



JSON 문법 학습



JSON 소개



JSON (JavaScript Object Notation) 이란

- 경량 DATA 교환 형식
- 사람이 읽고 쓰기에 용이 하며, 기계가 분석하고 생성하기도 용이하다.

JSON 소개



JSON 특징과 장점

- 텍스트 기반의 가벼운 데이터 형식이다.
- 계층적 구조 표현이 가능하다.
- JavaScript에서 파생되었지만, 프로그래밍언어에 독립적이다.
- Python, C-family, Java, JavaScript등 대부분의 프로그래밍 언어에서 지원하는 데이터 형식이다.
- 데이터 분석, 웹 개발, 소프트웨어 엔지니어링등 다양한 분야에서 데이터 교환에 사용된다.

JSON 기본 문법



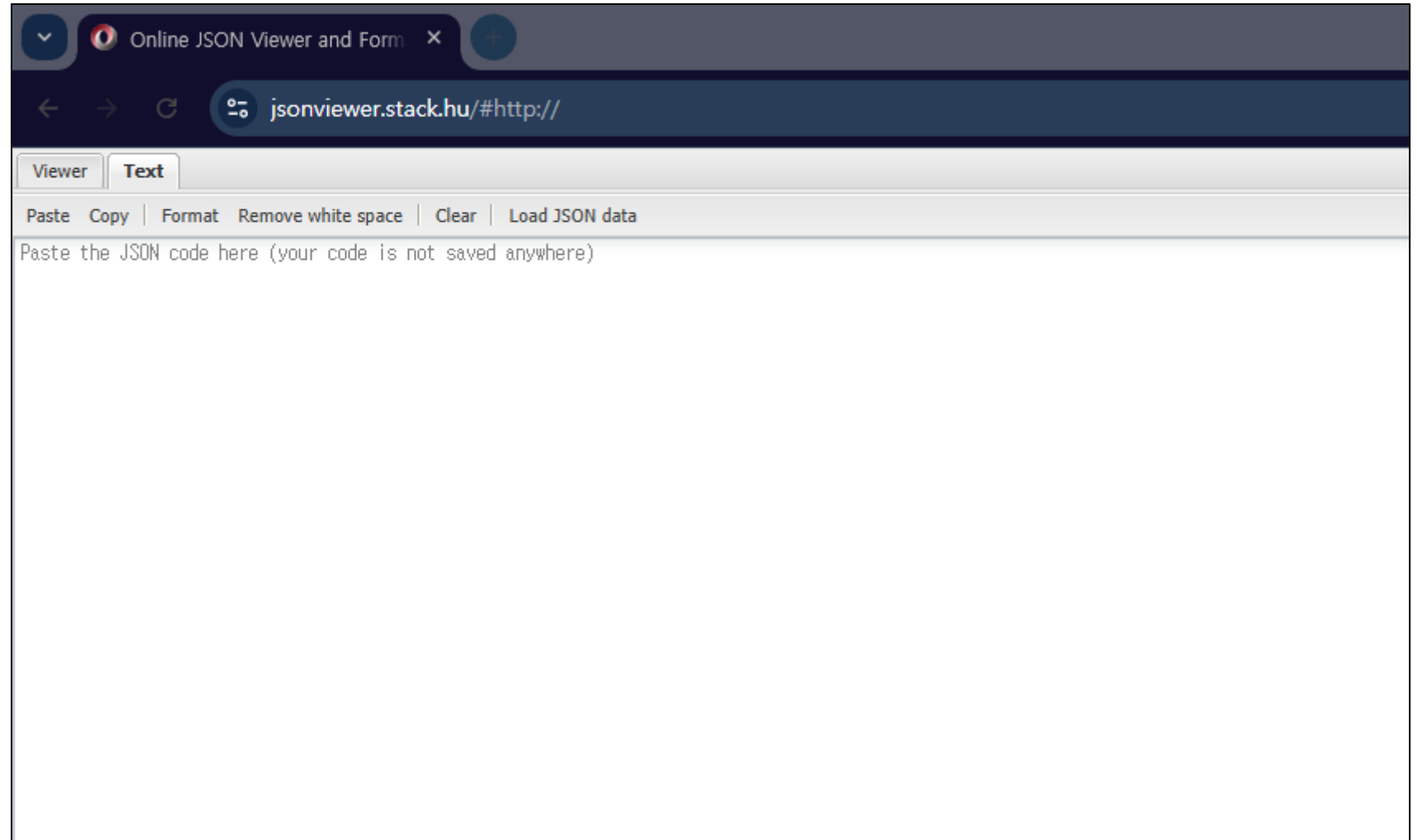
JSON 연습 사이트를 열자

유효성검사

- <https://jsonlint.com/>

json viewer

- <http://jsonviewer.stack.hu/>



JSON 기본 문법



Online JSON View 사용법

- Text 탭 : JSON 문법 작성
- Viewer 탭 : JSON 시각화

```
{
  "이름": "민코딩",
  "나이": 25,
  "키": 180,
  "몸무게": 77,
  "연락처": "010.0000.0000"
}
```

Name	Value
이름	"민코딩"
키	180
나이	25
몸무게	77
연락처	"010.0000.0000"

- Format 클릭 시 : 들여쓰기, 줄바꿈 등 Format 맞춰준다

```
{ "이름": "민코딩",
  "나이": 25, "키": 180, "몸무게": 77, "연락처": "010.0000.0000" }
```

```
{
  "이름": "민코딩",
  "나이": 25,
  "키": 180,
  "몸무게": 77,
  "연락처": "010.0000.0000"
}
```

JSON 기본 문법



JSON 의 최상위 구조

- JSON 데이터는 다음 두가지 형태중 하나로 시작 할 수 있다.
- 1. 객체(Object) : 중괄호{} 사용
- 2. 배열(Array) : 대괄호[] 사용

<pre>{ "name": "민코딩", "age": 25, "phone": "010.0000.1234" }</pre>	<div><div>ViewerText</div><div><div>JSON</div><div><div>name : "민코딩"</div><div>age : 25</div><div>phone : "010.0000.1234"</div></div></div><div><table><thead><tr><th>Name</th><th>Value</th></tr></thead><tbody><tr><td>phone</td><td>"010.0000.1234"</td></tr><tr><td>name</td><td>"민코딩"</td></tr><tr><td>age</td><td>25</td></tr></tbody></table></div></div>	Name	Value	phone	"010.0000.1234"	name	"민코딩"	age	25
Name	Value								
phone	"010.0000.1234"								
name	"민코딩"								
age	25								
<pre>["사과", "바나나", "오렌지"]</pre>	<div><div>ViewerText</div><div><div>JSON</div><div><div>0 : "사과"</div><div>1 : "바나나"</div><div>2 : "오렌지"</div></div></div><div><table><thead><tr><th>Name</th><th>Value</th></tr></thead><tbody><tr><td>2</td><td>"오렌지"</td></tr><tr><td>1</td><td>"바나나"</td></tr><tr><td>0</td><td>"사과"</td></tr></tbody></table></div></div>	Name	Value	2	"오렌지"	1	"바나나"	0	"사과"
Name	Value								
2	"오렌지"								
1	"바나나"								
0	"사과"								

JSON 기본 문법



객체(Object)의 구조와 표현

- 중괄호({})안에 구조가 담긴다.
- 가장 기본적인 구조는 key - value 쌍 이다.

```
{  
    "key" : value  
}
```

- Key : 반드시 문자열이어야 하며, 큰따옴표(“ ”)로 감싸야 한다.
- key는 한 객체 내에서 중복 될 수 없다.
- Value : 다양한 데이터 타입이 올 수 있다.
- 구분자 : 콜론(:)을 사용하여 key와 value를 구분한다.
- 여러 쌍은 콤마(,)로 구분한다.

```
{  
    "key1" : value1,  
    "key2" : value2,  
    "Key3" : value3  
}
```

Viewer Text

Paste Copy Format Remove white sp

```
{  
  "name": "민코딩",  
  "age": 25,  
  "phone": "010.0000.1234"  
}
```

Viewer Text

JSON	
name	민코딩
age	25
phone	010.0000.1234

Name	Value
phone	"010.0000.1234"
name	"민코딩"
age	25

JSON 기본 문법



배열(Array)의 구조와 표현

- 대괄호([])안에 구조가 담긴다.
- 값(value)들의 목록이다.
- 콤마(,)로 값 구분한다.
- 다양한 타입의 값을 섞어서 사용 가능하다.

```
[  
    value1,  
    "value2",  
    "value3",  
    value4  
]
```

Viewer Text

Paste Copy | Format Ren

```
[  
  "안녕하세요.",  
  123,  
  3.141592,  
  true,  
  null  
]
```

Viewer Text	
[] JSON	
0 : "안녕하세요."	Name Value
	4 null
	3 true
	2 3.141592
	1 123
	0 "안녕하세요."

JSON 기본 문법



JSON 값(Value) 데이터 타입

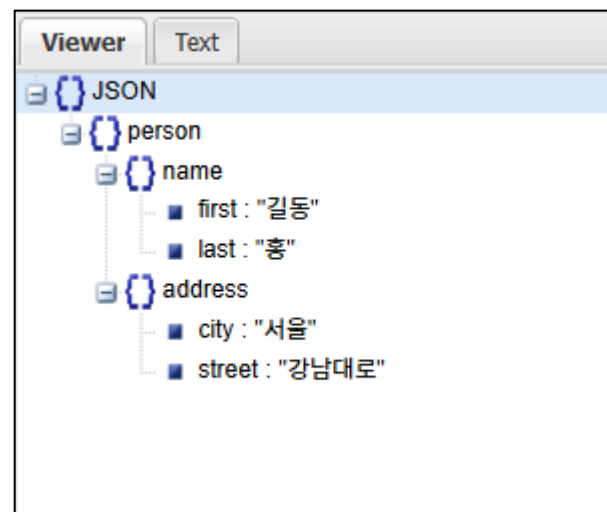
- 문자열(String) : "Hello World"
- 숫자(Number) : 27, 3.14
- 불리언(Boolean) : true, false
- null : null
- 객체(Object): {}
- 배열(Array): []

JSON 기본 문법



중첩구조 - 객체 안의 객체

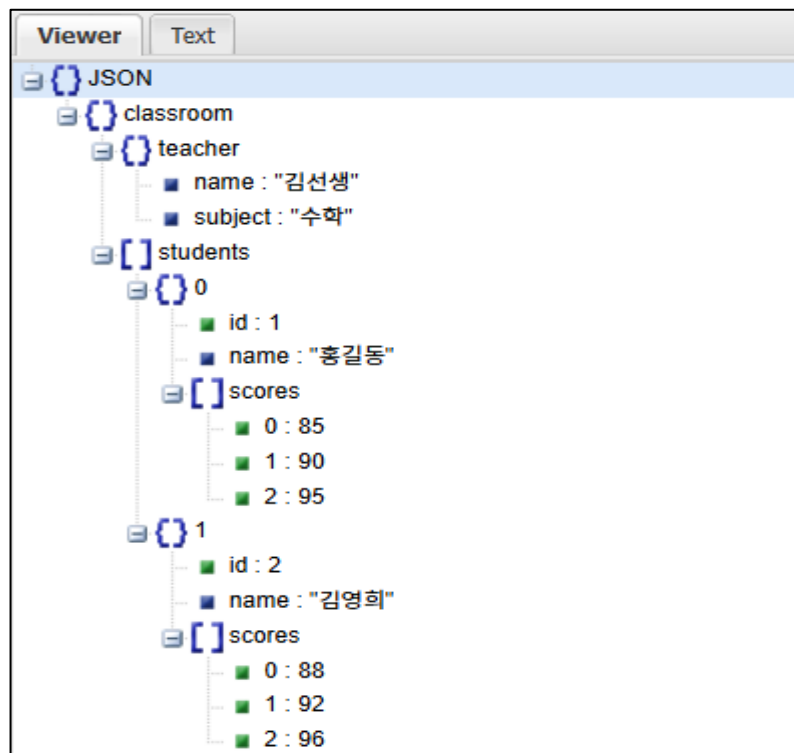
```
Viewer  Text
Paste Copy | Format Remove white space | Cle
{
  "person": {
    "name": {
      "first": "길동",
      "last": "홍"
    },
    "address": {
      "city": "서울",
      "street": "강남대로"
    }
  }
}
```



JSON 기본 문법

중첩구조 - 객체 안의 배열, 배열 안의 객체

```
Viewer  Text
Paste Copy Format Remove white space Clear
{
  "classroom": {
    "teacher": {
      "name": "김선생",
      "subject": "수학"
    },
    "students": [
      {
        "id": 1,
        "name": "홍길동",
        "scores": [85, 90, 95]
      },
      {
        "id": 2,
        "name": "김영희",
        "scores": [88, 92, 96]
      }
    ]
  }
}
```



JSON 기본 문법



JSON 작성시 주의사항

- 키(key)는 반드시 쌍따옴표(“ ”)로 감싸기
- 문자열은 반드시 쌍따옴표(“ ”)로 감싸기
- 숫자, Boolean, null은 따옴표 없어야 함
- 작은따옴표(‘ ’) 사용 불가
- 주석 사용 불가
- 마지막 항목 뒤에 콤마 사용 불가
- 들여쓰기는 가독성을 위한 선택사항

JSON 기본 문법



JSON 참고 사이트

JSON 문법

- <https://jsonlint.com/> : JSON Valid Check
- <http://jsonviewer.stack.hu/> : JSON Visualization

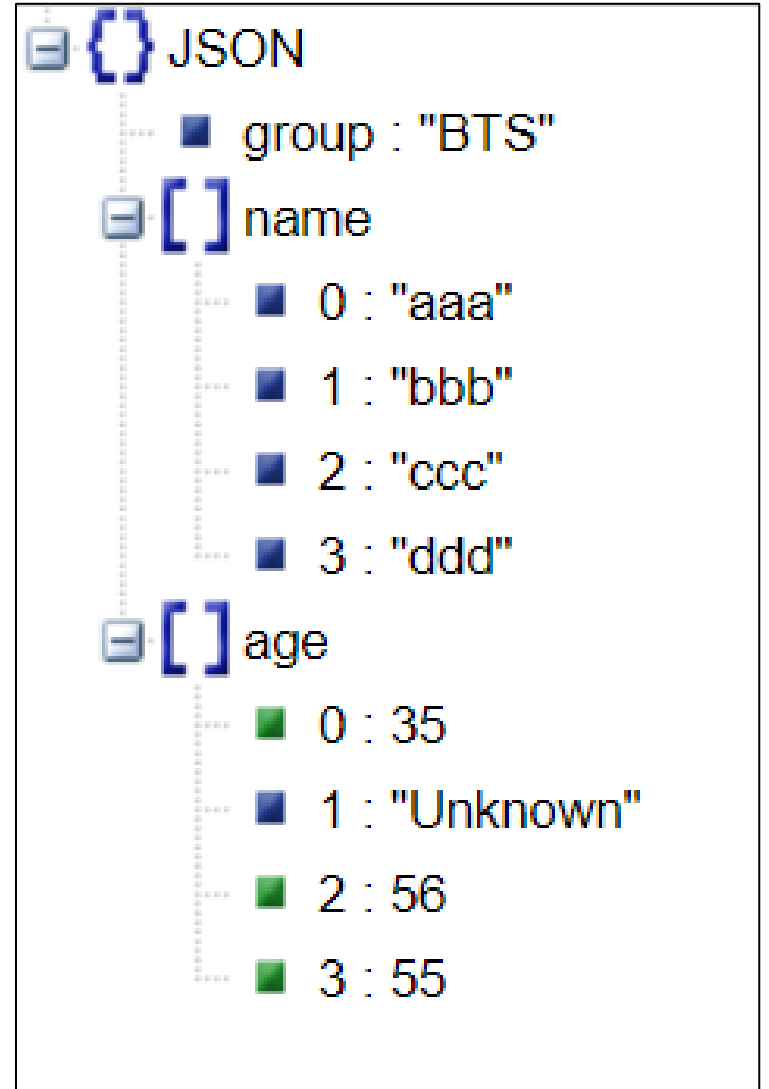
JSON Sample

- <https://json.org/example.html> : 다양한 JSON Sample Data

[도전1] JSON 도전문제

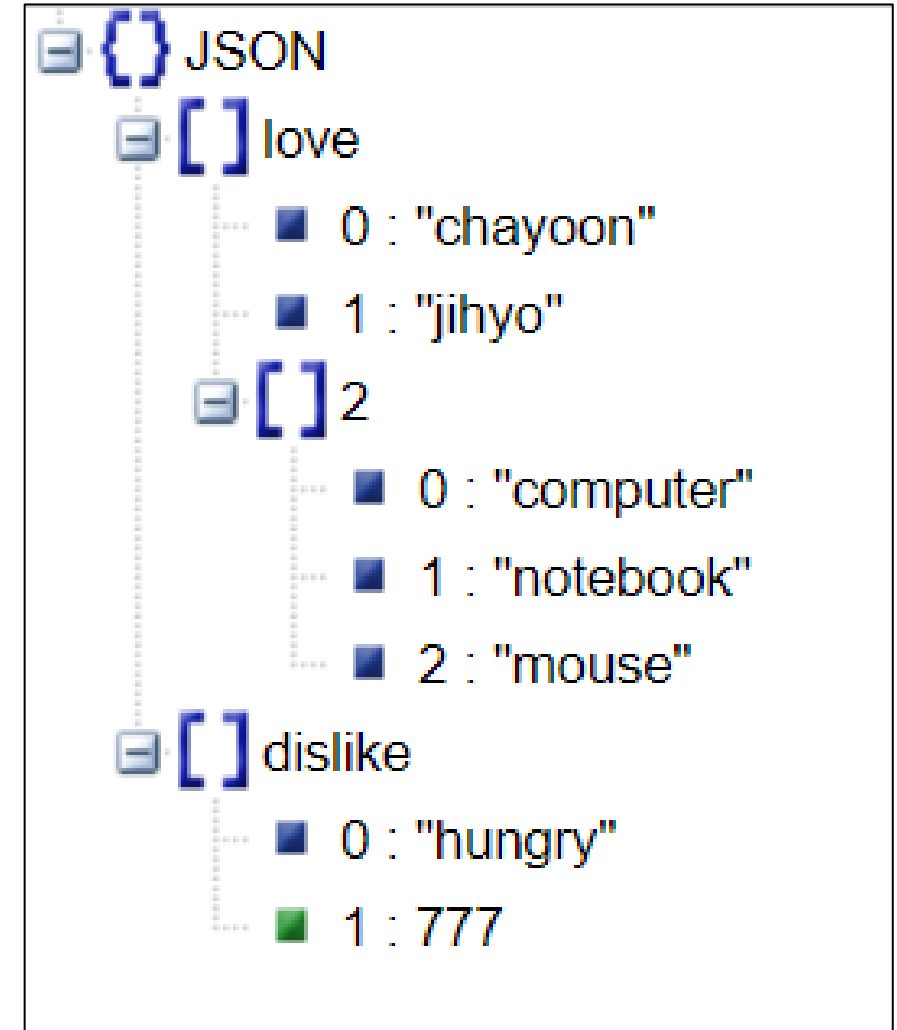


- 오른쪽 결과와 같은 JSON만들기



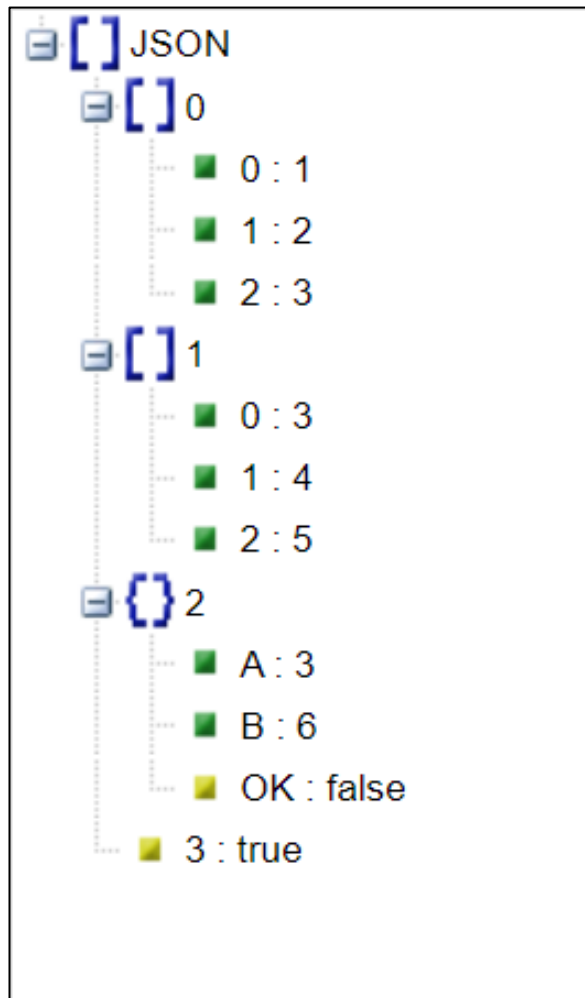
[도전2]JSON 도전문제

- 오른쪽 결과와 같은 JSON만들기



[도전3] JSON 도전문제

- 오른쪽 결과와 같은 JSON만들기





Encoding / Decoding



인코딩과 디코딩



Python 기준에서, Encoding

- Python 객체를 JSON Data로 쓰는 동작

Python 기준에서, Decoding

- 읽은 JSON Data를 Python 객체로 저장한다.

파이썬	→	JSON
dict		오브젝트 (object)
list, tuple		배열(array)
str		문자열(string)
int, float, int와 float에서 파생된 열거형		숫자(number)
True		true
False		false
None		null

Encoding

JSON	→	파이썬
오브젝트(object)		dict
배열(array)		list
문자열(string)		str
숫자 (정수)		int
숫자 (실수)		float
true		True
false		False
null		None

Decoding

json loads



Encoding 메서드

- `json.dumps(dict 객체 이름)`
- dict 객체를 손쉽게 Encoding 가능
- str type 객체를 생성한다.

```
kfc.py x
1  import json
2
3  a = dict()
4  a['name'] = 'sanghi'
5  a['price'] = 4900
6  a['brand'] = 'mcdonald'
7
8  b = json.dumps(a, indent=4)
9  print(b)
```

```
{
    "name": "sanghi",
    "price": 4900,
    "brand": "mcdonald"
}
```

json dumps



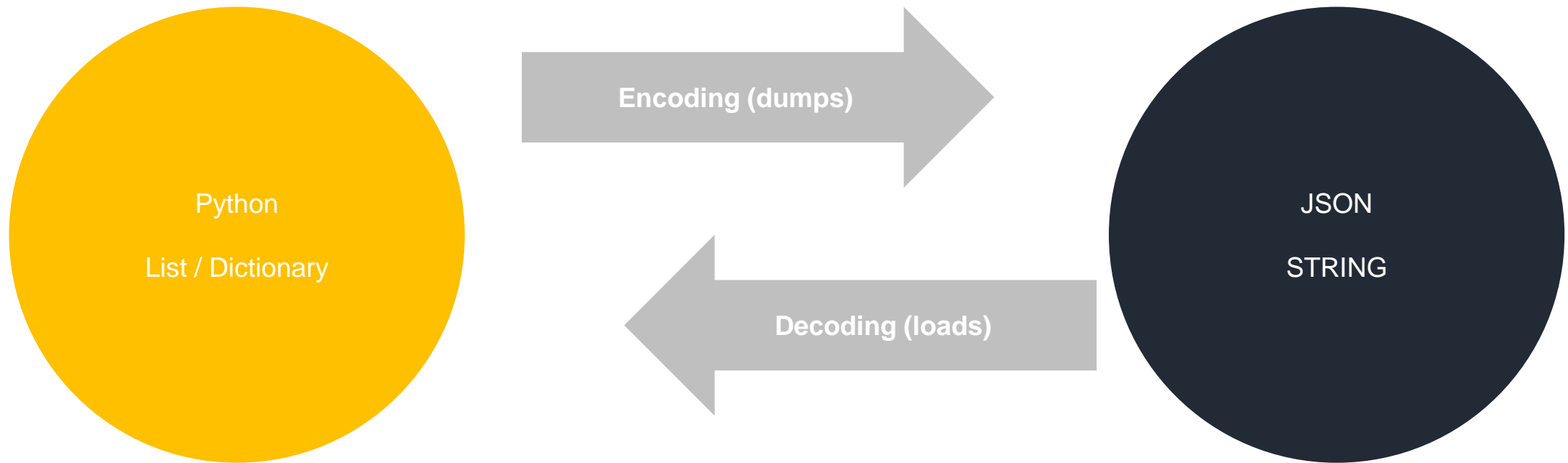
Decoding 메서드

- `a = json.loads(str객체이름)`
- string 을 파싱하여 python dict 객체로 만든다.

```
kfc.py x
1  import json
2
3  a = dict()
4  a['name'] = 'sanghi'
5  a['price'] = 4900
6  a['brand'] = 'mcdonald'
7
8  b = json.dumps(a, indent=4)
9
10 c = json.loads(b)
11 print(c)
```

```
{'name': 'sanghi', 'price': 4900, 'brand': 'mcdonald'}
```

Decoding / Encoding



json 파일 만들기



.json 파일을 생성한다.

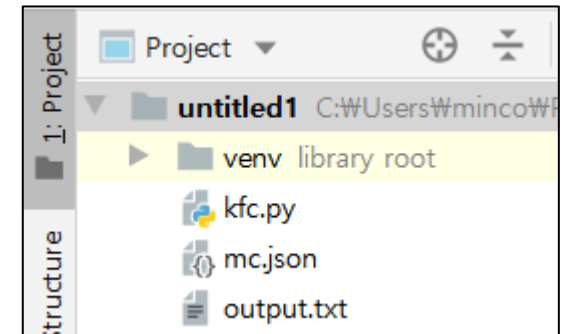
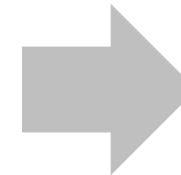
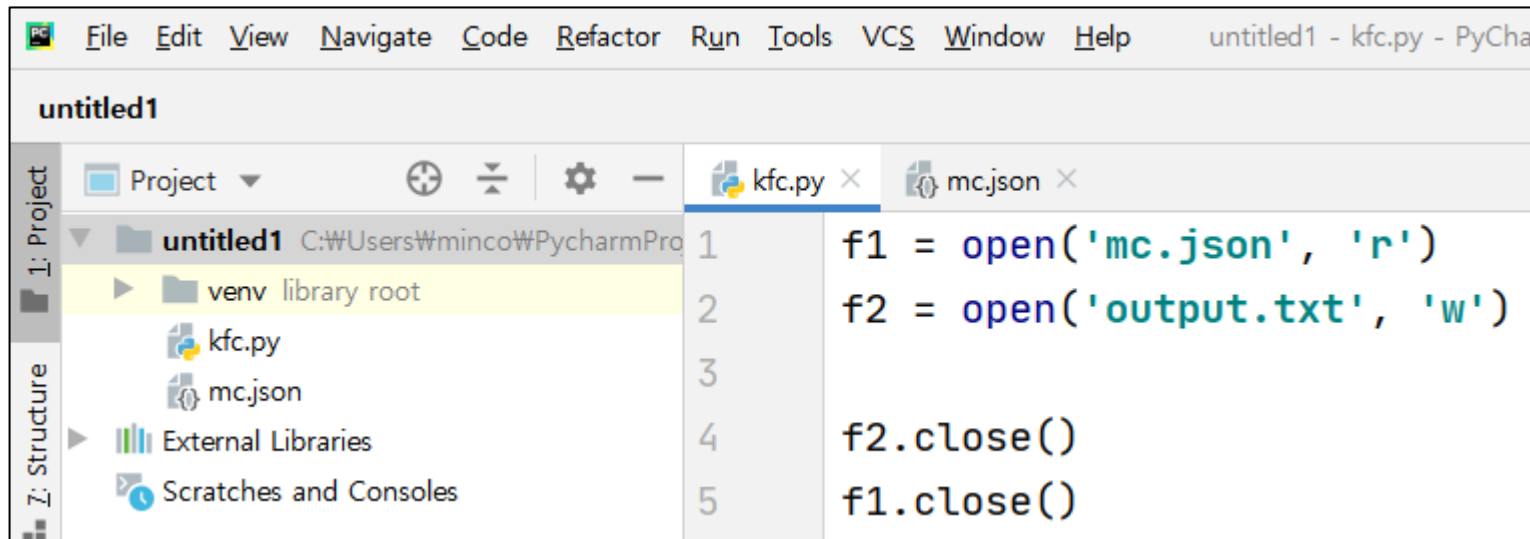
- <https://json.org/example.html> 에서 다음 예제를 복사
- 프로젝트 내 .json 파일을 생성하여 저장한다.

```
1 {"widget": {  
2     "debug": "on",  
3     "window": {  
4         "title": "Sample Konfabulator Widget",  
5         "name": "main_window",  
6         "width": 500,  
7         "height": 500  
8     },  
9     "image": {  
10        "src": "Images/Sun.png",  
11        "name": "sun1",  
12        "hOffset": 250,  
13        "vOffset": 250,
```

json 파일 open

read / write 모드로 file open을 수행

- write open 시 자동으로 프로젝트 내 파일이 추가 됨
- 'a' 모드로 읽으면 마지막 내용을 추가하는 방식

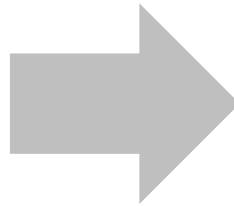


읽은 파일 출력해보기



파일을 전체 읽어서 출력해보자

```
kfc.py x mc.json x
1 f1 = open('mc.json', 'r')
2 f2 = open('output.txt', 'w')
3
4 txt = f1.read()
5 print(txt)
6
7 f2.close()
8 f1.close()
```



```
{"widget": {
  "debug": "on",
  "window": {
    "title": "Sample Konfabulator Widget",
    "name": "main_window",
    "width": 500,
    "height": 500
  },
  "image": {
    "src": "Images/Sun.png",
    "name": "sun1",
    "hoffset": 250,
```

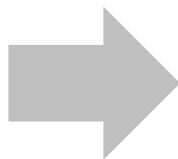

읽은 파일 모두 저장하기



output.txt에 모두 저장해보자

- with 문을 사용하면 자동으로 close 처리가 된다.

```
kfc.py x output.txt x mc.json x
1  f1 = open('mc.json', 'r')
2  f2 = open('output.txt', 'w')
3
4  txt = f1.read()
5  f2.write(txt)
6  print(txt)
7
8  f2.close()
9  f1.close()
10
```



```
kfc.py x output.txt x
1  txt = ""
2
3  with open('mc.json', 'r') as f1:
4      txt = f1.read()
5
6  with open('output.txt', 'w') as f2:
7      f2.write(txt)
8
```

[도전] JSON Decoding 후 출력

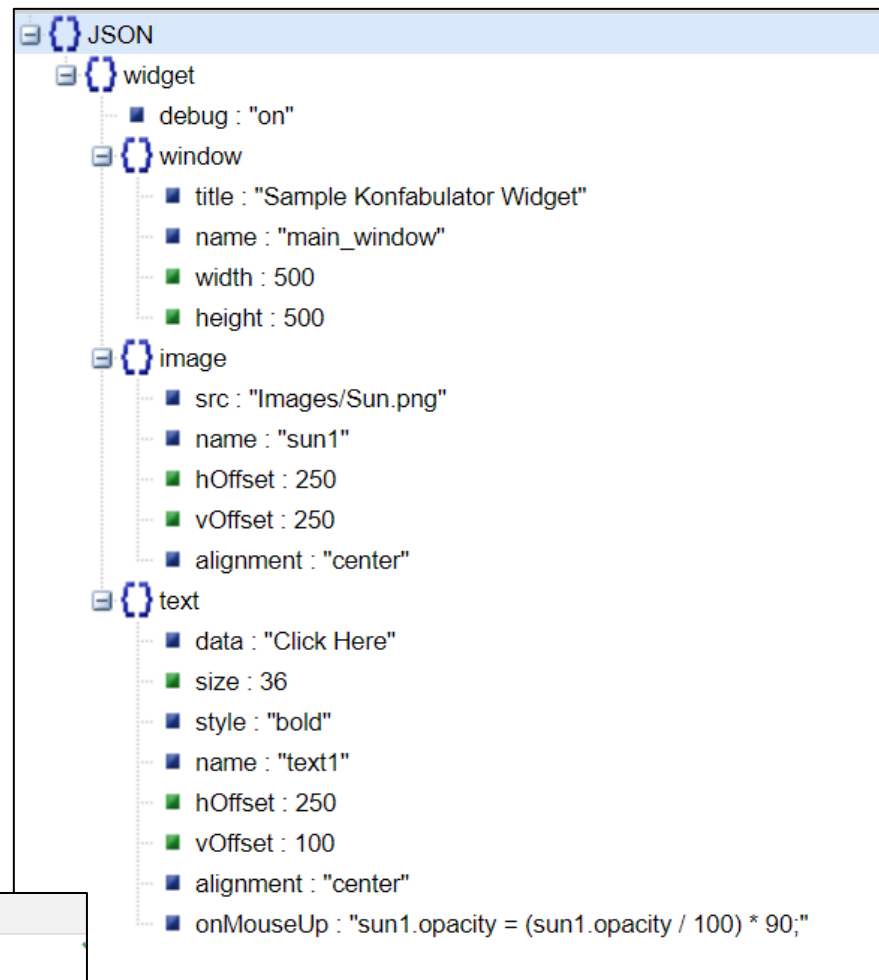
해당하는 요소만 뽑아내기

- 파일 출력을 한다 -> output.txt
- window 객체에서 key만, 모두 파일 출력
- image 객체에서 value만, 모두 파일 출력
- text 객체에서 items 메서드를 사용해서, key, value 세트 모두 출력하기

[참고] Dictionary 메서드

- keys()
- values()
- items()

```
kfc.py x output.txt x
1 title name width height
2 Images/Sun.png sun1 250 250 center
3 ('data', 'Click Here') ('size', 36) ('style', 'bold') ('name', 'text1')
```





JSON 파일 파싱하기



JSON 파일 구조 파악하기



Awesome JSON Datasets

- JSON 파일을 공유하는 사이트 : <https://github.com/jdorman/awesome-json-datasets>
- JSON 구조를 쉽게 파악할 수 있는 사이트 : <http://jsonviewer.stack.hu/>
- TV Shows 에 Netflix 의 Narcos 클릭, URL 을 복사하자

TV Shows

- [Mr. Robot \(USA\)](#)
- [Better Call Saul \(AMC\)](#)
- [Homeland \(Showtime\)](#)
- [Silicon Valley \(HBO\)](#)
- [The Walking Dead \(AMC\)](#)
- [South Park \(Comedy Central\)](#)
- [Game of Thrones \(HBO\)](#)
- [House of Cards \(Netflix\)](#)
- [The Big Bang Theory \(CBS\)](#)
- [Narcos \(Netflix\)](#)
- [Black Mirror \(Netflix\)](#)
- [Stranger Things \(Netflix\)](#)
- [Rick and Morty \(Adult Swim\)](#)
- [Westworld \(HBO\)](#)

JSON 파일 불러오기



JSON 파일을 가져오기

- requests Package 설치
- json 파일 읽어오기

```
kfc.py x  output.txt x
1  import requests
2
3  r = requests.get('주소')
4  print(r.text)
```

```
import json
import requests

url = 'http://api.tvmaze.com/singlesearch/shows?q=westworld&embed=episodes'
r = requests.get(url)
print(r.text)
```



Project 01

파이썬을 활용한 API 데이터 수집



프로젝트 목표



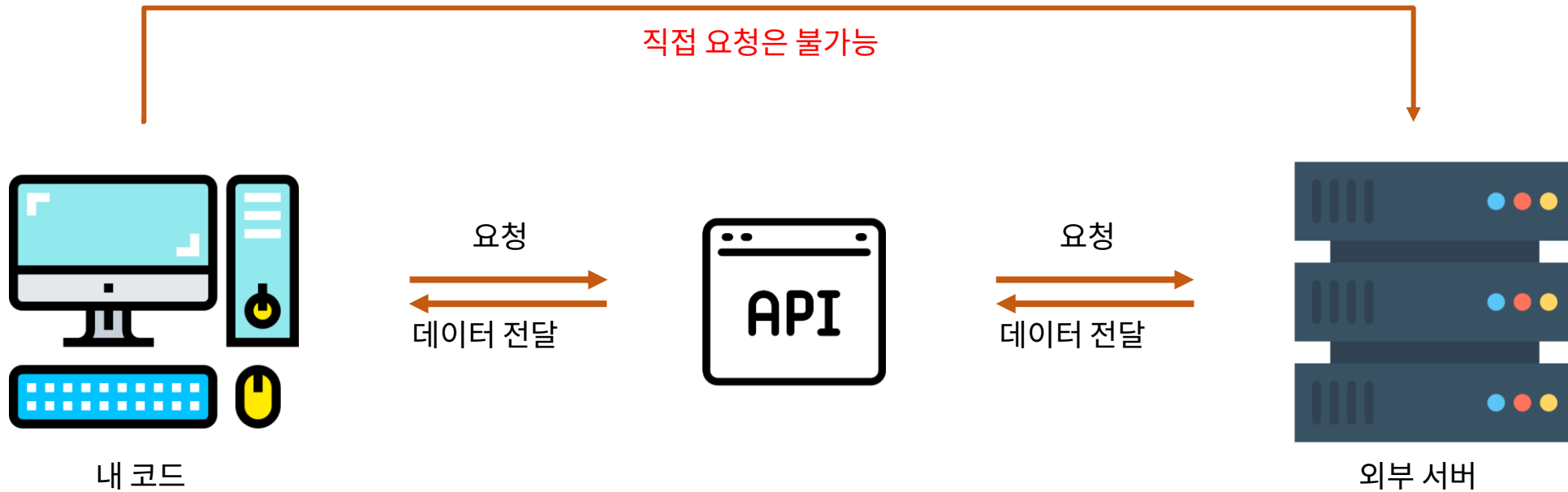
- ✓ Python 기본 문법 및 requests 패키지 사용법
- ✓ API 에 대한 이해
- ✓ JSON 에 대한 이해
- ✓ 실습 - 날씨 정보를 받아와 활용하기
- ✓ 도전 과제 - 금융 상품 정보를 받아와 활용하기

API 이해하기



API

- Application Programming Interface 의 약자
- 서로 다른 애플리케이션끼리 상호작용하고 데이터를 교환하기 위한 규칙과 도구



API 이해하기



OPEN API

- 외부에서 사용할 수 있도록 개방(OPEN) 된 API

프로젝트에서 사용되는 API

- OpenWeatherMap API: 기상 데이터 및 날씨 정보를 제공하는 오픈 API
(<https://openweathermap.org/api>)
- 금융상품통합비교공시 API: 금융감독원에서 제공하는 금융 상품 정보를 제공하는 오픈 API :
(<https://finlife.fss.or.kr/finlife/main/contents.do?menuNo=700029>)

API 이해하기



OPEN API 특징 및 주의 사항

- Open API 는 사용 시 API KEY 를 발급받아야 합니다.
- API KEY: 사용자가 API 를 사용할 수 있는 사용자인지 확인하는 역할
- 일부 API 는 사용량이 제한되어 있습니다.
- 공식 문서의 일일 및 월간 사용량 제한을 반드시 확인하여야 합니다.
- 사용량이 초과될 경우 요금이 청구될 수 있습니다.

API 이해하기



requests package

- 파이썬에서 HTTP 요청을 보낼 수 있는 라이브러리
- 설치

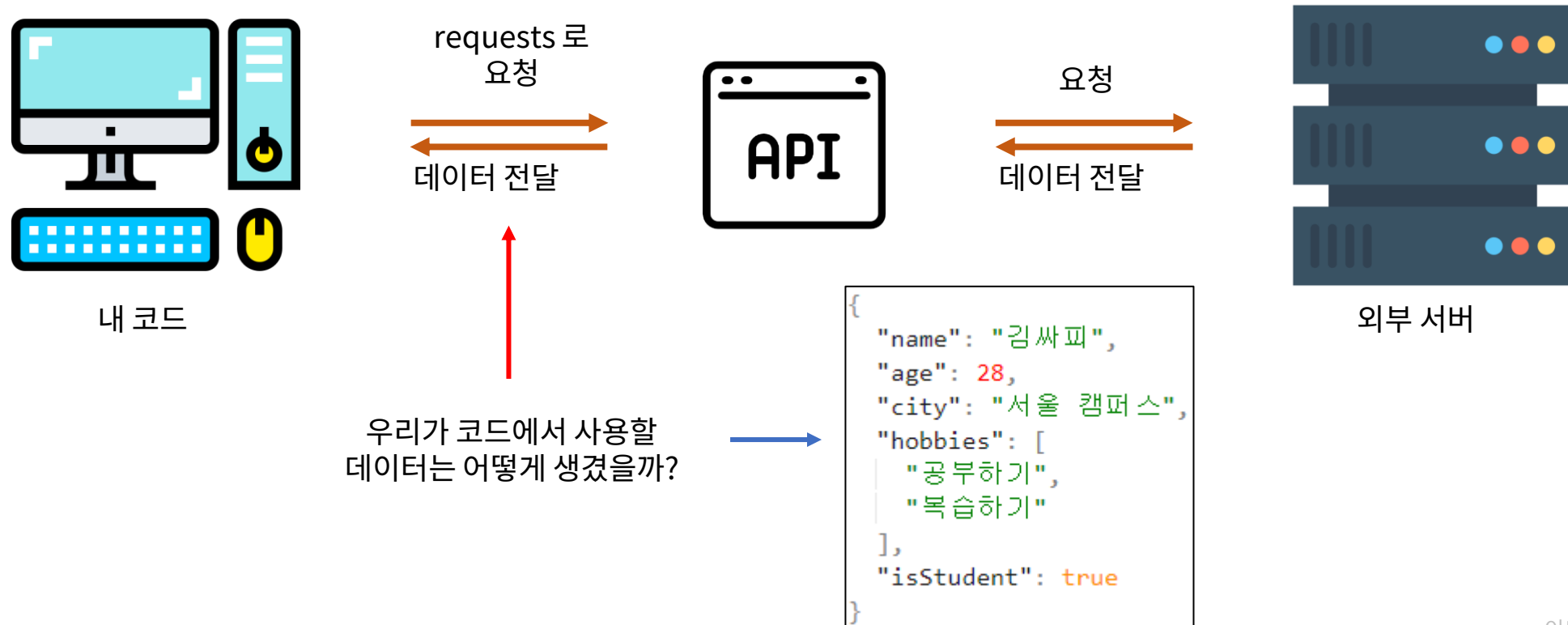
```
$ pip install requests
```

API 이해하기



API 가 사용하는 데이터 형식 - JSON

- API 가 반환하는 데이터는 어떻게 생겼을까?





Openweathermap API

- 기상 데이터 및 날씨 정보를 제공하는 오픈 API
- 전세계 날씨 데이터를 가져와 날씨, 일일 및 시간 별 예보 등 다양한 정보를 얻을 수 있습니다.

API 사용량 제한

- 60 calls/minute
- 1,000,000 calls/month



API 이해하기 - 실습



Openweathermap API - KEY 발급(1/3)

- 사이트 접속 및 회원가입 진행
- <https://openweathermap.org/api>

Members

home.openweathermap.org/users/sign_up

OpenWeather

Weather at your city

Guide API Dashboard Marketplace Pricing Maps Our Initiatives Partners Blog For Business Sign In Support

Create New Account

TEST

myemail@test.test

We will use information you provided for management and administration purposes, and for keeping you informed by mail, telephone, email and SMS of other products and services from us and our partners. You can proactively manage your preferences or opt-out of communications with us at any time using Privacy Centre. You have the right to access your data held by us or to request your data to be deleted. For full details please see the OpenWeather Privacy Policy.

☒ I am 16 years old and over


☒ I agree with Privacy Policy, Terms and conditions of sale and Websites terms and conditions of use

I consent to receive communications from OpenWeather Group of Companies and their partners.

☐ System news (API usage alert, system update, temporary system shutdown, etc)

☐ Product news (change to price, new product features, etc)

☐ Corporate news (our life, the launch of a new service, etc)

☐ 로봇이 아닙니다.  CAPTCHA
가장 쉽게 보충 - 1. 2. 3.

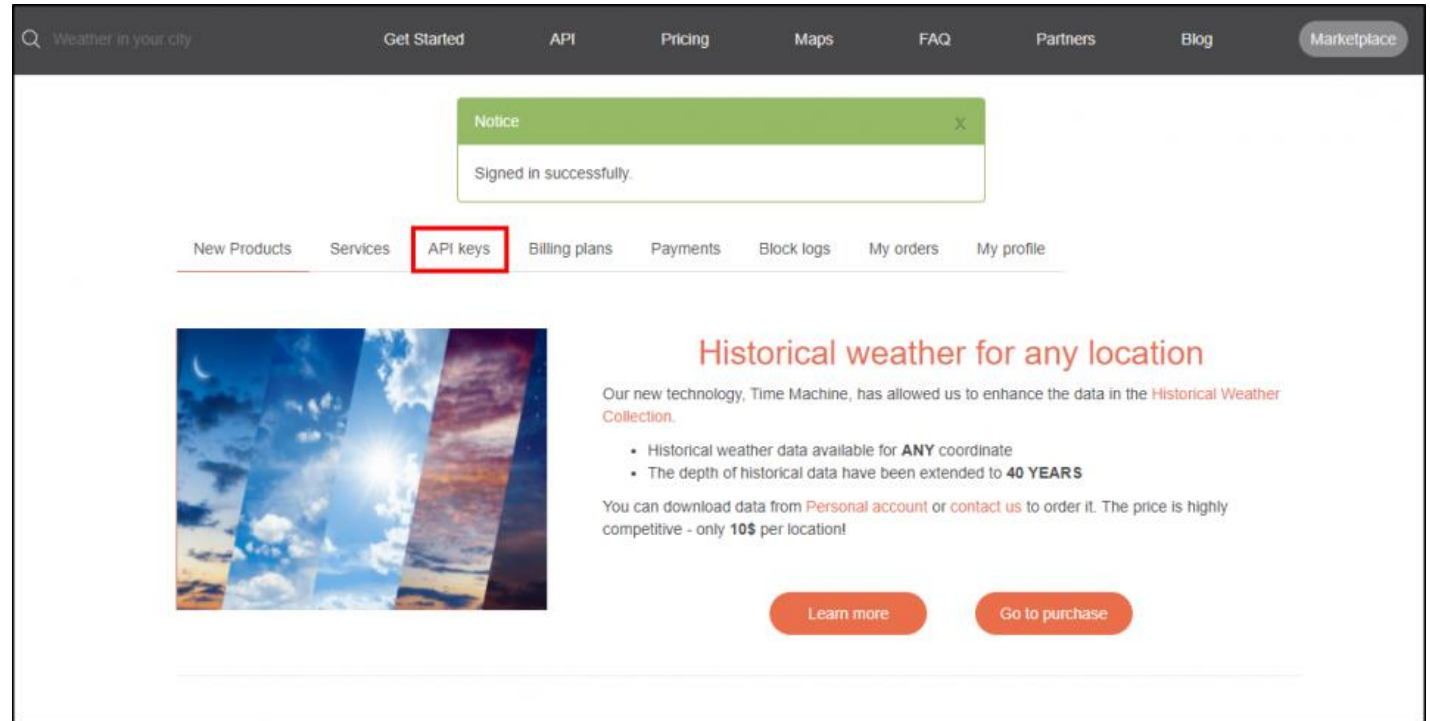
Create Account

API 이해하기 - 실습



Openweathermap API - KEY 발급(2/3)

- API Keys 탭으로 이동



API 이해하기 - 실습



Openweathermap API - KEY 발급(3/3)

- API Key 복사
- API Key 를 복사하여 코드에서 활용합니다.

The screenshot shows the OpenWeather API dashboard. The top navigation bar includes links like 'Weather in your city', 'Guide', 'API', 'Dashboard', 'Marketplace', 'Pricing', 'Maps', 'Our Initiatives', 'Partners', 'Blog', 'For Business', and a user profile 'giryun...'. Below this, a secondary navigation bar has links for 'New Products', 'Services', 'API keys' (which is underlined), 'Billing plans', 'Payments', 'Block logs', 'My orders', 'My profile', and 'Ask a question'.

A light blue informational box states: "You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them."

Below this is a table with the following columns: 'Key', 'Name', 'Status', and 'Actions'. There is one row with a yellow-highlighted key, the name 'Default', and the status 'Active'. The 'Actions' column for this row contains a toggle switch and a copy icon.

To the right of the table is a 'Create key' section with a text input field labeled 'API key name' and a 'Generate' button.

해당 부분의 Key 를 복사합니다.

API 이해하기 - 실습



Openweathermap API - 실습(1/4)

- 특정 지역의 현재 날씨에 대한 모든 정보 출력하기
- 공식 문서 참고

Call current weather data

How to make an API call

API call

```
https://api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&appid={API key}
```



- [참고] 서울의 위도: 37.56 / 경도: 126.97

```
# 서울의 위도
lat = 37.56
# 서울의 경도
lon = 126.97

# API 요청 URL
url = f'https://api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&appid={api_key}'
```

- 출력 결과
- 날짜에 따라 출력 결과는 다를 수 있습니다.

```
{'base': 'stations',
 'clouds': {'all': 0},
 'cod': 200,
 'coord': {'lat': 37.56, 'lon': 126.97},
 'dt': 1684293603,
 'id': 1835848,
 'main': {'feels_like': 300.17,
          'humidity': 39,
          'pressure': 1013,
          'temp': 300.43,
          'temp_max': 301.93,
          'temp_min': 297.96},
 'name': 'Seoul',
 'sys': {'country': 'KR',
         'id': 8105,
         'sunrise': 1684268515,
         'sunset': 1684319716,
         'type': 1},
 'timezone': 32400,
 'visibility': 10000,
 'weather': [{'description': 'clear sky',
               'icon': '01d',
               'id': 800,
               'main': 'Clear'}],
 'wind': {'deg': 200, 'speed': 6.69}}
```

API 이해하기 - 실습

Openweathermap API - 실습(2/4)

- 특정 도시의 현재 날씨를 도시 이름으로 요청하여 모든 정보 출력하기
- 공식 문서 참고

Built-in API request by city name

You can call by city name or city name, state code and country code. Please note that searching by states available only for the USA locations.

API call

```
https://api.openweathermap.org/data/2.5/weather?q={city name}&appid={API key}
```

JSON

XML

List of condition codes

Min/max temperature in current weather

API and forecast API

Other features

Geocoding API

Built-in geocoding

Built-in API request by city name

Built-in API request by city ID

Built-in API request by ZIP code

Format

- [참고] 도시 이름으로 요청 URL 만들기

```
# 검색 조건
city = "Seoul,KR"

# 다른 지역
# city = "Tokyo,JP"
# city = "New York,US"

# API 요청 URL
url = f'http://api.openweathermap.org/data/2.5/weather?q={city}&appid={api_key}'
```

- 출력 결과
- 날짜에 따라 출력 결과는 다를 수 있습니다.

```
{'base': 'stations',
 'clouds': {'all': 0},
 'cod': 200,
 'coord': {'lat': 37.56, 'lon': 126.97},
 'dt': 1684293603,
 'id': 1835848,
 'main': {'feels_like': 300.17,
          'humidity': 39,
          'pressure': 1013,
          'temp': 300.43,
          'temp_max': 301.93,
          'temp_min': 297.96},
 'name': 'Seoul',
 'sys': {'country': 'KR',
        'id': 8105,
        'sunrise': 1684268515,
        'sunset': 1684319716,
        'type': 1},
 'timezone': 32400,
 'visibility': 10000,
 'weather': [{'description': 'clear sky',
               'icon': '01d',
               'id': 800,
               'main': 'Clear'}],
 'wind': {'deg': 200, 'speed': 6.69}}
```

API 이해하기 - 실습



Openweathermap API - 실습(3/4)

- 서울의 현재 날씨 중 온도만 출력하기
- 기본적으로 켈빈(K) 온도를 반환합니다.
- 섭씨 온도 = (켈빈 온도 - 273.15) 로 계산할 수 있습니다.
- 출력 결과

```
켈빈 온도: 300.71K  
섭씨 온도: 27.56°C
```

API 이해하기 - 실습



Openweathermap API - 실습(4/4)

- 서울의 현재 날씨에 대한 설명(description) 데이터만 출력하기
- 출력 결과

```
'날씨 설명: clear sky'
```

- Json 형태의 데이터를 분석하여 원하는 부분만 가져오도록 구성합니다.

```
json_response = requests.get(url).json()
description = json_response['weather'][0]['description']
return f'날씨 설명: {description}'
```



공통 요구사항

- 외부 API 를 사용하여 데이터를 받아오고, 데이터를 원하는 형태로 가공하는 과정입니다.

금융상품통합비교공시 API

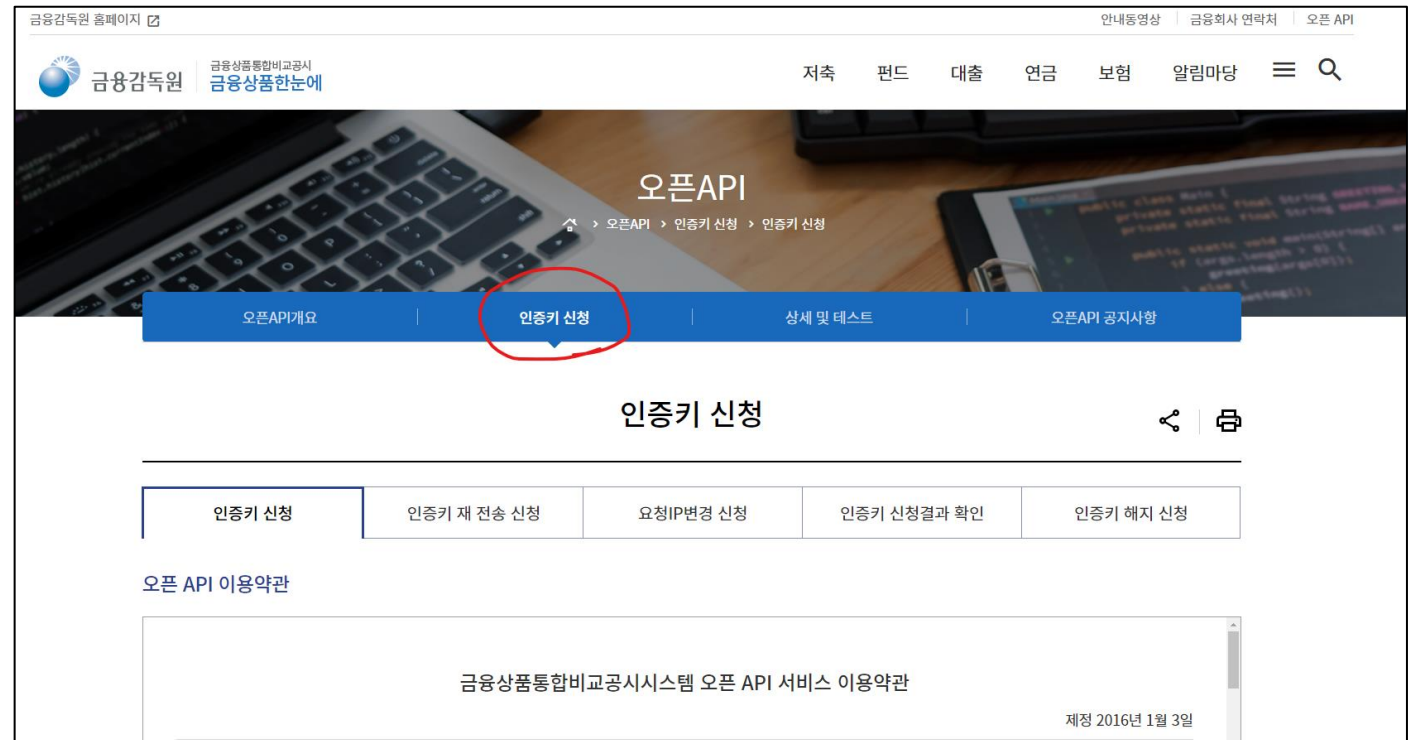
- 요구사항에서 사용할 API 는 금융상품통합비교공시 API 입니다.
- 제공하는 API 중 정기예금 API 를 활용합니다.
- 공식 문서를 보고 데이터의 구조를 먼저 파악한 후 요구사항을 구현합니다.

요구사항



공통 요구사항 - API KEY 발급(1/4)

- 사이트 접속 및 인증키 신청 탭 클릭
- <https://finlife.fss.or.kr/finlife/main/contents.do?menuNo=700029>





공통 요구사항 - API KEY 발급(2/4)

- 스크롤을 내려 하단의 이용약관에 동의합니다.

제 2 조 [용어의 정의]

1. 이 약관에서 사용하는 용어의 정의는 다음 각 호와 같습니다.

가. '금융상품통합비교공시시스템 오픈 API(Application Programming Interface)'라 함은 금융감독원이 금융상품통합비교공시시스템의 금융상품정보를 정보 이용자들이 자유롭게 이용할 수 있도록 하기 위하여 구축한 시스템을 말합니다.

나. 'API 서비스'라 함은 금융감독원 금융상품통합비교공시시스템의 금융회사 금융상품정보를 이용자들이 자유롭게 이용할 수 있도록 금융감독원에서 제공하는 서비스를 말합니다.

다. 인증키라 함은 API 이용자가 API 서비스 이용허가를 받은 "일반" 또는 "기관" 이용자임을 식별할 수 있도록 금융감독원이 API 이용자에게 개별적으로 부여하는 고유한 값을 말합니다.

라. 'API 이용자'라 함은 'API 서비스'를 이용하기 위하여 금융감독원의 승인에 따라 인증키를 발급받은 자로서 '개인' 또는 '법인(단체)'으로 구분합니다.

이용약관에 동의하십니까? ☒ 동의 ☐ 동의하지 않음

개인정보 수집 및 이용 동의

금융감독원은 금융상품통합비교공시시스템 오픈API 서비스를 제공하기 위해 필요한 최소한의 개인정보를 수집합니다.

가. 수집목적 : 인증키 발급

나. 수집항목 : 이메일 주소, 본인인증정보

다. 보유기간 : 인증키 등록 해지 시까지 (단, 관계법령의 규정에 의해 보존이 필요한 경우, 관계법령에서 정한 기간 동안 보관)

귀하는 개인정보 수집 동의를 거부할 수 있으며, 동의 거부시 오픈API 서비스 이용이 제한됩니다.

이용약관에 동의하십니까? ☒ 동의 ☐ 동의하지 않음

요구사항



공통 요구사항 - API KEY 발급(3/4)

- 스크롤을 내려 하단의 신청 정보를 작성합니다.

개인 선택

이용형태

휴대폰 본인인증 진행

☒ 개인 ☐ 단체

개인신청은 신청 즉시 인증키가 발급되고, 일일 조회수는 10,000건으로 제한하며, 원활한 서비스를 위해 기본적으로 1회 조회 시 100건씩(연금지축상품은 10건씩) 조회하여 페이지 처리하고 있습니다.
(i) 관리자의 판단에 따라 개인별로 1회 조회 건수가 변경될 수 있습니다.

표시 항목은 필수 입력 항목입니다.

본인 확인

본인 인증

이메일

@

직접입력

이메일 인증

이메일 인증 코드 입력 입력

(i) 이메일 인증코드는 등록하신 이메일을 통해 확인할 수 있습니다.

사용용도

☒ WEB ☐ APP ☐ WEB+APP

사용URL

인증키 신청

이메일 인증 진행

WEB 탭 클릭 및 사용용도 작성

해당 란은 입력하지 않습니다

모든 인증 완료 후 인증키 신청 버튼 클릭



공통 요구사항 - API KEY 발급(4/4)

- 인증한 메일로 전송된 인증키를 확인한 후 복사하여 활용합니다.



@ 부분을 제외 한 문자열만 복사하여 KEY 로 활용합니다



공통 요구사항 - API 사용 예제

```
import pprint
import requests

def get_deposit_products():
    api_key = "MY_API_KEY"

    url = 'http://finlife.fss.or.kr/finlifeapi/depositProductsSearch.json'
    params = {
        'auth': api_key,
        # 금융회사 코드 020000(은행), 030200(여신전문), 030300(저축은행), 050000(보험), 060000(금융투자)
        'topFinGrpNo': '020000',
        'pageNo': 1
    }
    # 응답을 json 형태로 변환
    response = requests.get(url, params=params).json()
    return response

if __name__ == '__main__':
    # json 형태의 데이터 반환
    result = get_deposit_products()
    # pprint.pprint(): json 을 보기 좋은 형식으로 출력
    pprint.pprint(result)
```

이메일로 전달받은 API KEY



세부 요구사항

- A. 데이터 추출 - Key 값 출력하기
 - B. 데이터 추출 - 전체 정기예금 상품 리스트
 - C. 데이터 가공 - 전체 정기예금 상품들의 옵션 정보 리스트
 - D. 데이터 가공 - 상품과 옵션 정보들을 담고 있는 새로운 값을 만들어 반환하기
-
- [참고] 모든 출력 결과는 예시입니다. 날짜에 따라 다르게 나올 수 있습니다.



A. 데이터 추출 - Key 값 출력하기

- 전체 정기예금의 응답을 json 형태로 변환 후 아래와 같이 Key 값만 출력하도록 구성합니다.

```
dict_keys(['prdt_div', 'total_count', 'max_page_no', 'now_page_no', 'err_cd', 'err_msg', 'baseList', 'optionList'])
```

- 공식 문서의 요청 변수 및 예제 요청결과(JSON) 부분을 참고합니다.
- [힌트] 모든 데이터는 JSON 객체의 “result” 키 값으로 조회할 수 있습니다.



B. 데이터 추출 - 전체 정기예금 상품 리스트

- 응답 중 정기예금 상품 리스트 정보만 출력하도록 구성합니다.
- 출력 결과 예시

```
[{'dcls_end_day': None,
  'dcls_month': '202304',
  'dcls_strt_day': '20230517',
  'etc_note': '- 가입기간: 1~36개월\n'
              '- 최소가입금액: 1만원 이상\n'
              '- 만기일을 일,월 단위로 자유롭게 선택 가능\n'
              '- 만기해지 시 신규일 당시 영업점과 인터넷 홈페이지에 고시된 계약기간별 금리 적용',
  'fin_co_no': '0010001',
  'fin_co_subm_day': '202305171024',
  'fin_prdt_cd': 'WR0001B',
  'fin_prdt_nm': 'WON플러스예금',
  'join_deny': '1',
  'join_member': '실명의 개인',
  'join_way': '인터넷,스마트폰,전화(텔레뱅킹)',
  'kor_co_nm': '우리은행',
  'max_limit': None,
  'mtrt_int': '만기 후\n'
              '- 1개월이내 : 만기시점약정이율×50%\n'
              '- 1개월초과 6개월이내: 만기시점약정이율×30%\n'
              '- 6개월초과 : 만기시점약정이율×20%\n'
              '\n'
              '※만기시점 약정이율 : 일반정기예금 금리',
  'spl_cnd': '해당사항 없음'},
 {'dcls_end_day': '99991231',
  'dcls_month': '202304',
  'dcls_strt_day': '20230420',
```



c. 데이터 가공 - 전체 정기예금 옵션 리스트

- 응답 중 정기예금 상품들의 옵션 리스트를 출력하도록 구성합니다.
- 이 때, 원하는 데이터만 추출하여 출력되는 결과를 아래와 같이 변경하여 반환하는 함수를 작성하시오.

- 출력 결과 예시

```
[{'금융상품코드': 'WR0001B',  
  '저축 금리': 3.39,  
  '저축 기간': '6',  
  '저축금리유형': 'S',  
  '저축금리유형명': '단리',  
  '최고 우대금리': 3.39},  
{ '금융상품코드': 'WR0001B',  
  '저축 금리': 3.57,  
  '저축 기간': '12',  
  '저축금리유형': 'S',  
  '저축금리유형명': '단리',  
  '최고 우대금리': 3.57},  
{ '금융상품코드': 'WR0001B',  
  '저축 금리': 3.36,
```

- [참고] Python Dictionary 원하는 키 값으로 데이터 추가하기

```
new_dict = {}  
new_dict['추가'] = "test"  
print(new_dict)
```

- 출력 결과

```
{'추가': 'test'}
```



D. 데이터 가공 - 새로운 값을 만들어 반환하기(1/2)

- 상품과 옵션 정보들을 담고 있는 새로운 값을 만들어 딕셔너리 형태로 반환하도록 구성합니다.
- 다음과 같은 값만 추출하여 새로운 값에 포함합니다.

=> 금융 상품: '금융회사명', '금융상품명', '금리정보'

=> 해당 금융 상품의 금리 정보(옵션): '저축금리유형', '저축금리유형명', '저축 기간', '저축 금리', '최고 우대금리'

- 하나의 금융 상품에 대해 여러 개의 옵션을 가질 수 있습니다.
- [힌트] 금융 상품 코드가 같은 금융 상품과 옵션을 하나의 딕셔너리로 만듭니다.



D. 데이터 가공 - 새로운 값을 만들어 반환하기(2/2)

- 출력 결과 예시

=> WON플러스예금에 대한 정보

=> 4가지 금리 정보를 포함하고 있습니다.

```
[{'금리정보': [{'저축 금리': 3.39,  
  '저축 기간': '6',  
  '저축금리유형': 'S',  
  '저축금리유형명': '단리',  
  '최고 우대금리': 3.39},  
{'저축 금리': 3.57,  
  '저축 기간': '12',  
  '저축금리유형': 'S',  
  '저축금리유형명': '단리',  
  '최고 우대금리': 3.57},  
{'저축 금리': 3.36,  
  '저축 기간': '24',  
  '저축금리유형': 'S',  
  '저축금리유형명': '단리',  
  '최고 우대금리': 3.36},  
{'저축 금리': 3.33,  
  '저축 기간': '36',  
  '저축금리유형': 'S',  
  '저축금리유형명': '단리',  
  '최고 우대금리': 3.33}],  
'금융상품명': 'WON플러스예금',  
'금융회사명': '우리은행'}
```