

공공데이터 Open API



1. 공공 데이터 Open API Key 발급 받기
2. XML 데이터를 pandas로
3. JSON 데이터를 pandas로

1. 목적

1. 공공데이터에서 Open API 인증키로 데이터 다운받기
2. XML 데이터를 Pandas로 변경하기
3. json 데이터를 Pandas로 변경하기

2.1 공공데이터 사이트_회원가입

<https://www.data.go.kr/> 에 접속하여 '국토교통부 아파트매매 실거래 상세자료' 를 입력하여 검색하면 공공데이터를 활용한 아파트 실거래가 자동 분석이 가능하다.

이 누리집은 대한민국 공식 전자정부 누리집입니다.

로그인 회원가입 사이트맵 ENGLISH

DATA 공공데이터포털 .GO.KR

데이터찾기 국가데이터맵 데이터요청 데이터활용 정보공유 이용안내

국토교통부 아파트매매 실거래 상세자료

인기검색어 ||

3. 기상청

검색조건 분류체계 서비스유형 확장자

① 검색도움말

콘텐츠 바로가기

2.2 공공데이터 자료 탐색

오픈 API(2건) 아래의 첫번째 XML 형태 파일 '국토교통부 아파트매매 실거래 상세자료' 미리보기

**DATA**
공공데이터포털
.GO.KR

데이터찾기

데이터목록

국토교통부 아파트매매 실거래 상세자료

연관

"국토교통부 아

오픈 API (2건)

공공행정

국가행정기관

국가중점

XML

국토교통부_아파트매매 실거래 상

부동산 거래신고에 관한 법률에 따라 신고된 주택의

제공기관 국토교통부 수정일 2023-08-31 2

공공행정

국가행정기관

국가중점

XML

국토교통부_아파트매매 실거래자료

부동산 거래신고에 관한 법률에 따라 신고된 주택의 실거래 자료를 제공

제공기관 국토교통부 수정일 2023-08-31 조회수 72402 활용신청 17457 키워드 주택,아파트,실거래가

활용신청

OpenAPI 미리보기

국토교통부_아파트매매 실거래 상세 자료

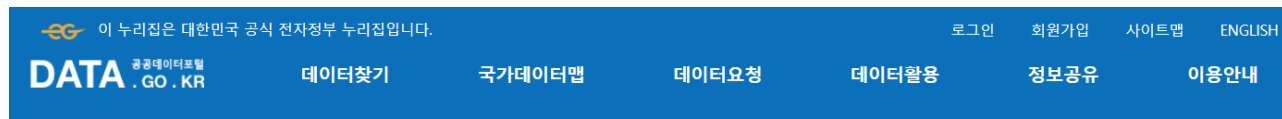
71

API 유형	REST
심의유형	개발단계 : 자동승인 / 운영단계 : 자동승인
비용부과 유무	무료
이용허락범위	이용허락범위 제한 없음
상세기능	국토교통부_아파트매매 상세자료 외 1건
등록일	2017-02-06

닫기

2.3 공공데이터 Open API 신청하기

오픈 API(2건) 아래의 첫번째 XML 형태 파일 '국토교통부 아파트매매 실거래 상세자료'를 활용신청



데이터목록

국토교통부 아파트매매 실거래 상세자료

AND

연관

제공기관별 검색 상세검색

"국토교통부 아파트매매 실거래 상세자료"에 대해 총 2건이 검색되었습니다.

오픈 API (2건)

더보기 >

공공행정	국가행정기관	국가중점	미리보기
XML	국토교통부_아파트매매 실거래 상세 자료		
부동산 거래신고에 관한 법률에 따라 신고된 주택의 실거래 자료를 제공			
제공기관	국토교통부	수정일	2023-08-31
조회수	114479	활용신청	16970
키워드	주택,아파트,실거래가		
공공행정	국가행정기관	국가중점	미리보기
XML	국토교통부_아파트매매 실거래자료		
부동산 거래신고에 관한 법률에 따라 신고된 주택의 실거래 자료를 제공			
제공기관	국토교통부	수정일	2023-08-31
조회수	72402	활용신청	17457
키워드	주택,아파트,실거래가		

2.4 공공데이터 Open API 신청받기

‘마이페이지’에서 일반 인증키(Encoding과 Decoding)을 확인하기

홈

개발계정 상세보기

마이페이지

데이터 활용 >

데이터 요청 >

나의 문의 >

회원정보 수정 >

기본정보

데이터명	국토교통부_아파트매매 실거래 상세 자료 상세설명		
서비스유형	REST	심의여부	자동승인
신청유형	개발계정 연장신청	처리상태	승인
활용기간	2023-07-29 ~ 2025-09-07		

서비스정보

데이터포맷	XML
End Point	http://openapi.molit.go.kr/OpenAPI_ToolInstallPackage/service/rest/RTMSOBSvc/getRTMSDataSvcAptTradeDev?_wadl&type=xml

API 환경 또는 API 호출 조건에 따라 인증키가 적용되는 방식이 다를 수 있습니다.

포털에서 제공되는 **Encoding/Decoding** 된 인증키를 적용하면서 구동되는 키를 사용하시기 바랍니다.

* 향후 포털에서 더 명확한 정보를 제공하기 위해 노력하겠습니다.

일반 인증키 (Encoding)	2mPwVWgxGYg%2BVqrxJrJHypUHRhjDGHNDyrymTEUogQPk1TO9Y%2Fa3PMR aBADnpUCtXidmrKd3C2XTCQ8Gg%3D%3D
일반 인증키 (Decoding)	2mPwVWgxOPGYg/a3PMRaBADnpUCtXidmrKd3C2XTCQ8Gg==

2.5 활용신청 데이터 탐색

활용신청 상세기능정보 '확인'을 누르면 페이지번호(pageNo)당 한 페이지 결과 수(numOfRows)는 10

활용신청 상세기능정보

NO	상세기능	설명	일일 트래픽	미리보기
1	국토교통부_아파트매매 상세자료	지역코드와 기간을 이용하여 해당기간, 해당지역의 아파트 매매 신고자료를 제공하는 아파트 매매 신고 정보 조회	1000	확인

요청변수(Request Parameter)

항목명	샘플데이터	설명
ServiceKey	-	공공데이터포털에서 받은 인증키
pageNo	1	페이지번호
numOfRows	10	한 페이지 결과 수
LAWD_CD	11110	지역코드
DEAL_YMD	201512	계약월

일일 트래픽 1,000줄을 받으려면
총 100페이지를 받으면 된다.

[미리보기](#)

2.5 활용신청 데이터 탐색

요청변수(Request Parameter)의 '미리보기'를 클릭하면

요청변수(Request Parameter)

항목명	샘플데이터	
ServiceKey	-	공공데이터포털에서 받은 인증키
pageNo	1	페이지번호
numOfRows	10	한 페이지 결과 수
LAWD_CD	11110	지역코드
DEAL_YMD	201512	계약월

미리보기

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <header>
    <resultCode>00</resultCode>
    <resultMsg>NORMAL SERVICE.</resultMsg>
  </header>
  <body>
    <items>
      <item>
        <거래금액> 82,500</거래금액>
        <거래유형> </거래유형>
        <건축년도>2008</건축년도>
        <년>2015</년>
        <도로명>사직로8길</도로명>
        <도로명건물본번번호코드>00004</도로명건물본번번호코드>
        <도로명건물부번번호코드>00000</도로명건물부번번호코드>
        <도로명시군구코드>11110</도로명시군구코드>
        <도로명일련번호코드>03</도로명일련번호코드>
        <도로명지상지하코드>0</도로명지상지하코드>
        <도로명코드>4100135</도로명코드>
        <등기일자> </등기일자>
        <법정동> 사직동</법정동>
        <법정동본번코드>0009</법정동본번코드>
        <법정동부번코드>0000</법정동부번코드>
        <법정동시군구코드>11110</법정동시군구코드>
        <법정동읍면동코드>11500</법정동읍면동코드>
        <법정동지번코드>1</법정동지번코드>
        <아파트>광화문스페이스본(101동~105동)</아파트>
        <월>12</월>
        <일>10</일>
        <일련번호>11110-2203</일련번호>
        <전용면적>94.51</전용면적>
        <중개사소재지> </중개사소재지>
        <지번>8</지번>
        <지역코드>11110</지역코드>
        <층>11</층>
        <해제사유발생일> </해제사유발생일>
        <해제여부> </해제여부>
      </item>
    </items>
  </body>
</response>
```


2.5 활용신청 데이터 탐색

상세설명을 클릭하면 REST API 프로토콜 방식으로 데이터 형식(포맷)은 XML, 신청가능 트래픽 1,000

개발계정 상세보기

기본정보

데이터명	국토교통부_아파트매매 실거래 상세 자료	상세설명
서비스유형	REST	심의여부
신청유형	개발계정 연장신청	처리상태
활용기간	2023-07-29 ~ 2025-09-07	

홈

오픈API 상세



XML 국토교통부_아파트매매 실거래 상세 자료

활용신청

부동산 거래신고에 관한 법률에 따라 신고된 주택의 실거래 자료를 제공

71 7 관심

OpenAPI 정보

메타데이터 다운로드

데이터 개선요청

오류신고 및 문의

분류체계	일반공공행정 - 일반행정	제공기관	국토교통부
관리부서명	거래신고관리부	관리부서 전화번호	053-663-8642
API 유형	REST	데이터포맷	XML
활용신청	16970	키워드	주택,아파트,실거래가
등록	2017-02-06	수정	2023-08-31
비용부과유무	무료	신청가능 트래픽	개발계정 : 1,000 / 운영계정 : 활용사례 등록시 신청하면 트래픽 증가 가능

2.5 활용신청 데이터 탐색

맨 아래를 보면 샘플코드가 있음 파이썬을 클릭

요청변수(Request Parameter)

항목명(국문)	항목명(영문)	항목크기	항목구분	샘플데이터	항목설명
서비스키	ServiceKey	20	필수	-	공공데이터포털에서 받은 인증키
페이지 번호	pageNo	4	옵션	1	페이지번호
한 페이지 결과 수	numOfRows	4	옵션	10	한 페이지 결과 수
지역코드	LAWD_CD	5	필수	11110	지역코드
계약월	DEAL_YMD	6	필수	201512	계약월

출력결과(Response Element)

항목명(국문)	항목명(영문)	항목크기	항목구분	샘플데이터	항목설명
결과코드	resultCode	2	필수	00	결과코드
결과메시지	resultMsg	50	필수	OK	결과메시지
한 페이지 결과 수	numOfRows	4	필수	10	한 페이지 결과 수
페이지 번호	pageNo	4	필수	1	페이지번호
전체 결과 수	totalCount	4	필수	3	전체 결과 수

샘플코드

목록

2.6 REST API로 자료 받기

파이썬 샘플코드를 복사하여 인증받은 서비스키를 params의 'serviceKey' 값으로 입력하고 구글 코랩에서 실행

샘플코드

```
Java Javascript C# PHP Curl Objective-C Python Nodejs R

# Python3 샘플 코드 #

import requests

url = 'http://openapi.molit.go.kr/OpenAPI_ToolInstallPackage/service/rest/RTMSOBSvc/getRTMSDataSvcAptTradeDev'
params = {'serviceKey': '서비스키', 'pageNo': '1', 'numOfRows': '10', 'LAWD_CD': '11110', 'DEAL_YMD': '201512' }

response = requests.get(url, params=params)
print(response.content)
```

2mPwVWgxOPGYg/a3PMRaBADnpUCtXidmrKd3C
2XTCQ8Gg==

목록

Python3 샘플 코드

import requests

url = 'http://openapi.molit.go.kr/OpenAPI_ToolInstallPackage/service/rest/RTMSOBSvc/getRTMSDataSvcAptTradeDev'
params = {'serviceKey': '서비스키', 'pageNo': '1', 'numOfRows': '10', 'LAWD_CD': '11110', 'DEAL_YMD': '201512' }

response = requests.get(url, params=params)
print(response.content)

공공데이터 Open API

1. 공공 데이터 Open API Key 발급 받기

2. XML 데이터를 pandas로

3. JSON 데이터를 pandas로

My Drive > bb_파이썬프로그래밍(2) > week-05 ▾



week-05_공공데이터_open_API.ipynb

2.7 Request API

```
import requests

url = 'http://openapi.molit.go.kr/OpenAPI_ToolInstallPackage/service/rest/RTMSOBSvc/getRTMSDataSvcAptTradeDev'

service_key = '2mPwVWgxOPGYg+VqrxJrJHypUHRhjDUogkAQPk1TOPMRaBADXidmrKd3C2XTCQ8Gg=='

params = {'serviceKey' : service_key,
          'pageNo' : '1', 'numOfRows' : '10',
          'LAWD_CD' : '11110', 'DEAL_YMD' : '201512' }

response = requests.get(url, params=params)

print(response.content)
```

response

response.content

2.7 Request API

response

<Response [200]>

response.content

```
b'<?xml version="1.0" encoding="UTF-8" standalone="yes"?><response><header><resultCode>00</resultCode><resultMsg>NORMAL SERVICE.</resultMsg></header><body><items><item>
<#xea#xb1#xb0#xeb#x9e#x98#xea#xb8#x88#xec#x95#xa1>      82,500</#xea#xb1#xb0#xeb#x9e#x98#xea#xb8#x88#xec#x95#xa1><#xea#xb1#xb0#xeb#x9e#x98#xec#x9c#xa0#xed#x98#x95>
</#xea#xb1#xb0#xeb#x9e#x98#xec#x9c#xa0#xed#x98#x95><#xea#xb1#xb4#xec#xb6#x95#xeb#x85#x84#xeb#x8f#x84>2008</#xea#xb1#xb4#xec#xb6#x95#xeb#x85#x84#xeb#x8f#x84>
<#xeb#x85#x84>2015</#xeb#x85#x84><#xeb#x8f#x84#xeb#xa1#x9c#xeb#xaa#x85>#xec#x82#xac#xec#a7#x81#xeb#xa1#x9c#xea#xb8#xb8</#xeb#x8f#x84#xeb#xa1#x9c#xeb#xaa#x85>
<#xeb#x8f#x84#xeb#xa1#x9c#xeb#xaa#x85#xea#xb1#xb4#xeb#xac#xb3#xb8#xeb#xb2#x88#xed#x98#xb8#xec#xbd#x94#xeb#x93#x9c>00004</#xeb#x8f#x84#xeb#xa1#x9c#xeb#xaa#x85#xea#xb1
<#xeb#x8f#x84#xeb#xa1#x9c#xeb#xaa#x85#xea#xb1#xb4#xeb#xac#xb3#xb8#xeb#xb2#x88#xed#x98#xb8#xec#xbd#x94#xeb#x93#x9c>00000</#xeb#x8f#x84#xeb#xa1#x9c#xeb#xaa#x85#xea#xb1
<#xeb#x8f#x84#xeb#xa1#x9c#xeb#xaa#x85#xec#x8b#x9c#xea#xb5#xb0#xea#xb5#xac#xec#xbd#x94#xeb#x93#x9c>11110</#xeb#x8f#x84#xeb#xa1#x9c#xeb#xaa#x85#xec#x8b#x9c#xea#xb5#xb0#xea#xb5
<#xeb#x8f#x84#xeb#xa1#x9c#xeb#xaa#x85#xec#x9d#xb3#xeb#xa0#xa8#xeb#xb2#x88#xed#x98#xb8#xec#xbd#x94#xeb#x93#x9c>03</#xeb#x8f#x84#xeb#xa1#x9c#xeb#xaa#x85#xec#x9d#xb3#xeb#xa0#xa
<#xeb#x8f#x84#xeb#xa1#x9c#xeb#xaa#x85#xec#xa7#x80#xec#x83#x81#xec#a7#x80#xed#x95#x98#xec#xbd#x94#xeb#x93#x9c>0</#xeb#x8f#x84#xeb#xa1#x9c#xeb#xaa#x85#xec#xa7#x80#xec#x83#x81
<#xeb#x8f#x84#xeb#xa1#x9c#xeb#xaa#x85#xec#xbd#x94#xeb#x93#x9c>4100135</#xeb#x8f#x84#xeb#xa1#x9c#xeb#xaa#x85#xec#xbd#x94#xeb#x93#x9c>
<#xeb#x93#xb1#xea#xb8#xb0#xec#x9d#xb3#xec#x9e#x90> </#xeb#x93#xb1#xea#xb8#xb0#xec#x9d#xb3#xec#x9e#x90><#xeb#xb2#x95#xec#xa0#x95#xeb#x8f#x99>
```

```
pip install xmltodict
```

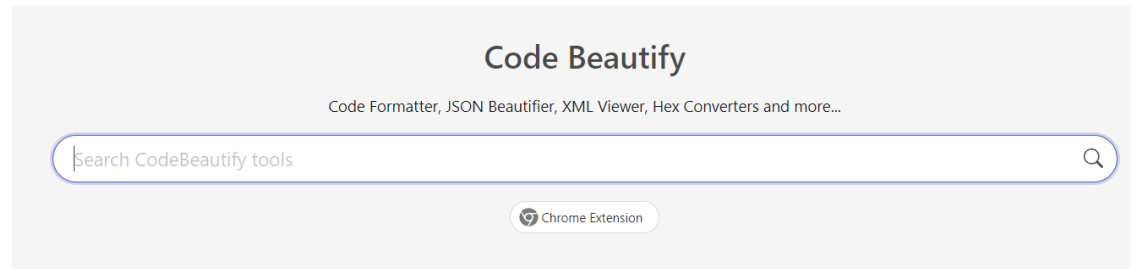
```
import xmltodict
```

```
xmltodict.parse(resp.content.decode('utf-8'))
```

```
{'response': {'header': {'resultCode': '00', 'resultMsg': 'NORMAL SERVICE.'},
  'body': {'items': {'item': [{'거래금액': '82,500',
    '거래유형': 'None',
    '건축년도': '2008',
    '년': '2015',
    '도로명': '사직로8길',
    '도로명건물본번호코드': '00004',
    '도로명건물부번호코드': '00000',
    '도로명시군구코드': '11110',
    '도로명일련번호코드': '03',
    '도로명지번': '11110-03'
  ]}]}}
```

Code Beautify

XML, JSON파일을 미리보기하여 csv
파일로 다운로드 가능한 사이트
(<https://codebeautify.org/>)



Recently used

XML Viewer

Ln: 294 Col: 24 size: 9.09 KB

Tree View

Advertisements

Google 광고

의견 보내기

이 광고가 표시된 이유

Beautify Minify

to JSON to CSV

Download

Output

html ..

body ..

parsererror ..

@style: display: block; white-space: pre; border: 2px solid #c77; padding: 0 1em 0 1em; margin: 1em; background-color: #fdd; color: black

h3 This page contains the following errors:

div error on line 1 at column 1: Document is empty

@style: font-family:monospace;font-size:12px

h3 Below is a rendering of the page up to the first error.

2.7 Request API

```
xmltodict.parse(resp.content.decode('utf-8'))
data = xmltodict.parse(resp.content.decode('utf-8'))
```

xml을 dataframe으로 변경하기

```
data.keys()
data['response'].keys()
data['response']['body']['items']['item']
import pandas as pd
df = pd.DataFrame(data['response']['body']['items']['item'])
df
```

	거래금액	거래유형	건축년도	년	도로명	도로명건물본번호코드	도로명건물부번호코드	도로명시군구코드	도로명일련번호코드	도로명지상지하코드	...	월	일	일련번호	전용면적	중개사소재지	지번	지역코드	층	해제사유발생일	해제여부
0	82,500	None	2008	2015	사직로8길	00004	00000	11110	03	0	...	12	10	11110-2203	94.51	None	9	11110	11	None	None
1	60,000	None	1981	2015	세종대로23길	00047	00000	11110	02	0	...	12	22	11110-12	149.95	None	145	11110	8	None	None
2	130,000	None	2004	2015	경희궁2길	00005	00005	11110	01	0	...	12	8	11110-118	194.43	None	110-15	11110	6	None	None
3	105,000	None	2004	2015	사직로8길	00024	00000	11110	05	0	...	12	14	11110-115	124.17	None	71	11110	8	None	None
4	120,000	None	2003	2015	사직로8길	00020	00000	11110	05	0	...	12	24	11110-107	146.33	None	95	11110	4	None	None
5	17,000	None	2014	2015	대학로	00047	00000	11110	05	0	...	12	17	11110-2359	16.98	None	195-10	11110	8	None	None
6	17,000	None	2014	2015	대학로	00047	00000	11110	05	0	...	12	18	11110-2359	16.98	None	195-10	11110	4	None	None
7	57,000	None	2006	2015	혜화로3가길	00030	00000	11110	01	0	...	12	29	11110-973	106.98	None	19	11110	3	None	None
8	44,000	None	1995	2015	창경궁로	00265	00000	11110	07	0	...	12	1	11110-25	84.8	None	4	11110	18	None	None
9	52,000	None	1995	2015	창경궁로	00265	00000	11110	07	0	...	12	10	11110-25	84.9	None	4	11110	12	None	None

10 rows × 29 columns

실습과제 : 1000개 행을 받기

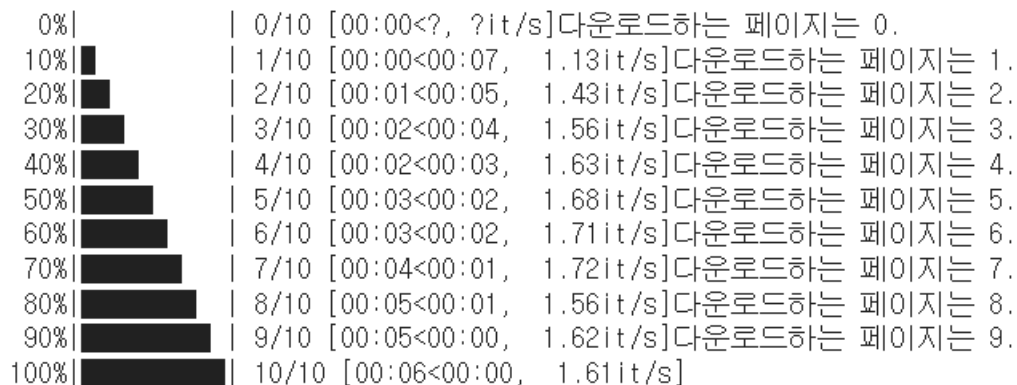
Hint : 3 페이지 받아 보기 : 총 30페이지

```
for page_num in range(3):  
    page = str(page_num)  
    params={'serviceKey' : '당신의 인증키',  
           'pageNo' : page, 'numOfRows' : '10', 'LAWD_CD' : '11110', 'DEAL_YMD' : '201512' }  
    print(params)
```

실습과제 : 1000개 행을 받기

```
from tqdm import tqdm
df = pd.DataFrame()
```

```
df = df.append(tf)
```



Progress	Page	Time	Speed	Message
0%	0/10	[00:00<?, ?it/s]		다운로드하는 페이지는 0.
10%	1/10	[00:00<00:07,	1.13it/s]	다운로드하는 페이지는 1.
20%	2/10	[00:01<00:05,	1.43it/s]	다운로드하는 페이지는 2.
30%	3/10	[00:02<00:04,	1.56it/s]	다운로드하는 페이지는 3.
40%	4/10	[00:02<00:03,	1.63it/s]	다운로드하는 페이지는 4.
50%	5/10	[00:03<00:02,	1.68it/s]	다운로드하는 페이지는 5.
60%	6/10	[00:03<00:02,	1.71it/s]	다운로드하는 페이지는 6.
70%	7/10	[00:04<00:01,	1.72it/s]	다운로드하는 페이지는 7.
80%	8/10	[00:05<00:01,	1.56it/s]	다운로드하는 페이지는 8.
90%	9/10	[00:05<00:00,	1.62it/s]	다운로드하는 페이지는 9.
100%	10/10	[00:06<00:00,	1.61it/s]	

실습과제 : 1000개 행을 받기

```
# 저장하기  
df.to_csv('table.csv', encoding='utf-8-sig')
```

공공데이터 Open API

1. 공공 데이터 Open API Key 발급 받기
2. XML 데이터를 pandas로
3. JSON 데이터를 pandas로

My Drive > bb_파이썬프로그래밍(2) > week-05 ▾



week-05_json2dict-dict2json.ipynb

3.1 json 파일 만들기

- ▼ JSON파일은 사전형의 'str' 형태로 JSON to Dictionary로 변경하려면 - json.loads()

```
json_example = '''
{
    "olivia": {"gender": "female", "age": 25, "hobby": ["reading", "music"]},
    "Tyler": {"gender": "male", "age": 28, "hobby": ["development", "painting"]}
}
'''

print("JSON Type : ", type(json_example))
print("JSON : ", json_example)
```

```
JSON Type : <class 'str'>
JSON :
{
    "olivia": {"gender": "female", "age": 25, "hobby": ["reading", "music"]},
    "Tyler": {"gender": "male", "age": 28, "hobby": ["development", "painting"]}
}
```

```
import json
dict_example = json.loads(json_example) # Convert json to dict
print("JSON Type : ", type(dict_example))
print("JSON : ", dict_example)
```

```
JSON Type : <class 'dict'>
JSON : {'olivia': {'gender': 'female', 'age': 25, 'hobby': ['reading', 'music']}, 'Tyler': {'gender': 'male', 'age': 28, 'hobby': ['development', 'painting']}}
```

3.2 Convert json, dict to dataframe

1. Convert json to dataframe

```
[5] import pandas as pd
```

```
[6] df = pd.read_json(json_example)
     df
```

	olivia	Tyler
gender	female	male
age	25	28
hobby	[reading, music]	[development, painting]

2. convert dict to dataframe

```
[7] df = pd.DataFrame.from_dict(dict_example)
     df
```

	olivia	Tyler
gender	female	male
age	25	28
hobby	[reading, music]	[development, painting]

```
df = pd.DataFrame(dict_example)
df
```

	olivia	Tyler
gender	female	male
age	25	28
hobby	[reading, music]	[development, painting]

3.3 Convert dataframe to json

```
json_output = df.to_json()
print(json_output)
print(type(json_output))
```

```
{"olivia":{"gender":"female","age":25,"hobby":["reading","music"]},"Tyler":{"gender":"male","age":28,"hobby":["development","painting"]}}
```

<class 'str'>

```
json.loads(json_output)
```

```
{'olivia': {'gender': 'female', 'age': 25, 'hobby': ['reading', 'music']},  
 'Tyler': {'gender': 'male', 'age': 28, 'hobby': ['development', 'painting']}}
```

```
print(type(json.loads(json_output)))
```

<class 'dict'>

Convert DataFrame to JSON

```
json_output = df.to_json(orient="records") # 행 단위로 만들기
print(json_output)
print(type(json_output))
```

```
[{"olivia":"female","Tyler":"male"}, {"olivia":25,"Tyler":28}, {"olivia":["reading","music"],"Tyler":["development","painting"]}]
```

<class 'str'>

3.4 Convert dataframe to dict(1/3)

df

	olivia	Tyler
gender	female	male
age	25	28
hobby	[reading, music]	[development, painting]

```
# Access Olivia's gender
```

```
olivia_gender = df["olivia"]["gender"]
```

```
olivia_gender
```

```
# Access Tyler's age
```

```
tyler_age = df["Tyler"]["age"]
```

```
tyler_age
```

```
# Access Tyler's hobbies
```

```
tyler_hobbies = df["Tyler"]["hobby"]
```

```
tyler_hobbies
```


3.4 Convert dataframe to dict(2/3)

```
for index, row in df.iterrows():
    print(f'index :---> {index}')
    print(f'row :---> {row}')
    print('*'*100)
```

```
index :---> gender
row :---> olivia    female
Tyler        male
Name: gender, dtype: object
*****
index :---> age
row :---> olivia    25
Tyler        28
Name: age, dtype: object
*****
index :---> hobby
row :---> olivia    [reading, music]
Tyler    [development, painting]
Name: hobby, dtype: object
*****
```

df.T	gender	age	hobby
olivia	female	25	[reading, music]
Tyler	male	28	[development, painting]

```
# Transpose the data frame and convert it to a dictionary
data_dict = df.T.to_dict(orient='index')
print(data_dict)
```

```
{'olivia': {'gender': 'female', 'age': 25, 'hobby': ['reading', 'music']}, 'Tyler': {'gender': 'male', 'age': 28, 'hobby': ['development', 'painting']}}
```

3.4 Convert dataframe to dict(3/3)

```
df.to_dict(orient='dict')
```

```
{'olivia': {'gender': 'female', 'age': 25, 'hobby': ['reading', 'music']},  
 'Tyler': {'gender': 'male', 'age': 28, 'hobby': ['development', 'painting']}}
```

실습 과제 : 데이터프레임을 사전형으로 변환하기

DB에 있는 아래 데이터프레임(테이블)을 HTTP 프로토콜 get요청이 있을 경우 json(dict)파일로 변경하여 전송하려고 한다. 테이블의 자료를 사전형으로 변경하라

```
pip install faker
from faker import Faker
fake = Faker('tr_TR')
```

	name	phone #	city
0	Okt. Maşide Elamiye Korutürk Şener	0469-672 0965	Andiçside
1	Coşkun Arpağ Seven Kısakürek	06848960764	Port Denkfelfurt
2	Öğr. Sayın Akmaner Korutürk	068 3 673	Aliseydiland
3	Yoruç Yüksel	+90(586)782-0539x5873	Dilfezaborough
4	Yağızkurt Mansuralı Bilir Mansız	0344-228 8857	Gülkadinhaven
...
95	Melül Nevsale Bilge Hayrioğlu	(156)024 2596	Şalimouth
96	Gülhisar Elvan Alemdar	+90 (415) 4839093	Fahrullahside
97	Eral Hudavent Aksu Gülen	+90(665)219-9620x9935	New Öncel
98	Kalgay Ayşan Güçlü Demirel	00423900737	Soykanchester
99	Çağlar Necva Şener	(743)271 0140	Port Tekecan

100 rows × 3 columns

```
{0: {'name': 'Prof. Kay ıt Akçay',
      'phone #': '+90(719)611-6609x8341',
      'city': 'New Solmabury'},
 1: {'name': 'Şerman Vecide Tarhan Alemdar',
      'phone #': '+90 (049) 8800985',
      'city': 'North Gökçil'},
 2: {'name': 'Şeyda Azade Mans ız Aslan',
      'phone #': '+90(525)3962482',
      'city': 'Mans ızland'},
 3: {'name': 'Mishat Aksu',
      'phone #': '+90(755)7695510',
      'city': 'Lake Sidarchester'},
 4: {'name': 'Acarkan Akça', 'phone #': '116 9 636', 'city': 'Duranville'},
 5: {'name': 'Mücellib Ertaş',
      'phone #': '(410)155 3853',
      'city': 'North Hanbikenfort'},
 ...,
 94: {'name': 'Canfeza Aysevrim Demirel',
      'phone #': '+90(183)424-4316x213',
      'city': 'Lake Oganer'},
 95: {'name': 'Teber İteriş Çamurcuoğlu',
      'phone #': '0359-472 2094',
      'city': 'Safuramouth'},
 96: {'name': 'Ayn ımah Sanavber Bilir',
      'phone #': '+90(625)7218043',
      'city': 'South Nurbanuport'},
 97: {'name': 'Akver Arsoy',
      'phone #': '+90 (126) 1773131',
      'city': 'South Nafii'},
 98: {'name': 'Öğr. Özlem Akdeniz Şener',
      'phone #': '0339 954 98 79',
      'city': 'Gölchester'},
 99: {'name': 'Mükramin Fehim Ertaş',
      'phone #': '(823)637 0595',
      'city': 'South Kelâmiview'}}
```