온다

get_result_onpage

zero-base /



In [60]:

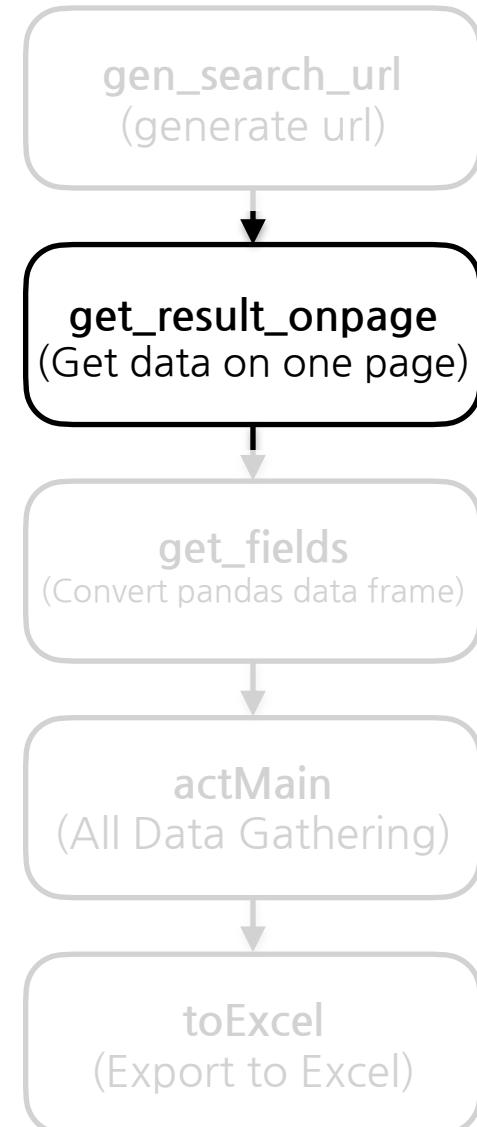
```

1 import json
2 import datetime
3
4 def get_result_onpage(url):
5     request = urllib.request.Request(url)
6     request.add_header("X-Naver-Client-Id", client_id)
7     request.add_header("X-Naver-Client-Secret", client_secret)
8     response = urllib.request.urlopen(request)
9     print("[%s] Url Request Success" % datetime.datetime.now())
10    return json.loads(response.read().decode("utf-8"))
  
```

- 혹시나 하고 말씀드리는데..
- 여기에서 우리가 모르는 것은 없습니다~^^

get_result_onpage

zero-base /



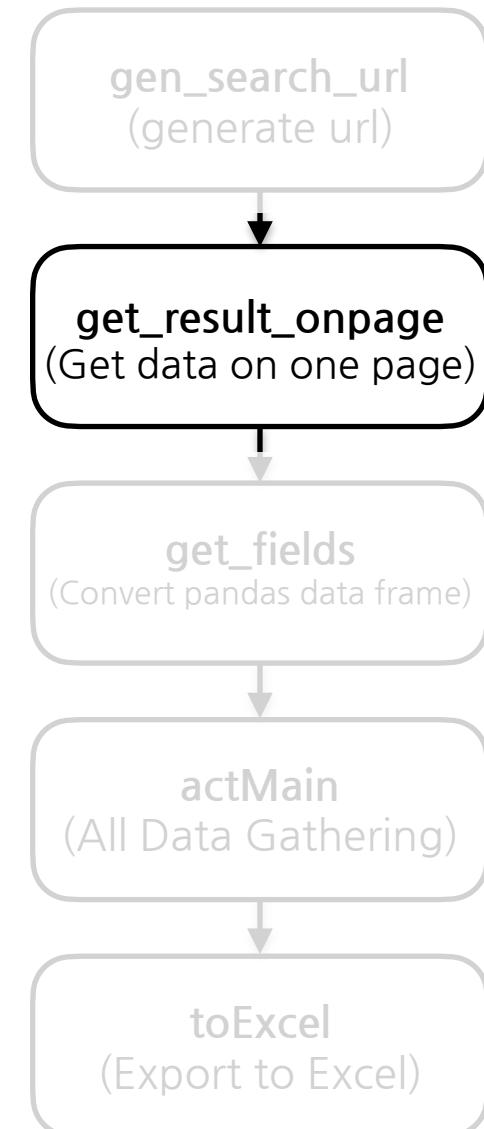
In [60]:

```

1 import json
2 import datetime
3
4 def get_result_onpage(url):
5     request = urllib.request.Request(url)
6     request.add_header("X-Naver-Client-Id", client_id)
7     request.add_header("X-Naver-Client-Secret", client_secret)
8     response = urllib.request.urlopen(request)
9     print("[%s] Url Request Success" % datetime.datetime.now())
10    return json.loads(response.read().decode("utf-8"))
  
```

- 혹시나 하고 말씀드리는데..
- 여기에서 우리가 모르는 것은 없습니다~^^

get_result_onpage



```
In [62]: 1 client_id = "sqqPisXs1ZSkUEB5xQXT"
2 client_secret = "_rxReBpug1"
3
4 url = gen_search_url("shop", "몰스킨", 1, 5)
5 one_result = get_result_onpage(url)
```

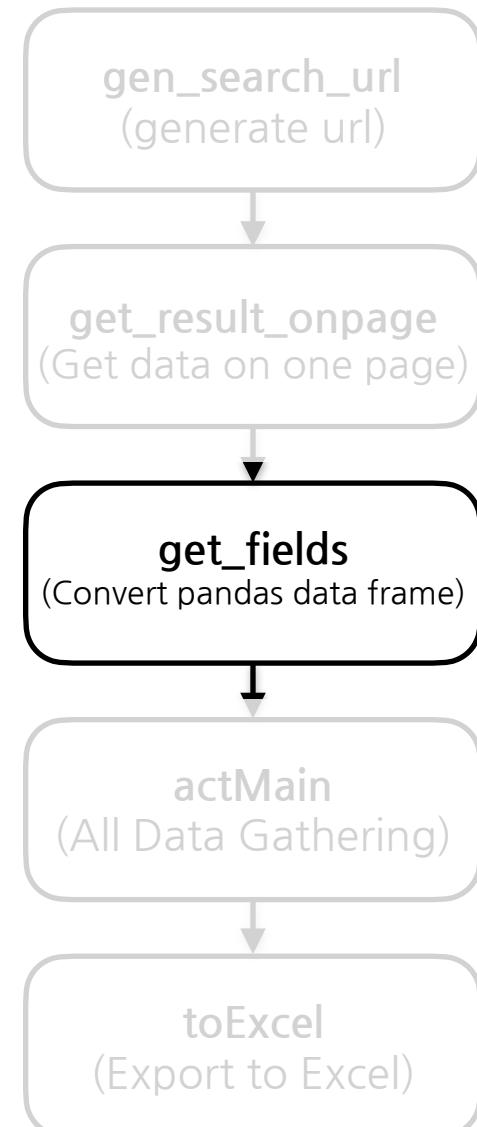
[2021-09-28 15:05:07.876484] Url Request Success

```
In [71]: 1 one_result
```

```
Out[71]: {'lastBuildDate': 'Tue, 28 Sep 2021 15:05:07 +0900',
'total': 28081,
'start': 1,
'display': 5,
'items': [{'title': '한정판 <b>몰스킨</b> 2022 해리포터 데일리 다이어리 포켓',
'link': 'https://search.shopping.naver.com/gate.nhn?id=28719894754',
'image': 'https://shopping-phinf.pstatic.net/main_2871989/28719894754.20210906035419.jpg',
'lprice': '31840',
'hprice': '',
'mallName': '네이버',
'productId': '28719894754',
```

get_fields

zero-base /



In [68]: 1 one_result["items"][0]["title"]
Out[68]: '한정판 몰스킨 2022 해리포터 데일리 다이어리 포켓'

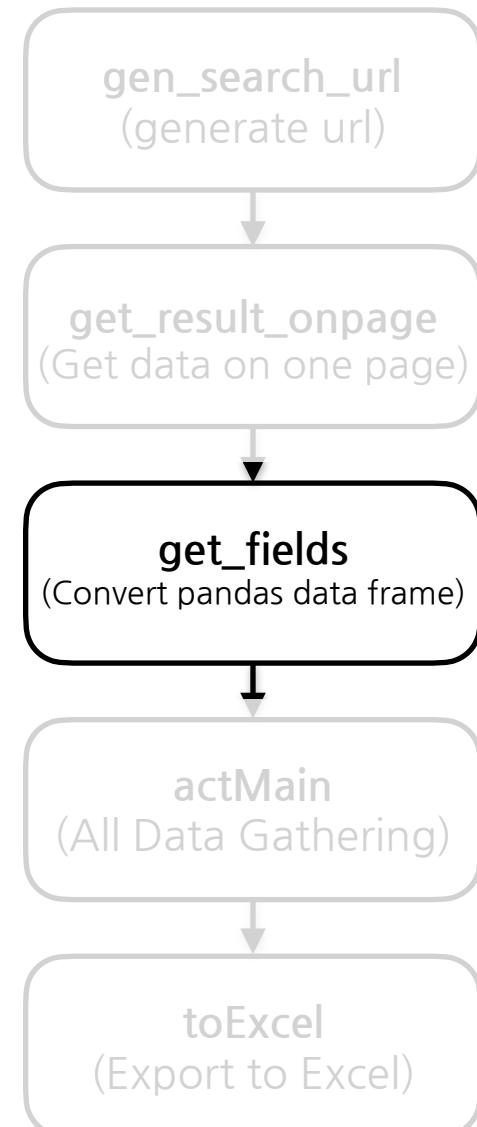
In [69]: 1 one_result["items"][0]["lprice"]
Out[69]: '31840'

In [70]: 1 one_result["items"][0]["mallName"]
Out[70]: '네이버'

- **one_result** 의 “items” 항목 안에서 검색 결과 가져오기

get_fields

zero-base /



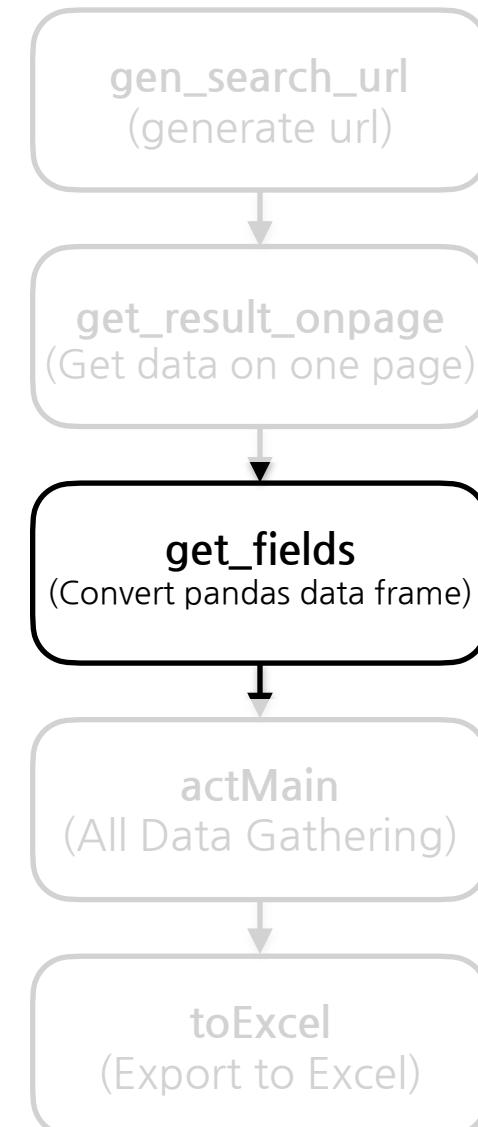
```

1 import pandas as pd
2
3
4 def get_fields(json_data):
5     title = [each["title"] for each in json_data['items']]
6     link = [each["link"] for each in json_data['items']]
7     lprice = [each["lprice"] for each in json_data['items']]
8     mall_name = [each["mallName"] for each in json_data['items']]
9
10    result_pd = pd.DataFrame({
11        "title": title,
12        "lprice": lprice,
13        "link": link,
14        "mall": mall_name,
15    }, columns=["title", "lprice", "link", "mall"])
16    return result_pd
  
```

- 이렇게 해서 간단하게 **pandas data frame**으로 출력

get_fields

zero-base /



In [40]: 1 get_fields(one_result)

Out[40]:

	title	lprice	link	mall
0	한정판 몰스킨 2022 해리포터 데일리 다이어리 포켓	31750	https://search.shopping.naver.com/gate.nhn? id=...	네이버
1	몰스킨 클래식노트 플레인 하드 L	15330	https://search.shopping.naver.com/gate.nhn? id=...	네이버
2	몰스킨 클래식 노트 소프트커버 도트 포켓 머틀그린	18810	https://search.shopping.naver.com/gate.nhn? id=...	네이버
3	몰스킨 2022 다이어리 데일리 위클리 포켓 고급 명품 감성 휴대용 여...	24100	https://search.shopping.naver.com/gate.nhn? id=...	에코펜
4	한정판 몰스킨 2022 피너초 위클리 다이어리 포켓	29640	https://search.shopping.naver.com/gate.nhn? id=...	네이버

- 그런데 출력 결과에 라는 태그가 있다…



get_fields

zero-base /

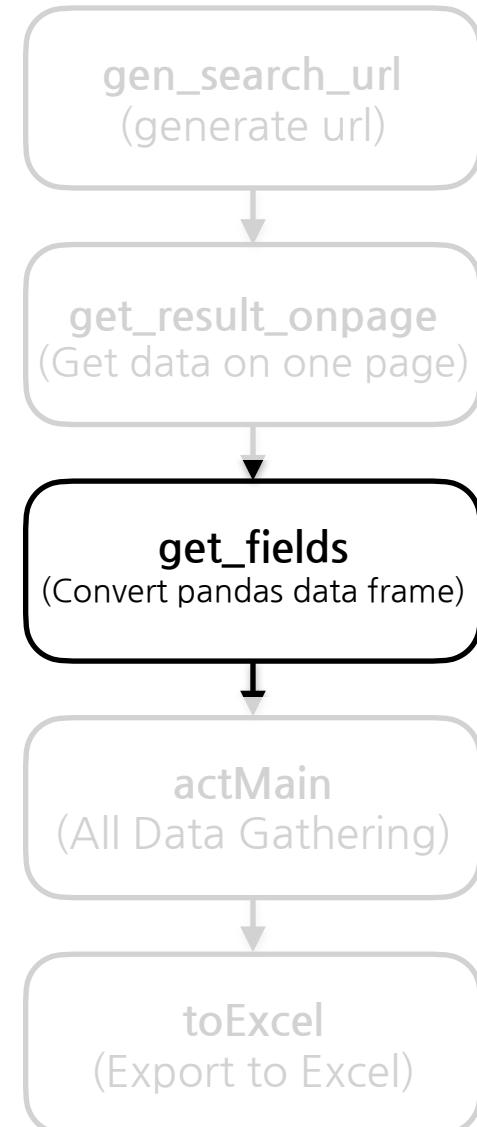
In [103]:

```
1 def delete_tag(input_str):  
2     input_str = input_str.replace("<b>", "")  
3     input_str = input_str.replace("</b>", "")  
4     return input_str
```

- 를 없애는 함수 작성

get_fields

zero-base /

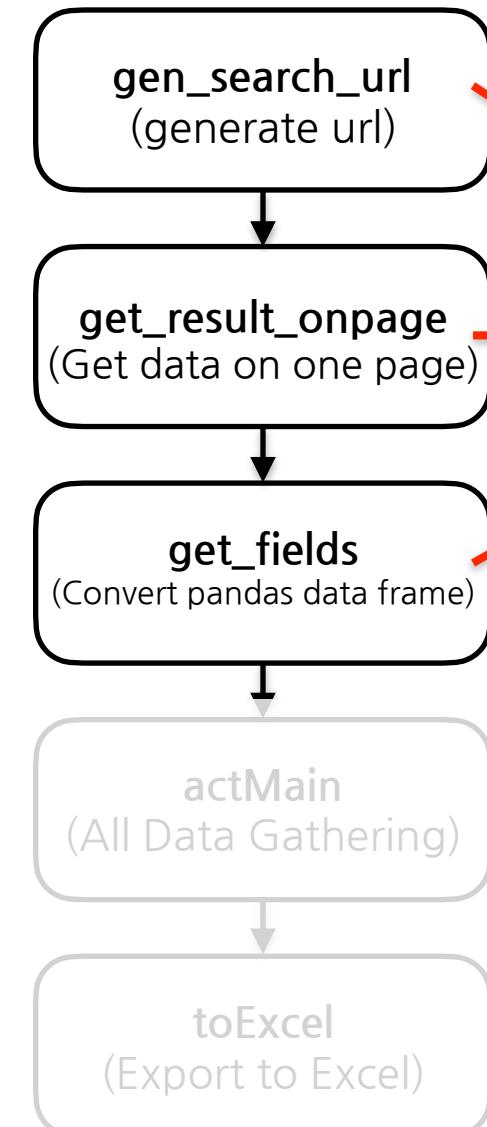


In [21]:

```

1 import pandas as pd
2
3 def get_fields(json_data):
4     title = [delete_tag(each["title"]) for each in json_data['items']]
5     link = [each["link"] for each in json_data['items']]
6     lprice = [each["lprice"] for each in json_data['items']]
7     mall_name = [each["mallName"] for each in json_data['items']]
8
9     result_pd = pd.DataFrame({
10         "title": title,
11         "lprice": lprice,
12         "link": link,
13         "mall": mall_name,
14     }, columns=["title", "lprice", "link", "mall"])
15
16     return result_pd
  
```

- 를 없애는 함수를 title에 적용



```

1 url = gen_search_url("shop", "몰스킨", 1, 5)
2 json_result = get_result_onpage(url)
3 pd_result = get_fields(json_result)
  
```

[2021-09-28 23:19:43.375182] Url Request Success

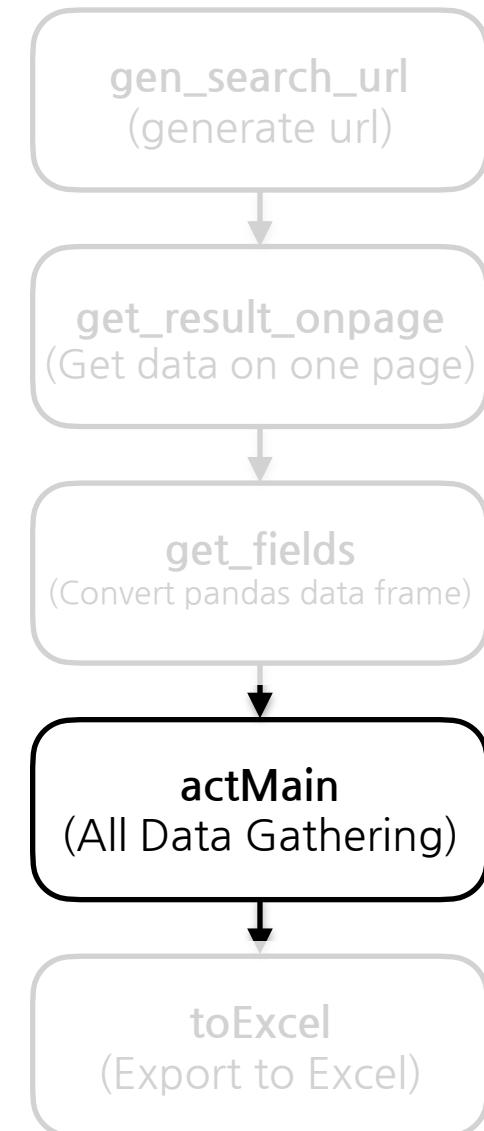
In [23]: 1 pd_result

Out[23]:

		title	lprice	link	mall
0	한정판 몰스킨 2022 해리포터 데일리 다이어리 포켓	31790	https://search.shopping.naver.com/gate.nhn? id=...	네이버	
1	몰스킨 클래식노트 플레인 하드 L	15330	https://search.shopping.naver.com/gate.nhn? id=...	네이버	
2	몰스킨 클래식 노트 소프트커버 도트 포켓 머틀그린	18810	https://search.shopping.naver.com/gate.nhn? id=...	네이버	
3	몰스킨 클래식노트 소프트커버	14630	https://search.shopping.naver.com/gate.nhn? id=...	네이버	
4	몰스킨 2022 다이어리 데일리 위클리 포켓 고급 명품 감성 휴대용 여자 각인 해리포터	24100	https://search.shopping.naver.com/gate.nhn? id=...	에코펜	

actMain

zero-base /



In [33]:

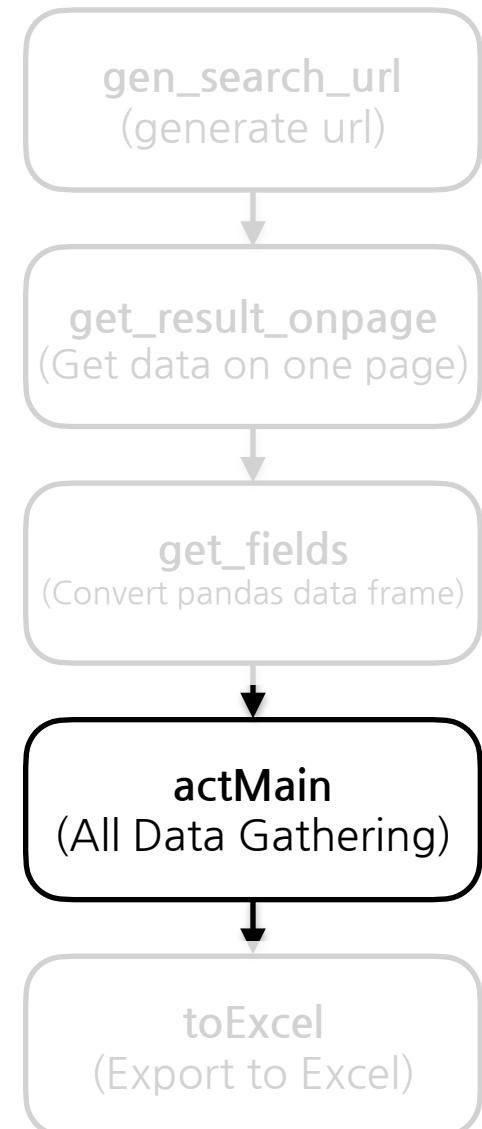
```
1 for n in range(1, 1000, 100):  
2     print(n)
```

```
1  
101  
201  
301  
401  
501  
601  
701  
801  
901
```

- 간단한 for문 테스트

actMain

zero-base /



In [24]:

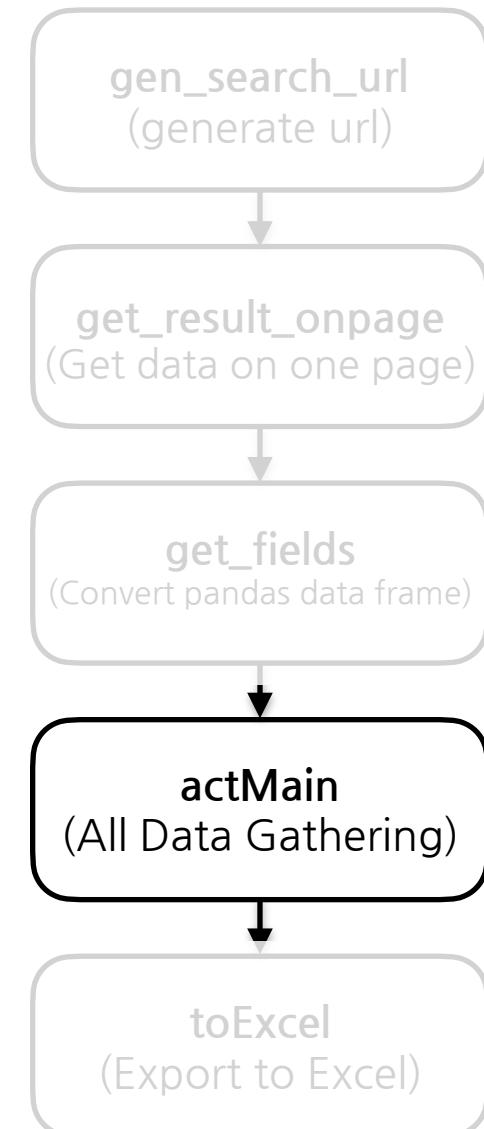
```

1 result_mol = []
2
3 for n in range(1, 1000, 100):
4     url = gen_search_url("shop", "몰스킨", n, 100)
5     json_result = get_result_onpage(url)
6     pd_result = get_fields(json_result)
7
8     result_mol.append(pd_result)
9
10 result_mol = pd.concat(result_mol)
  
```

[2021-09-28 23:19:54.907838] Url Request Success
 [2021-09-28 23:19:55.198471] Url Request Success
 [2021-09-28 23:19:55.505816] Url Request Success
 [2021-09-28 23:19:55.612894] Url Request Success
 [2021-09-28 23:19:55.708310] Url Request Success
 [2021-09-28 23:19:55.798511] Url Request Success
 [2021-09-28 23:19:55.898960] Url Request Success
 [2021-09-28 23:19:56.000708] Url Request Success
 [2021-09-28 23:19:56.099755] Url Request Success
 [2021-09-28 23:19:56.215878] Url Request Success

actMain

zero-base /



In [25]:

1 result_mol.info()

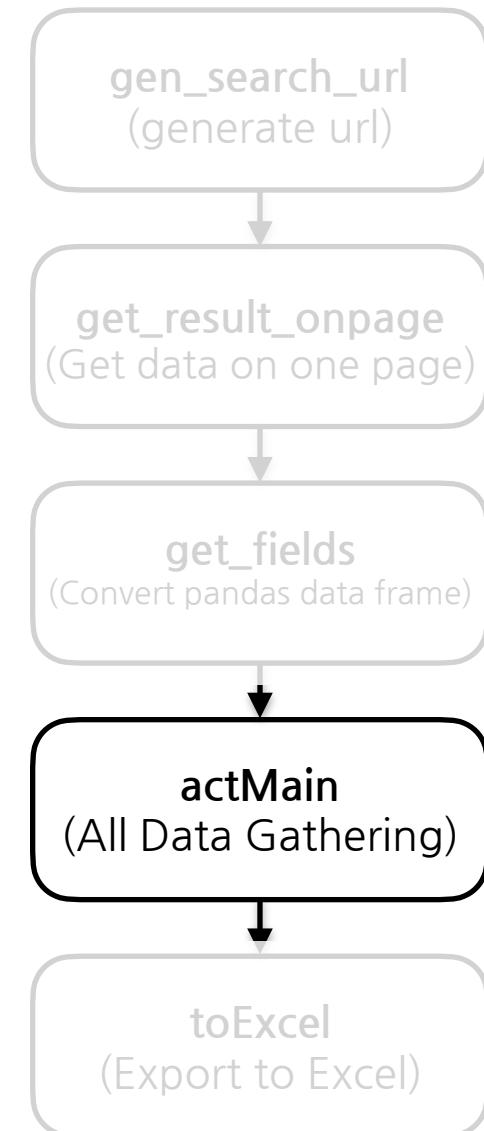
```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000 entries, 0 to 99
Data columns (total 4 columns):
 #   Column   Non-Null Count   Dtype  
--- 
 0   title    1000 non-null    object 
 1   lprice   1000 non-null    object 
 2   link     1000 non-null    object 
 3   mall     1000 non-null    object 
dtypes: object(4)
memory usage: 39.1+ KB
  
```

- 다른 데이터는 괜찮은데…
- index가 이상하다

actMain

zero-base /



In [26]:

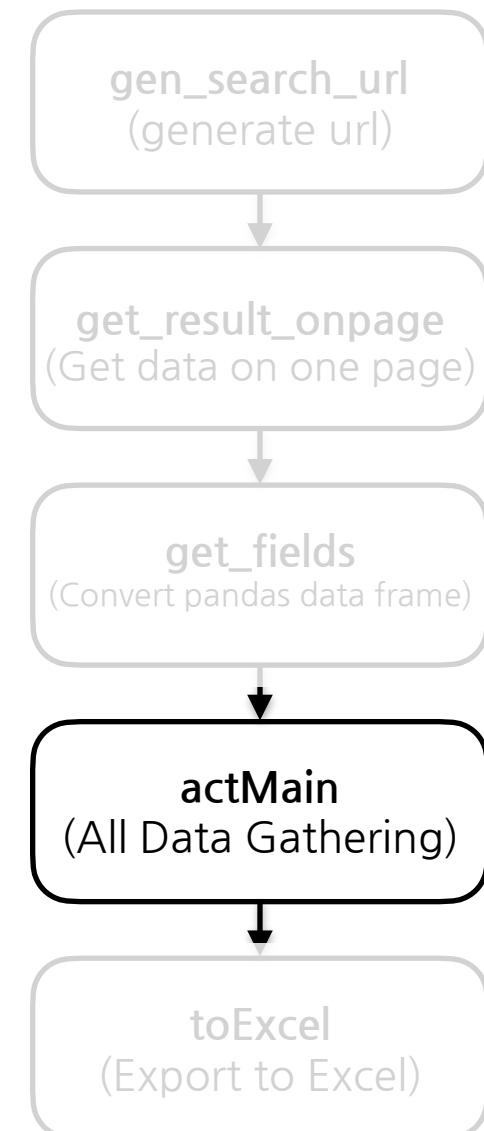
```
1 result_mol.reset_index(drop=True, inplace=True)
2 result_mol.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 4 columns):
 #   Column   Non-Null Count   Dtype  
 ---  --       --           --      
 0   title    1000 non-null    object 
 1   lprice   1000 non-null    object 
 2   link     1000 non-null    object 
 3   mall     1000 non-null    object 
dtypes: object(4)
memory usage: 31.4+ KB
```

• OK~~

actMain

zero-base /



In [26]:

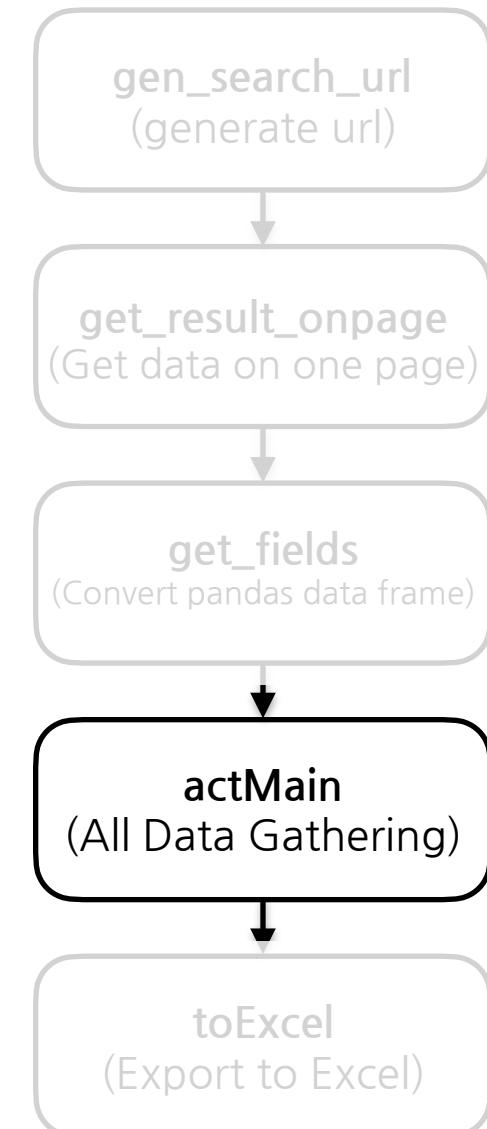
```
1 result_mol.reset_index(drop=True, inplace=True)
2 result_mol.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 4 columns):
 #   Column   Non-Null Count   Dtype  
 ---  --       --           --      
 0   title    1000 non-null    object 
 1   lprice   1000 non-null    object 
 2   link     1000 non-null    object 
 3   mall     1000 non-null    object 
dtypes: object(4)
memory usage: 31.4+ KB
```

• OK~~

actMain

zero-base /



In [34]: 1 result_mol.head()

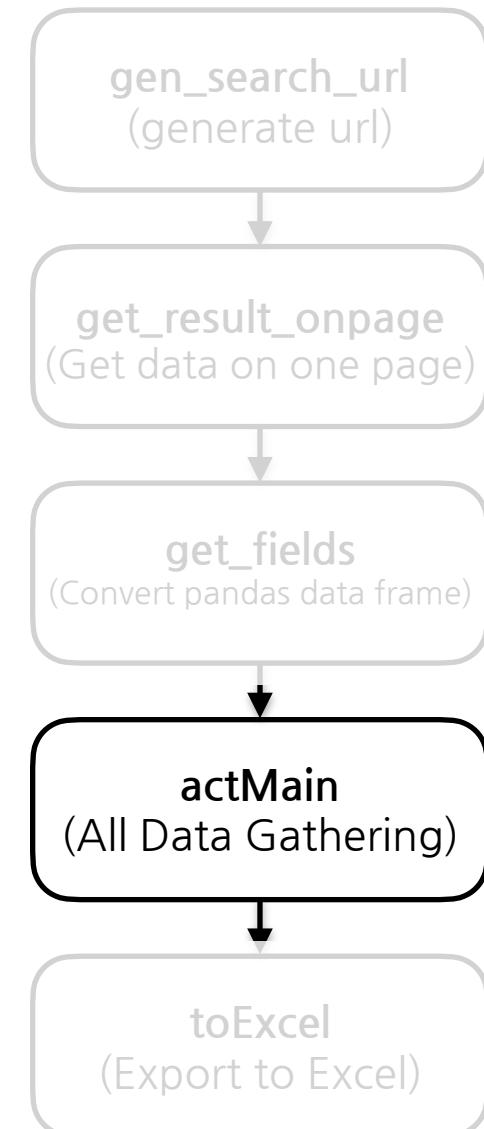
Out[34]:

	title	lprice	link	mall
0	한정판 몰스킨 2022 해리포터 데일리 다이어리 포켓	31790.0	https://search.shopping.naver.com/gate.nhn?...id=...	네이버
1	몰스킨 클래식노트 플레인 하드 L	15330.0	https://search.shopping.naver.com/gate.nhn?...id=...	네이버
2	몰스킨 클래식 노트 소프트커버 도트 포켓 머틀그린	18810.0	https://search.shopping.naver.com/gate.nhn?...id=...	네이버
3	몰스킨 클래식노트 소프트커버	14630.0	https://search.shopping.naver.com/gate.nhn?...id=...	네이버
4	몰스킨 2022 다이어리 데일리 위클리 포켓 고급 명품 감성 휴대용 여자 각인 해리포터	24100.0	https://search.shopping.naver.com/gate.nhn?...id=...	에코펜

- 매우 OK~~

actMain

zero-base /



In [28]:

```

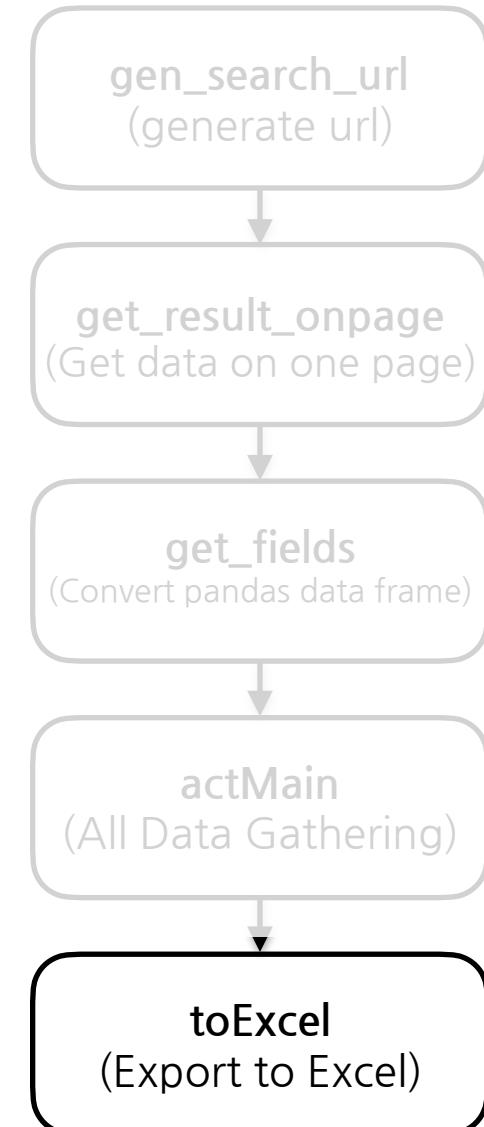
1 result_mol["lprice"] = result_mol["lprice"].astype("float")
2 result_mol.info()
  
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 4 columns):
 #   Column   Non-Null Count   Dtype    
--- 
 0   title    1000 non-null    object   
 1   lprice   1000 non-null    float64  
 2   link     1000 non-null    object   
 3   mall     1000 non-null    object   
dtypes: float64(1), object(3)
memory usage: 31.4+ KB
  
```

- 가격은 숫자형으로~~

toExcel



In [30]:

```

1 writer = pd.ExcelWriter("../data/06_molskin_diary_in_naver_shop.xlsx", engine="xlsxwriter")
2 result_mol.to_excel(writer, sheet_name="Shee1")
3
4 workbook = writer.book
5 worksheet = writer.sheets["Shee1"]
6 worksheet.set_column("A:A", 4)
7 worksheet.set_column("B:B", 60)
8 worksheet.set_column("C:C", 10)
9 worksheet.set_column("D:D", 10)
10 worksheet.set_column("E:E", 50)
11 worksheet.set_column("F:F", 10)
12
13 worksheet.conditional_format("C2:C1001", {"type": "3_color_scale"})
14 writer.save()
  
```

- pip install xlsxwriter
- ds_study/data 폴더에 저장

시각화

In [125]:

```
1 import matplotlib.pyplot as plt
2 import seaborn as sns
3 %matplotlib inline
4
5 from matplotlib import font_manager, rc
6 plt.rcParams['axes.unicode_minus'] = False
7
8 f_path = "/Library/Fonts/Arial Unicode.ttf"
9 font_name = font_manager.FontProperties(fname=f_path).get_name()
10 rc("font", family=font_name)
```

- matplotlib 한글 대응

In [131]:

```
1 plt.figure(figsize=(15, 6))
2 sns.countplot(
3     result_mol["mall"],
4     data=result_mol,
5     palette="RdYlGn",
6     order=result_mol["mall"].value_counts().index
7 )
8 plt.xticks(rotation=90)
9 plt.show()
```

- 몰스킨 노트 판매자 조회

시각화

zero-base /

