

| 카카오톡 봇 강좌 | >

[고급] 강좌 #6 객체 (응용) - 길찾기 만들기



doami2005 챗봇 창시자 🌟 1:1 채팅

2019.12.31. 20:18 조회 626

💬 댓글 12 URL 복사 ⋮

<https://cafe.naver.com/nameyee/4908>

↳ 객체 초급

<https://cafe.naver.com/nameyee/8232>

↳ 객체 응용

※주의※ 본글은 위 두글과 객체를 어느정도 이해하고 있어야 이해가 편합니다.

지금까지 이야기해본 것들을 요약하겠습니다.

객체는 키:값 으로 이루어진 것들이 모여있는 것을 말합니다.

예시) {"a":1, "b":"Hello"}

객체 안에는 객체, 배열이 또 들어갈 수도 있습니다.

예시) {"user" : [{"name":"도미", "type":"학생"}, {"name":"사로로", "type":"군바(?)"}, {"name":"줄려", "type":"14살(?)"}], "Hello" : "World!"}

이제 위 객체처럼 복잡한, 혹은 더 복잡한 객체들의 구조를 파악하는 방법을 알아보겠습니다.

1. 객체를 돌려본다.

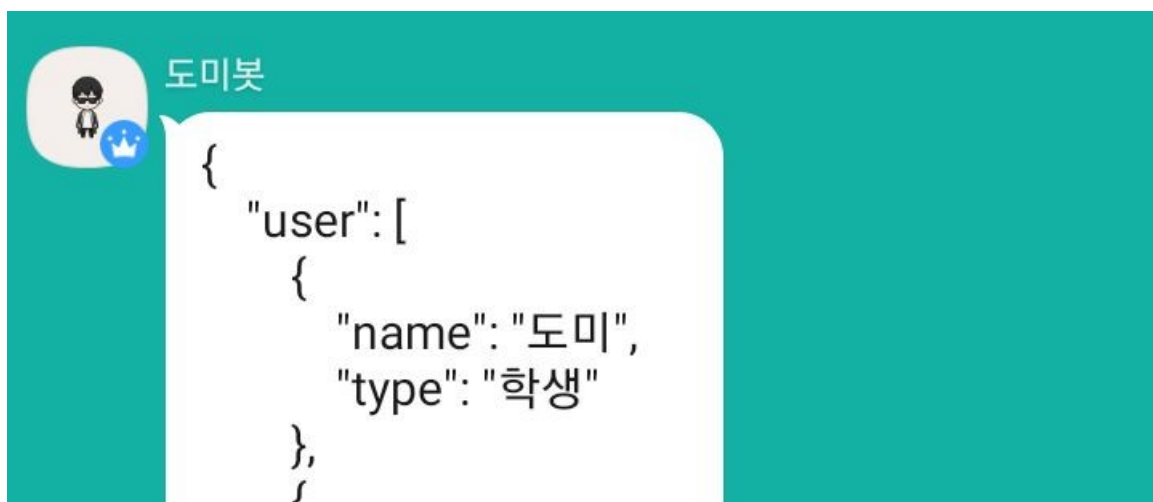
못 알아보겠으면 2단계로 이동

2. JSON.stringify(json, null, 4) 로 본다.

예시를 들어보겠습니다.

위에 있던 객체를 이용하여

JSON.stringify(json, null, 4) 를 반환한 결과,



```
{
  "name": "사로로",
  "type": "군바(?)",
},
{
  "name": "줄려",
  "type": "14살(?)",
}
],
"Hello": "World!"
}
```

99

오후 7:17

라는 깔끔한 결과를 얻게 되었습니다.

JSON.stringify(json) 는
해당 객체를 그대로 문자열로 만들어 반환하는 함수입니다.
JSON.parse는 그 반대입니다.

예시1)

```
json = {a:1}
```

```
json["a"]
// 1
```

```
str = JSON.stringify(json)
// '{"a":1}'
```

```
str["a"]
// undefined
```

예시2)

```
str = '{"Hello":"World"}'
```

```
str["Hello"]
// undefined
```

```
json = JSON.parse(str)
// {"Hello":"World"}
```

```
json["Hello"]
// "World"
```

3. 제일 가까운 키를 입력하여 범위를 줄여나간다.

복잡한 객체들은 구조를 파악한 뒤 반복문 같은 걸 쓰면 금방 해체해서 정리할 수 있습니다.

제가 지금까지 객체들을 보며
가장 복잡했던 것은 네이버 길찾기 소스를 만들 때입니다.

<https://m.map.naver.com/apis/rp/pubtrans/summary?apiVersion=3&searchType=0&start=126.8948249,37.5179987,문래역%202호선&destination=126.9238987,37.5215199,여의도역%205호선>

위 링크는 문래역 2호선에서 여의도역 5호선까지 가는 경로입니다.

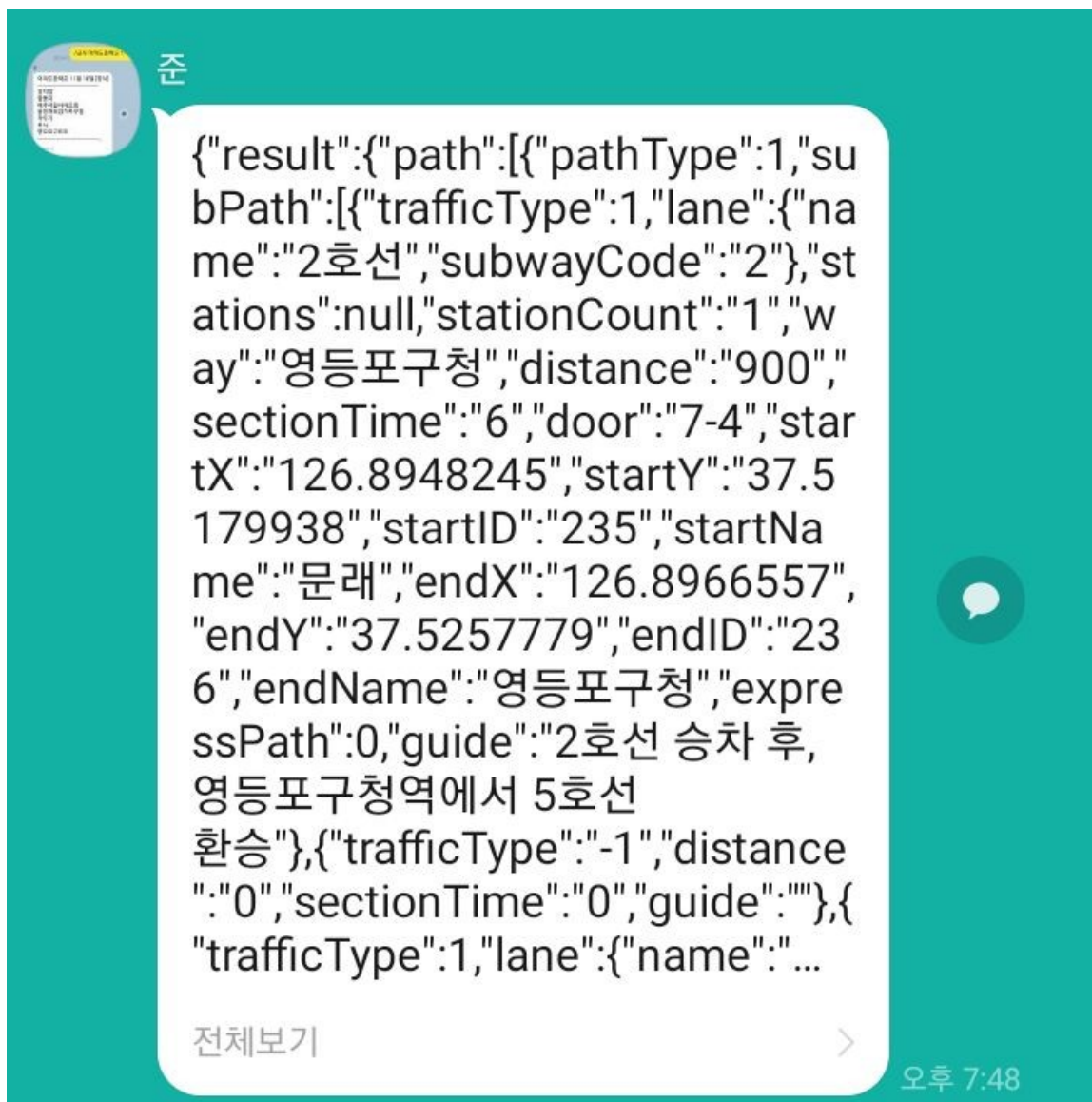
객체가 매우 복잡해 보입니다.

해당 객체를 json이라 할 때,
json["result"]["path"]의 결과는 [로 시작되는 걸 보아 배열이며,
더 자세히 살펴보면 경로들인 것을 알 수 있습니다.

그 안에 subPath 에는 상세한 방법이 쓰여 있네요.

그럼 한번 저 링크를 파싱해보도록 하겠습니다.

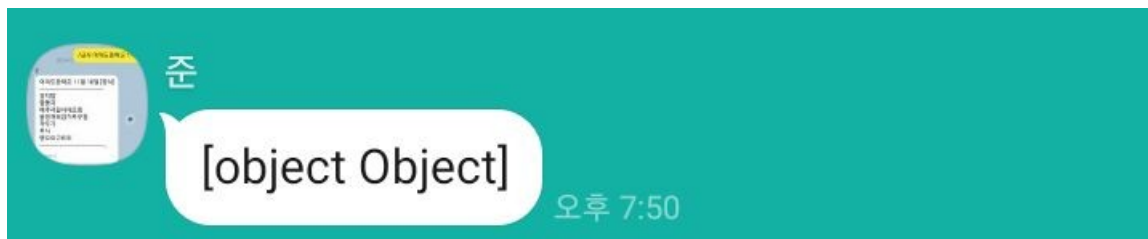
```
parse = org.jsoup.Jsoup.connect("https://m.map.naver.com/apis/rp/pubtrans/summary?apiVersion=3&searchType=0&start=126.8948249,37.5179987,문래역%202호선&destination=126.9238987,37.5215199,여의도역%205호선").ignoreContentType(true).get().text()
```



이와같이 객체가 나옵니다.

근데 객체를 사용하려면 저런 문자열이 아닌 Object 로 만들어야 하니
JSON.parse를 씁워줍니다.

```
json = JSON.parse(parse)
```



자 이제 분석할 수 있게 되었습니다.

```
json = json["result"]["path"]
```



총 21개의 객체가 담긴 배열이 나왔네요
(경로 수 : 21가지)

이제 배열의 메서드인 map을 사용해 봅시다.

*갑자기 시작된 map 강좌

객체를 아신다면 다들 배열을 아실거라 생각합니다.

```
arr = [1, 2, 3, 4, 5]  
arr = arr.map(a => a*2)  
// [2, 4, 6, 8, 10]
```

이는 배열 안에 담긴 객체에도 사용할 수 있습니다.

```
arr = [{"name": "홍길동", "age": 1}, {"name": "동길홍", "age": 2}];
```

```
arr = arr.map(a => "이름 : " + a["name"] + "\n나이 : " + a["age"]).join("\n\n");
```

```
/*  
이름 : 홍길동  
나이 : 1
```

```
이름 : 동길홍  
나이 : 2  
*/
```

이렇게 map은 다양한 곳에 유용하게 쓰입니다. 저기에도 시도해 보도록 하겠습니다.

```
path1 = JSON.stringify(json[0])  
을 해보면
```

SKT 7:58 

    25% 

← 전체보기



```
{"pathType":1,"subPath":[{"trafficType":1,"lane":{"name":"2호선","subwayCode":"2"},"stations":null,"stationCount":"1","way":"영등포구청","distance":"900","sectionTime":"6","door":"7-4","startX":"126.8948245","startY":"37.5179938","startID":"235","startName":"문래","endX":"126.8966557","endY":"37.5257779","endID":"236","endName":"영등포구청","expressPath":0,"guide":"2호선 승차 후, 영등포구청역에서 5호선 환승"},{"trafficType":-1,"distance":"0","sectionTime":"0","guide":""},{"trafficType":1,"lane":{"name":"5호선","subwayCode":"5"},"stations":[{"id":"524","name":"영등포시장","isNonStop":"0"},{"id":"525","name":"신길","isNonStop":"0"}],"stationCount":"3","way":"여의도","distance":"3000","sectionTime":"13","door":null,"startX":"126.8954458","startY":"37.5241988","startID":"523","startName":"영등포구청","endX":"126.9242845","endY":"37.5217254","endID":"526","endName":"여의도","endExitNo":"1","endExitX":"126.923566198587","endExitY":"37.521301613773","expressPath":0,"guide":"5호선 승차 후, 여의도역 하차"},{"trafficType":"3","distance":"47","sectionTim
```



```
e":"1","guide":"1번 출구로 나와서 여의도역 5호선까지  
약 47m  
걷기"}], "info":{"mapObj":"2:2:235:236@5:2:523:526|  
0:0:0:0@0:0:126.923566198587:37.52130161377  
3","payment":"1250","busTransitCount":"0","subwa  
yTransitCount":"2","busStationCount":"0","subway  
StationCount":"4","totalStationCount":"4","totalTim  
e":"22","totalWalk":"47","trafficDistance":"3900","tot  
alDistance":"3947","firstStartStation":"문래","lastEn  
dStation":"여의도","totalWalkTime":"1","subwayExpr  
ess":"0"}}}
```

첫번째 경로의 정보가 나옵니다.
다른 거 다 무시하고
상세경로만 가져오도록 하겠습니다.


분석해보면

subPath에 배열이 더 있고, 그 배열의 guide에 상세경로가 있네요.

여기선 for문이나 map을 쓰는데,
전 map을 쓰도록 하겠습니다.
(for문 쓰는 건 제 길찾기 오픈소스 참고)

```
arr = new Array(path1["subPath"].length).fill()
```

```
arr = arr.map((_, index) => (index+1)+") "+path1["subPath"][index]["guide"]).join("\n\n")  
// _ 는 첫번째 인자가 필요없기 때문에 그냥 한 것 (변수명 아끼기)
```



준

- 1) 2호선 승차 후, 영등포구청역에서 5호선 환승
- 2)
- 3) 5호선 승차 후, 여의도역 하차
- 4) 1번 출구로 나와서 여의도역 5호선까지 약 47m 걷기

오후 8:06

이렇게 첫 경로의 상세경로가 나왔네요.
(저기 2번 칸이 비어있는 이유는 api 자체 버그입니다. splice 등으로 뺄 수 있습니다. 하지만 귀찮으니 내버려두기ㅎ)

이렇게 상세경로를 가져와 봤으니
이제 이중 map을 사용하여
모든 경로의 상세경로들을 가져오겠습니다.

```
parse = org.jsoup.Jsoup.connect("https://m.map.naver.com/apis/rp/pubtrans/summary?apiVersion=3&searchType=0&start=126.8948249,37.5179987,문래역%20호선&destination=126.9238987,37.5215199,여의도역%205호선").ignoreContentType(true).get().text();
json = JSON.parse(parse);
json = json["result"]["path"];
json.map((route, wayIndex) => (wayIndex+1)+"번째 방법\n\n"+new Array(route["subPath"].length).fill().map((_, index) => (index+1)+ " "+route["subPath"][index]["guide"]).join("\n"))).join("\n\n")
```

SKT 8:16 TALK

🕒 🔇 📶 📶 38% 🔋

← 전체보기



1번째 방법

- 1) 2호선 승차 후, 영등포구청역에서 5호선 환승
- 2)
- 3) 5호선 승차 후, 여의도역 하차
- 4) 1번 출구로 나와서 여의도역 5호선까지 약 47m 걷기

2번째 방법

- 1) 영등포구청역 6번 출구까지 약 770m 걷기
- 2) 5호선 승차 후, 여의도역 하차
- 3) 1번 출구로 나와서 여의도역 5호선까지 약 47m 걷기

3번째 방법

- 1) 당산동진로아파트 정류장까지 약 526m 걷기
- 2) 5012 승차 후, 여의도역 정류장 하차
- 3) 여의도역 5호선까지 약 132m 걷기

4번째 방법

- 1) 다사도지르아파트 정류장까지 약 526m 걷기

1) 영등포역 정류장까지 약 320m 걷기

2) 6628 승차 후, 전경련회관 정류장 하차

3) 여의도역 5호선까지 약 641m 걷기

5번째 방법

1) 문래역 정류장까지 약 169m 걷기

2) 영등포05 승차 후, 영등포역 정류장 하차

3) 영등포역 정류장까지 약 133m 걷기

4) 5618 승차 후, 여의도역 정류장 하차

SKT 8:17  

    38% 

← 전체보기



3) 당산역 12번 출구까지 약 177m 걷기

4) 9호선 승차 후, 여의도역 하차

5) 1번 출구로 나와서 여의도역 5호선까지 약 102m 걷기

19번째 방법

1) 문래역 정류장까지 약 181m 걷기

2) 6211 승차 후, 노량진수산물시장.CTS기독교TV 정류장 하차

3) 노량진역 6번 출구까지 약 96m 걷기

4) 9호선 승차 후, 여의도역 하차

5) 1번 출구로 나와서 여의도역 5호선까지 약 102m 걷기

20번째 방법

1) 2호선 승차 후, 구로디지털단지역 하차

2) 1번 출구로 나와서 구로디지털단지역 정류장까지 약 94m 걷기

3) 5623 승차 후, 여의도역 정류장 하차

4) 여의도역 5호선까지 약 167m 걷기

21번째 방법

- 1) 영등포구청역 6번 출구까지 약 770m 걷기
- 2) 5호선 승차 후, 여의도역에서 9호선 환승
- 3)
- 4) 9호선 승차 후, 샛강역 하차
- 5) 4번 출구로 나와서 여의도역 5호선까지 약 699m 걷기



이렇게 21번째 경로까지 잘 나오는 것을 확인하실 수 있습니다.
나머지 정보들도 저런 식으로 활용하여 집어넣을 수 있습니다.

SKT 8:18  

    38% 

← 전체보기



[!] 문래역 2호선 -> 여의도역 5호선 의 경로

총 21개

1번째

- 1) 2호선 승차 후, 영등포구청역에서 5호선 환승 (900m, 6분)
- 2) (0m, 0분)
- 3) 5호선 승차 후, 여의도역 하차 (3000m, 13분)
- 4) 1번 출구로 나와서 여의도역 5호선까지 약 47m 걷기 (47m, 1분)

초 GRANDE · 1250원

총 시간 : 22분
도보 시간 : 1분
총 거리 : 3947m
도보 거리 : 47m
버스 환승 횟수 : 0회
지하철 환승 횟수 : 2회
지나가는 정류장 : 0개
지나가는 역 : 4개

2번째

(오픈소스 결과물)

이 밖에도 배열, 객체 메서드들은 구글링하면 많은 정보들이 있으니 참고하시기 바랍니다.

이렇게 길찾기를 만드는 방법을 설명해 보았는데요,

무엇보다 중요한 것은 보는 것이 아닌 본인이 직접 시도해서 만들어 보는 것이 제일 가치있고 중요하다고 생각합니다.

현재 시도하고 있는 것이 너무 어렵다면
그 전 난이도부터 천천히 마스터하고, 시간이 흐르면 다시 시도해 보세요

그 전엔 안 보이던 것들이 한눈에 보이고 쉽다고 느껴집니다.

/* 도미의 강좌 시리즈 */

[try문]

- <https://cafe.naver.com/nameyee/4540>

[배열]

- <https://m.cafe.naver.com/nameyee/9524>

[객체]

- 초급) <https://cafe.naver.com/nameyee/4908>

- 응용) <https://cafe.naver.com/nameyee/8232>

- 심화) <https://cafe.naver.com/nameyee/9501>

[Tip]

eval 의 위험성과 사용법

- <https://cafe.naver.com/nameyee/4341>

파싱 시 Utils 와 Jsoup 의 장단점

- <https://cafe.naver.com/nameyee/4342>



doami2005님의 게시글 더보기 >

❤ 좋아요 2 💬 댓글 12

🔗 공유 | 신고

댓글 등록순 최신순 🔄

댓글알림 ☐



즐러
나다

2019.12.31. 20:36 답글 쓰기



doami2005 작성자
라마

2019.12.31. 20:37 답글 쓰기



즐러
doami2005 가는 어디에?

2019.12.31. 20:38 답글 쓰기



doami2005 작성자
즐러 방금 쓰셨네요

2019.12.31. 20:38 답글 쓰기



즐러
doami2005 ㅎㅎㅎㅎ

2019.12.31. 20:38 답글 쓰기



NotreDame

와,,, 너무 어려워요 ㄷ 진짜 하나도 모르게써여

2019.12.31. 20:59 답글 쓰기



NotreDame

근데 fill에 아무것도 안써도 돼여? array.fill(4) .fill(0,2,4) 이래야되는거 아님?

2019.12.31. 21:01 답글 쓰기



doami2005 작성자
map 쓰려고 빈 배열 만든거예영
해당 index 없으면 맵 안써져서ㅠ

2019.12.31. 21:02 답글 쓰기



NotreDame

doami2005 그런 거시군요오

2019.12.31. 21:03 답글 쓰기



| 화배 |

그냥 JSON.stringify(객체,null,4) 하면 띄어쓰기 4개로 들여쓰기됩니다

2020.02.11. 06:45 답글 쓰기



도미 doami2005 작성자
오잉 위에서 썼었는데

너무 길어져서 짧릴까바 거대객체에는 안썼어요 ㅠㅠ

2020.02.11. 08:25 답글 쓰기



야옹
히익

2020.06.02. 13:17 답글 쓰기

Hibot

댓글을 남겨보세요



등록

'| 카카오톡 봇 강좌 |' 게시판 글

전체 [고급] 말머리 글

이 게시판 새글 구독하기 ☐

핑퐁 핸드폰으로 하기! 🤖 [2]	소	2020.01.03.
[중급] 강좌 #7 배열 - 자세한 메서드의 사용 [10]	doami2005	2020.01.01.
[고급] 강좌 #6 객체 (응용) - 길찾기 만들기 🤖 [12]	doami2005	2019.12.31.
[초급] [라코스크립트] 라코스크립트란 무엇인가 / 특정 채팅에 반응 🤖 [2]	Dark Tornado	2019.12.29.
[초급] 청정수가 알려주는 왕초보강좌 1편 [18]	임티봇	2019.12.27.

1 2 3

전체보기

이 카페 인기글

error(재팅수(LTE
Error: "FS" is not
=O)#14) at
=O):14 (respons

채팅순위봇이 갑자기 안되네요..ㅠ

난미!!!!!!!!!!!!!!

저도

[확습] 모니카는 여기에 잘 있음

난미!!!!!!!!!!!!!!

[카카오톡 봇] 양산형 자동학습 봇 소스 공유

Dark Tornado

eval 질문

Kiri

♡0 💬5

태양, 달 정보 구현 완료

틱택토 (카카오링크 적용)

오른이발 괴롭히기(메봇R)

5ml doami2005

ERROR

♡0 💬7

카톡방 나가기

hajuhee01

♡0 💬15

덧셈/그룹챗 구분

시초는중국발

♡0 💬8

가르치기 리로드..

Milk2

♡0 💬9

1 2 3 4 5