

2024년 상반기 K-디지털 트레이닝

# JSON

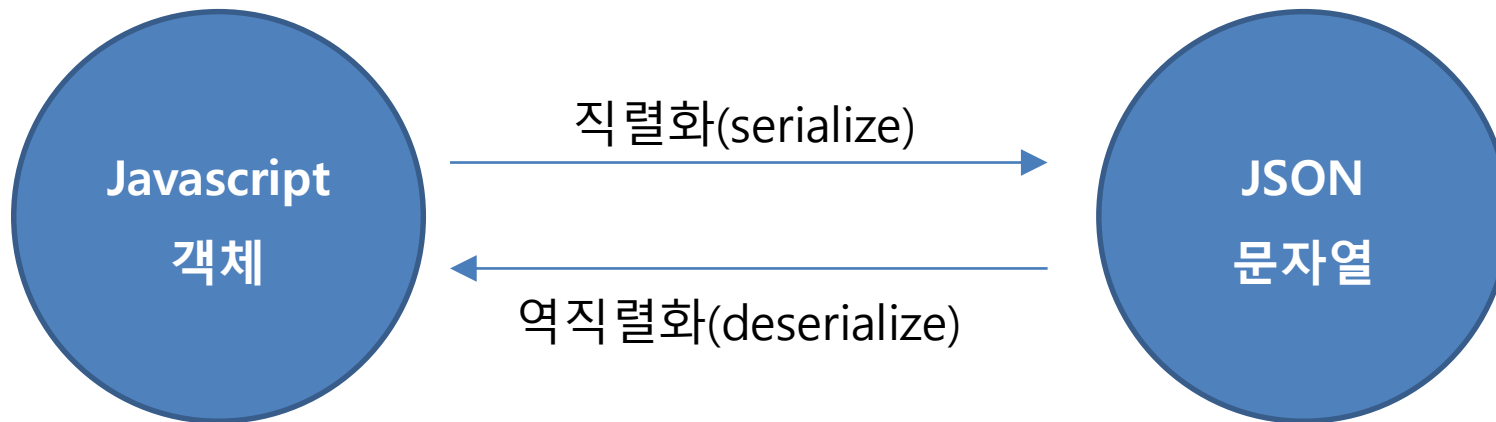
---

[KB] IT's Your Life

# 1 JSON

## ✓ JSON JavaScript Object Notation

- Javascript 객체 표기 방법 대로 객체(인스턴스)의 내용을 문자열로 표현한 것
- 텍스트 기반 통신에서 구조화된 정보를 문자열로 표현하는 용도로 활용
- Javascript 객체의 데이터 속성만 표현, 메서드는 제외(보안 문제 때문)
- Javascript의 표준 JSON 객체로 처리



## ✓ JSON 데이터 유형

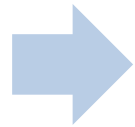
- 배열: [ ]
- Boolean: true/false
- Null: null
- 숫자: 정수, 실수
- 객체: { }
- 문자열: ""
  - 반드시 큰 따옴표를 사용해야 함
  - 속성명도 반드시 큰 따옴표로 기술해야 함

## ✓ 직렬화

- Javascript 객체를 문자열로 변환하는 것
- `JSON.stringify(객체)`
  - JSON 문자열 리턴

```
let obj = {  
  Influencer : {  
    name : 'Jaxon' ,  
    age : 42 ,  
    city: "New York"  
  }  
}
```

**let str = JSON.stringify(obj)**



```
{ "Influencer" : { "name" : "Jaxon" , "age" : 42 , "city" : "New York" } }
```

## ○ 사용 시점

- 클라이언트가 서버로 데이터를 전송할 때(request)
  - POST와 PUT에서 body 내용을 JSON 인코딩
- 서버가 클라이언트로 데이터를 응답할 때(response)

## ✓ 역직렬화

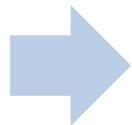
- JSON 문자열을 Javascript 객체로 변환하는 것

- JSON.parse(문자열)

- Javascript 객체 리턴

```
let str = { "Influencer" : { "name" : "Jaxon" , "age" : 42 , "city" : "New York" } }
```

```
let obj = JSON.parse(str)
```



```
{  
  Influencer : {  
    name : 'Jaxon' ,  
    age : 42 ,  
    city: "New York"  
  }  
}
```

- 사용 시점

- 서버가 POST/PUT 요청을 받고, Body를 해석할 때
- 클라이언트가 서버로부터 데이터를 수신한 후 이를 처리할 때

## ✓ 직렬화/역직렬화 작업

- 일반적으로 통신 라이브러리(axios)에서 처리해 줌

## ✓ form 요소 데이터의 인코딩 방식

- **enctype** 속성으로 설정
  - application/x-www-form-urlencoded
    - 브라우저가 전송할 때 사용할 때 디폴트 방식  
키1=값1&키2=값2&...
  - application/json
    - Javascript로 전송할 때 사용  
{ "키1" : 값1 , "키2" : 값2, ... }
  - multipart/form-data
    - 첨부파일과 같은 바이너리 데이터를 전송할 때 반드시 지정