Api Json, Gson

JSON 객체란?

Javascript Object Notation의 줄임말로

자바스크립트에서의 객체를 의미한다.

자바의 객체와 자바스크립트의 객체 비교하기

자바

Student타입의 hodong 객체 선언

```
Student hodong = new Student
```

정보 꺼내기

```
hodong.getName()
```

자바스크립트

hodong이라는 객체를 let 을 이용하여 선언

```
let hodong = { name : "호동이"
```

정보 꺼내기

```
hodong.name
```

클라이언트 - 서버 간 데이터 통신 시

자바에서 JSON 객체를 사용해야하는 경우가 존재하게 된다.

서버에서 응답데이터로 JSON String을 전달하고

자바에서 JSON 객체를 생성하고, JSON String으로 변환하는 과정을 가진다.

즉, JSON String → "{ name : "호동이" , kor : 98, eng : 80, math : 80 }" 이라는 데이터를 받게 되고
요청데이터로 넘어온 JSON String 사용하기 위해
자바에서 JSON String을 JSON 객체로 변환한다는 뜻

단순히 JDK안에 포함되어있는 라이브러리(클래스 파일들) 로는 불가능

(→ 자바 자체적으로는 불가능하다는 의미)

해결법 : 외부 라이브러리 가져오기 (게임 DLC 같은 느낌)

구글에 maven repository 검색 → 사이트 접속 → json simple 검색 → 1.1.1 버전 선택
→ Files에서 View All 선택 후 .jar 파일 다운로드

다운로드 받은 .jar 파일을 프로젝트 내부에서 사용할 수 있게하기
프로젝트 내부에 폴더를 하나 생성해서 그 안에 .jar 파일 옮기기

Menu - Project Structure - Libraries 에서 해당 파일 추가

해결법 적용 후, 자바(인텔리제이) 안에서도 JSON 객체 생성이 가능해진다 !

```
JSONObject jsonObject = new JSONObject();
        System.out.println(jsonObject);
```

JSON 객체 데이터 특징

- JSON 객체는 내부에 (Key, Value) 쌍으로 구성된 데이터를 가진다.
- Key값은 String 타입, Value는 아무 타입 다 가능한 형태를 지닌다.
- 자바의 Map<String, Object>와 유사하다.

데이터 추가하기 **.put**

```
jsonObject.put("name", "아이유");
System.out.println(jsonObject);
jsonObject.put("age", "32");
System.out.println(jsonObject);
```

데이터 수정하기 **.put (→ 덮어쓰기하는 느낌)**

```
jsonObject.put("age", 31);
        System.out.println(jsonObject);
```

Value에 배열(jsonArray) 넣기

- JSONArray는 ArrayList와 사용법이 같음

```
JSONArray jsonArray = new JSONArray();
          System.out.println(jsonArray);
```

- 생성한 jsonArray에 값 추가하기 **.add**

```
jsonArray.add("블루밍");
        jsonArray.add("밤편지");
        jsonArray.add("팔레트");
        jsonArray.add(100); // JSONArray 또한 타입에 자유롭다.
        System.out.println(jsonArray);
```

- **jsonObject라는 JSON 객체에 Key = “songList” 이고, Value = “블루밍,밤편지,팔레트,100”을 담은 배열(jsonArray)을 넣음**

```
jsonObject.put("songList", jsonArray);
System.out.println(jsonObject);

// 단, jsonObject를 print해도 자바의 Map처럼 데이터 입력순으로 출력된다
// 정렬이 되어 출력된다던가하지 않고 무의미한 순서로 출력됨

System.out.println(jsonObject.toString()); // JSON String 리턴
System.out.println(jsonObject.toJSONString()); // JSON String
```

- Value에 JSON 객체 추가

```
JSONObject mellomangs = new JSONObject();
        mellomangs.put("name", "멜로망스");
        mellomangs.put("age", 33);

JSONArray songs = new JSONArray();
        songs.add("선물");
        songs.add("좋은날");

mellomangs.put("songs", songs);
        System.out.println(mellomangs);
// mellomangs 객체에 Key "songs"와 "선물","좋은날"을 담은 배열이 Val

        jsonObject.put("friend", mellomangs);
        System.out.println(jsonObject);
// 아까 사용한 아이유와 노래 리스트가 담긴 jsonObject에 추가적으로 Key
// Value가 배열이 담긴 mellomangs !객체!가 들어오게 됨
```

- Key에 대한 Value 꺼내기

```
System.out.println(jsonObject.get("name"));
        // 값을 꺼낼 시, Object 타입으로 리턴됨,
                // 원하는 타입에 담기 위해서는 강제 형변환이 필요함

        String name = (String) jsonObject.get("name");
```

```
                // Object name = jsonObject.get("name");
        System.out.println(name);

                int age = (int) jsonObject.get("age");
        System.out.println(age);

        JSONArray iuSongs = (JSONArray) jsonObject.get("songL
        System.out.println(iuSongs);


                JSONObject friend = (JSONObject) jsonObject.g
        System.out.println(friend);

        String mello = (String) friend.get("name");
        System.out.println(mello);

                System.out.println(jsonObject.get("frie"));
                // -> "frie"라는 Key는 없으므로 Null을 출력
```

✔️ **JSON Parser Online으로 검색하면 가독성 좋게 구분해주는 사이트가 있음**

```
// 서버로부터 JSON String을 받은 경우
String serverData = "{\"name\":\"아이유\",\"friend\":{\"songs\"

// JSON String 을 JSON 객체로 변환해야 Key 값을 이용하여
        // Value를 꺼낼 수 있다.
        // JSON String -> JSONObject 변환을 위해 JSONParser 객체
        JSONParser jsonParser = new JSONParser();

        JSONObject jsonData = (JSONObject) jsonParser.parse(s

        System.out.println(jsonData.get("name"));
```

연습 문제 : JSON 객체 내 '선물' 꺼내기

```
System.out.println(jsonData.get("friend"));

        JSONObject friendData = (JSONObject) jsonData.get("fr

        JSONArray friendArray = (JSONArray) friendData.get("s

        System.out.println(friendArray.get(0));
```

GSON 라이브러리 사용하기

해결법 : 외부 라이브러리 가져오기 (게임 DLC 같은 느낌)

구글에 maven repository 검색 → 사이트 접속 → GSON 검색 →
→ Files에서 View All 선택 후 .jar 파일 다운로드

다운로드 받은 .jar 파일을 프로젝트 내부에서 사용할 수 있게하기
프로젝트 내부에 폴더를 하나 생성해서 그 안에 .jar 파일 옮기기

Menu - Project Structure - Libraries 에서 해당 파일 추가

- GSON 라이브러리의 JSON객체 생성하기

```
JsonObject gsonObject = new JsonObject();
        System.out.println(gsonObject);

// 아무 데이터도 담지 않은 {} 출력
```

- gsonObject 에 값 추가하기 **.addProperty**

```
gsonObject.addProperty("name", "아이유");
gsonObject.addProperty("age", 31);
      System.out.println(gsonObject.toString());
```

- Gson 객체 생성하기

```
// 심플하게 생성하기
Gson gson = new Gson();

// GsonBuilder로 생성하기
GsonBuilder gsonBuilder = new GsonBuilder();
        Gson gson2 = gsonBuilder.create(); //  위와 같음 (1번)
        Gson gson3 = gsonBuilder.setPrettyPrinting().create()
```

- JSON String 얻기

```
String gsonString = gsonObject.toString();
        System.out.println(gsonString);

        String gsontString2 = gson.toJson(gsonObject);
        System.out.println(gsontString2);

        String gsonString3 = gson3.toJson(gsonObject);
        System.out.println(gsonString3);
```

JSON simple 라이브러리와 별 차이 없는 GSON

🔴 JSON simple 라이브러리 GSON의 차이점

(강제 형변환을 하지 않아도 된다. —> 직렬화가 가능하다 !)

```
Student hobbang = new Student("호빵맨", 98, 63, 50);
```

- **.toJson :** gson을 이용하여 자바 객체를 JSON String으로 변환이 가능하다 (→ 직렬화)

```
String strHobbang = gson.toJson(hobbang);
        System.out.println(strHobbang);
```

- **.toJson :** 자바의 배열을 Json String으로 변환이 가능하다

```
String[] sutArray = {"호빵맨", "찐빵맨", "식빵맨"};
        String strArrray = gson.toJson(sutArray);
        System.out.println(strArrray);
```

- **.toJson : 자바의 ArrayList를 Json String으로 변환**이 가능하다

```
ArrayList<String> stuList = new ArrayList<>();
        stuList.add("호동이");
        stuList.add("호랑이");
        stuList.add("호돌이");
        String strList = gson.toJson(stuList);
        System.out.println(strList);
```

- **.toJson :** ArrayList 안에 객체가 들어있어도 무리없이 변환이 가능하다

```
ArrayList<Student> students = new ArrayList<>();
        students.add(new Student("호빵이", 80, 90, 40));
        students.add(new Student("식빵이", 63, 83, 93));
        students.add(new Student("밤빵이", 60, 94, 59));
```

```
        String strStu = gson.toJson(students);
        System.out.println(strStu);
```

- **.fromJson :** Json String에 대해 Student 객체로 변환 가능이 가능하다 (→ 역직렬화)

▼ 대신 변환하고자 하는 객체가 해당하는 클래스를 확실히 하기 위해 .class를 추가해줘야 한다

```
System.out.println(strHobbang);
        Student hobbangStu = gson.fromJson(strHobbang, Studen
        System.out.println(hobbangStu); // String 타입이었던 st
        System.out.println(hobbangStu.getName()); // .getName
        System.out.println(hobbangStu.getMath());
```

연습 문제

String resp = "{\"success\": true, \"timeseries\": true, \"base\": \"USD\", \"start_date\": \"2020-08-01\", \"end_date\": \"2022-08-01\", \"rates\": {\"2020-08-01\": {\"JPY\": 105.65, \"KES\": 107.57, \"KGS\": 76.52, \"KHR\": 4092.64, \"KMF\": 417.5, \"KPW\": 898.38, \"KRW\": 1192.25}, \"2020-08-02\": {\"JPY\": 105.74, \"KES\": 107.69, \"KGS\": 76.62, \"KHR\": 4100, \"KMF\": 418.06, \"KPW\": 899.58, \"KRW\": 1193.46}}}";

2020년 8월 2일의 한국 환율(KRW) 값을 꺼내라.
결과 : 1193.46

데이터를 가독성 좋게 구분해주는 사이트를 참고해도 좋다

```
JSONParser respParser = new JSONParser();
        JSONObject jsonResp = (JSONObject) respParser.parse(r

        JSONObject jsonRates = (JSONObject) jsonResp.get("rat
//        System.out.println(jsonRates);
```

```
        JSONObject jsonDate = (JSONObject) jsonRates.get("202
//        System.out.println(jsonDate);
        System.out.println(jsonDate.get("KRW"));
```

선생님 버전

```
JSONObject respJson = (JSONObject) respParser.parse(resp);

        System.out.println(gson3.toJson(respJson));
        // 나 = JSON Parser Online을 사용해 가독성 좋게 확인(인터넷)
        // 선생님 =  Gson gson3 = gsonBuilder.setPrettyPrinting

        // 아래는 변수명만 다를뿐 풀이과정은 같음
        JSONObject rates = (JSONObject)respJson.get("rates");
        System.out.println(rates);

        JSONObject date = (JSONObject) rates.get("2020-08-02"
        System.out.println(date);

        double krw = (double)date.get("KRW");
        // KRW 의 Value 값은 Object가 아닌 double타입이므로 double
        System.out.println(krw);
                // 나 : "KRW"의 Value를 바로 출력
                // 선생님 : "KRW"의 Value를 double에 담기 위해 강제
                // double타입 객체를 출력해서 확인함.
```