

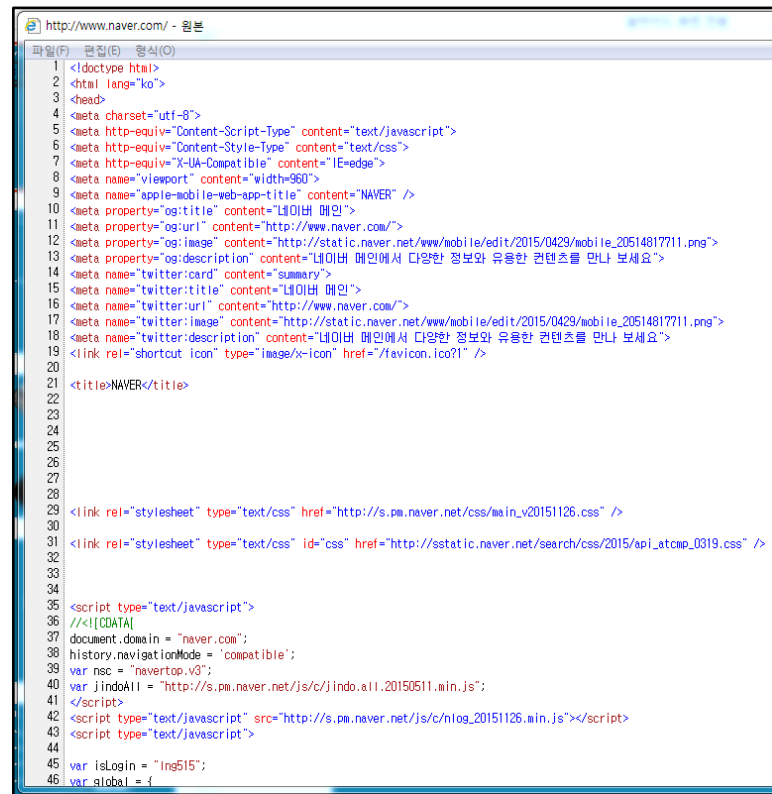


**{비동기 통신}**



# Ajax 왜 필요한가?

<http://www.naver.com/>





# Ajax 왜 필요한가?

```
http://www.naver.com/ - 원본
파일(F) 편집(E) 형식(O)
1 <!doctype html>
2 <html lang="ko">
3 <head>
4 <meta charset="utf-8">
5 <meta http-equiv="Content-Script-Type" content="text/javascript">
6 <meta http-equiv="Content-Style-Type" content="text/css">
7 <meta http-equiv="X-UA-Compatible" content="IE=edge">
8 <meta name="viewport" content="width=960">
9 <meta name="apple-mobile-web-app-title" content="NAVER" />
10 <meta property="og:title" content="네이버 메인">
11 <meta property="og:url" content="http://www.naver.com/">
12 <meta property="og:image" content="http://static.naver.net/www/mobile/edit/2015/0429/mobile_20514817711.png">
13 <meta property="og:description" content="네이버 메인에서 다양한 정보와 유용한 콘텐츠를 만나 보세요">
14 <meta name="twitter:card" content="summary">
15 <meta name="twitter:title" content="네이버 메인">
16 <meta name="twitter:url" content="http://www.naver.com/">
17 <meta name="twitter:image" content="http://static.naver.net/www/mobile/edit/2015/0429/mobile_20514817711.png">
18 <meta name="twitter:description" content="네이버 메인에서 다양한 정보와 유용한 콘텐츠를 만나 보세요">
19 <link rel="shortcut icon" type="image/x-icon" href="/favicon.ico?1" />
20
21 <title>NAVER</title>
22
23
24
25
26
27
28
29
30 <link rel="stylesheet" type="text/css" href="http://s.pw.naver.net/css/main_v20151126.css" />
31 <link rel="stylesheet" type="text/css" id="css" href="http://sstatic.naver.net/search/css/2015/apj_atcmp_0319.css" />
32
33
34
35 <script type="text/javascript">
36 //
37 document.domain = "naver.com";
38 history.navigationMode = 'compatible';
39 var nsc = "navertop.v3";
40 var jindoAll = "http://s.pw.naver.net/js/c/jindo.all.20150511.min.js";
41 &lt;/script&gt;
42 &lt;script type="text/javascript" src="http://s.pw.naver.net/js/c/nlog_20151126.min.js"&gt;&lt;/script&gt;
43 &lt;script type="text/javascript"&gt;
44
45 var isLogin = "lng515";
46 var global = {</pre></div><div data-bbox="508 398 881 517" data-label="Text"><p>서버로부터 응답받는 페이지의<br/>중복되는 코드발생</p></div><div data-bbox="525 625 857 681" data-label="Text"><p>네트워크 대역폭 자원 낭비</p></div>
```



# Ajax

(Asynchronous Javascript And XML)

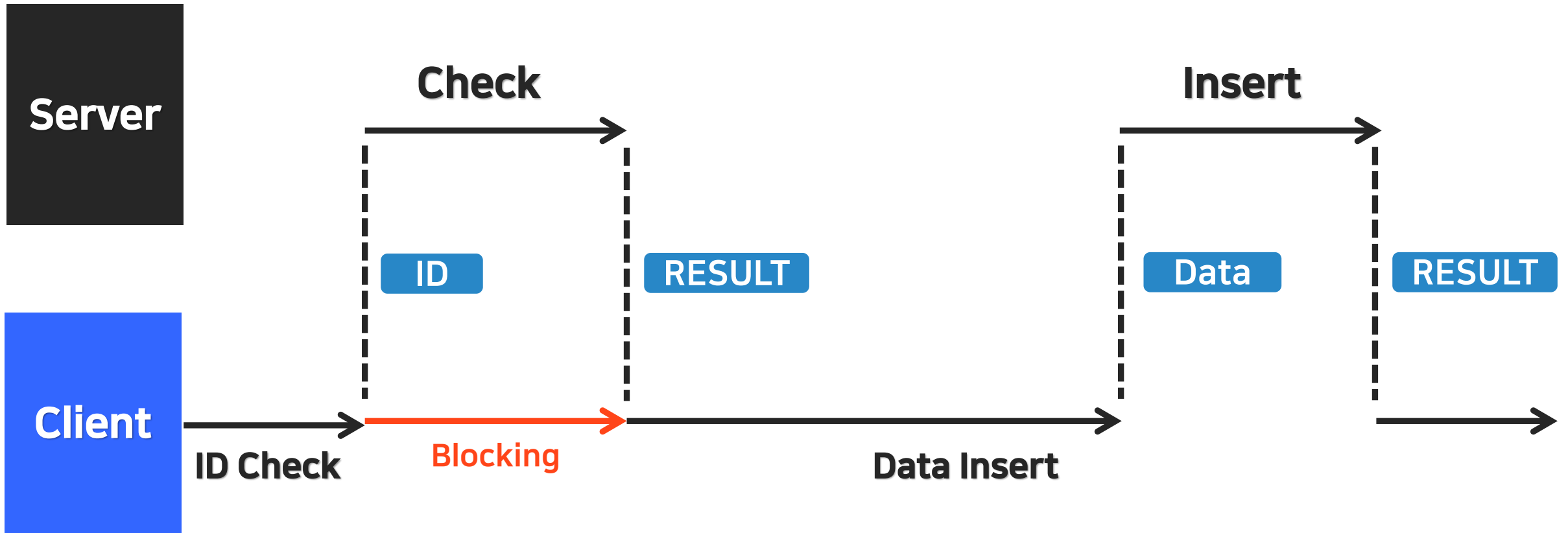
**필요한 데이터만을 웹 서버에 요청해서 받은 후**  
**클라이언트에서 데이터에 대한 처리를 진행하는**  
**비동기적인 웹 어플리케이션을 제작하기 위한 웹 개발 기법**

# Ajax

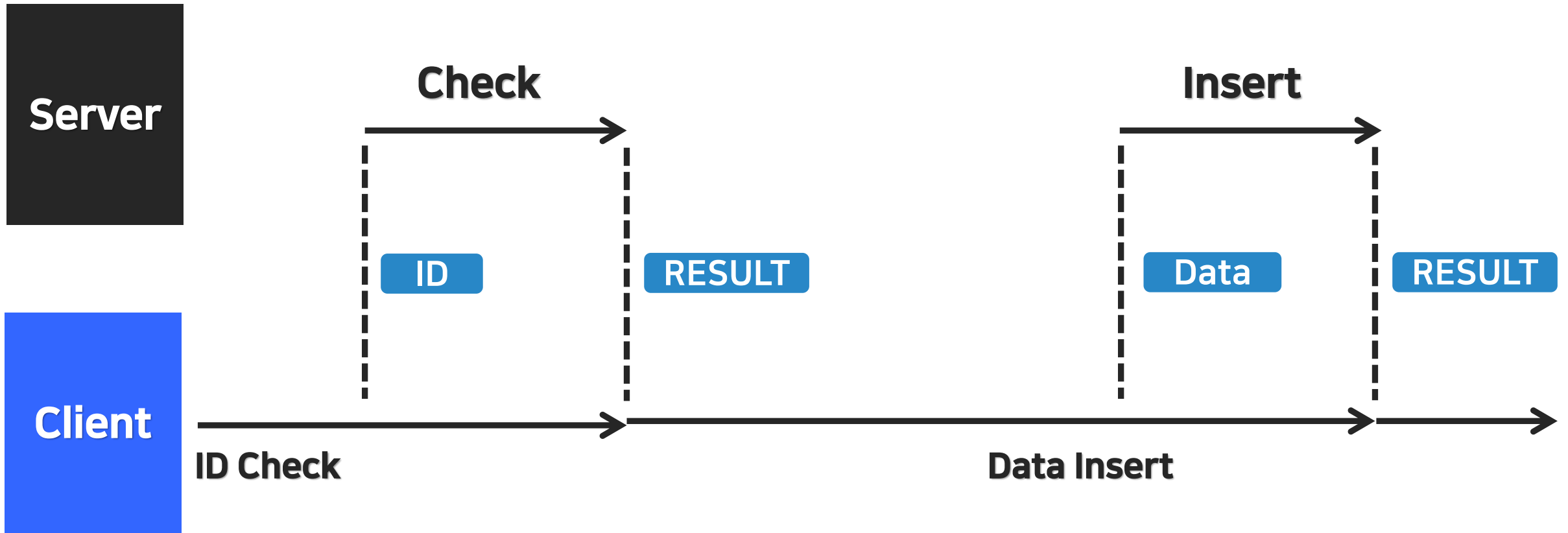
(Asynchronous Javascript And XML)

웹 문서 전체를 다시 불러오지 않고 일부만 가져오는 방식

## 동기식 통신



## 비동기식 통신



사용자 입장에서는 화면갱신도 없고,  
요청과 응답 사이 시간에도 다른 일을 할 수 있어  
편리하고 빠른듯한 환경을 느낄 수 있다.



```
$.ajax({  
    url : "http://.....",
```

←----- Server URL

←----- 전송방식

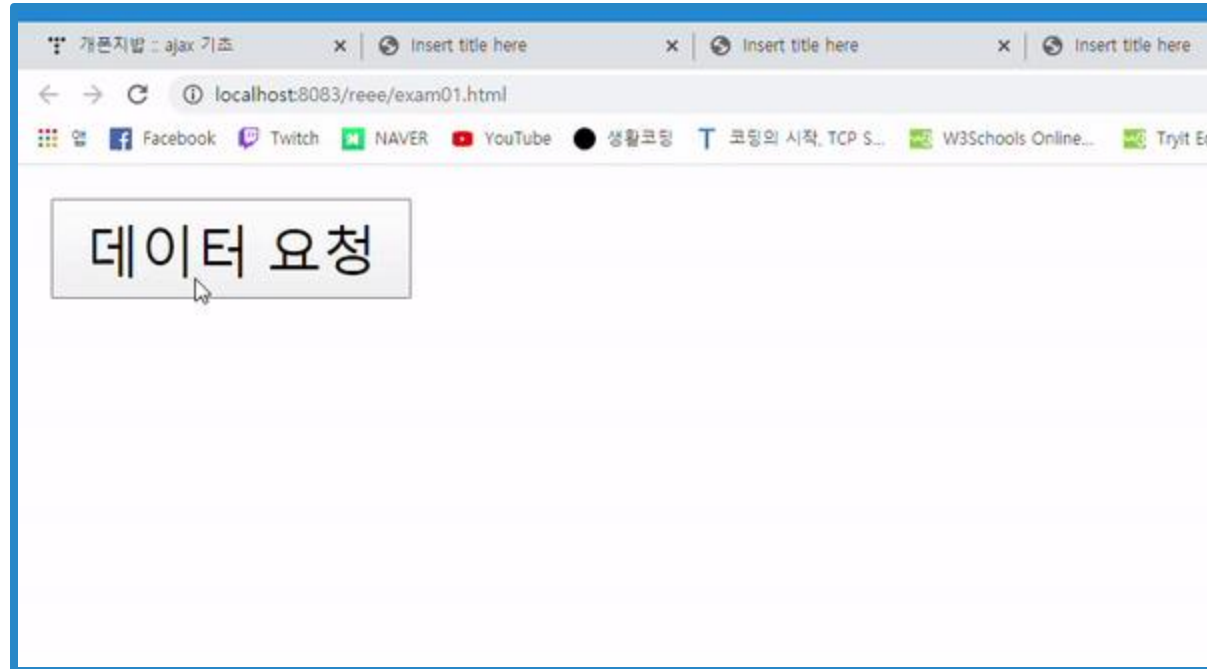
←----- 전송할 데이터

←----- 성공 callback 함수

←----- 실패 callback 함수

```
});
```

속성	설명
type	http 전송 method를 지정한다. ( GET, POST )
url	요청 URL. GET 방식일경우 URL 뒤에 파라미터를 붙여서 사용해도 된다.
dataType	Ajax를 통해 호출한 페이지의 Return 형식이다. 형식에 따라 xml, json, html, text 등을 사용하면 된다.
data	서버에 데이터를 전송할 때 사용된다.
error	에러났을때의 처리 부분이다.
success	성공했을때의 처리 부분이다. 보통 해당 부분에서 데이터 핸들링을 하면 된다.



Ajax 데이터 전송시 사용하는 형식은  
**CSV, XML, JSON** 방식이 있다.

사과, 바나나, 포도, 오렌지  
**용량이 적지만 가독성이 떨어진다**

```
<weeklyBoxOffice>
  <rnum>1</rnum>
  <rank>1</rank>
  <rankInten>0</rankInten>
  <rankOldAndNew>OLD</rankOldAndNew>
  <movieNm>미션임파서블:고스트프로토콜</movieNm>
  <openDt>2011-12-15</openDt>
  <salesAmt>7840509500</salesAmt>
  <salesShare>35.8</salesShare>
  <salesInten>-1706758500</salesInten>
</weeklyBoxOffice>
```

태그를 사용해 가독성이 좋지만  
데이터의 기본 부파가 크다



# JSON

(JavaScript Object Notation)

**속성-값 쌍으로 이루어진 데이터를  
전달하기 위한 개방형 표준 포맷**



# JSON

(JavaScript Object Notation)

자바스크립트 객체 표기법을 사용하는 텍스트 형식의 데이터





# JSON 개념 및 구조

## | JSON 형식

이름은 반드시 큰 따옴표(" ")를 사용해야 한다.

```
{  
  "이름": "값",  
  ...  
}
```

값은 문자열, 숫자, 논리, 배열, 객체를 정의할 수 있다.

ex)

문자열 -> "Hello"

숫자 -> 0, 10, 23

논리 -> true or false

배열 -> [1,2,3,4,5]

객체 -> { "이름": "값" }

**{**

**"name": "홍길동",**

**"age": 100,**

**"weight": 70.5,**

**"baby": false,**

**"money": null**

**}**

- 문자열
- 숫자(정수)
- 숫자(실수)
- true/false
- null

```
{  
  "info" : {"name": "홍길동", "age": 12},  
  "work" : 20  
}
```

→ JSON Object

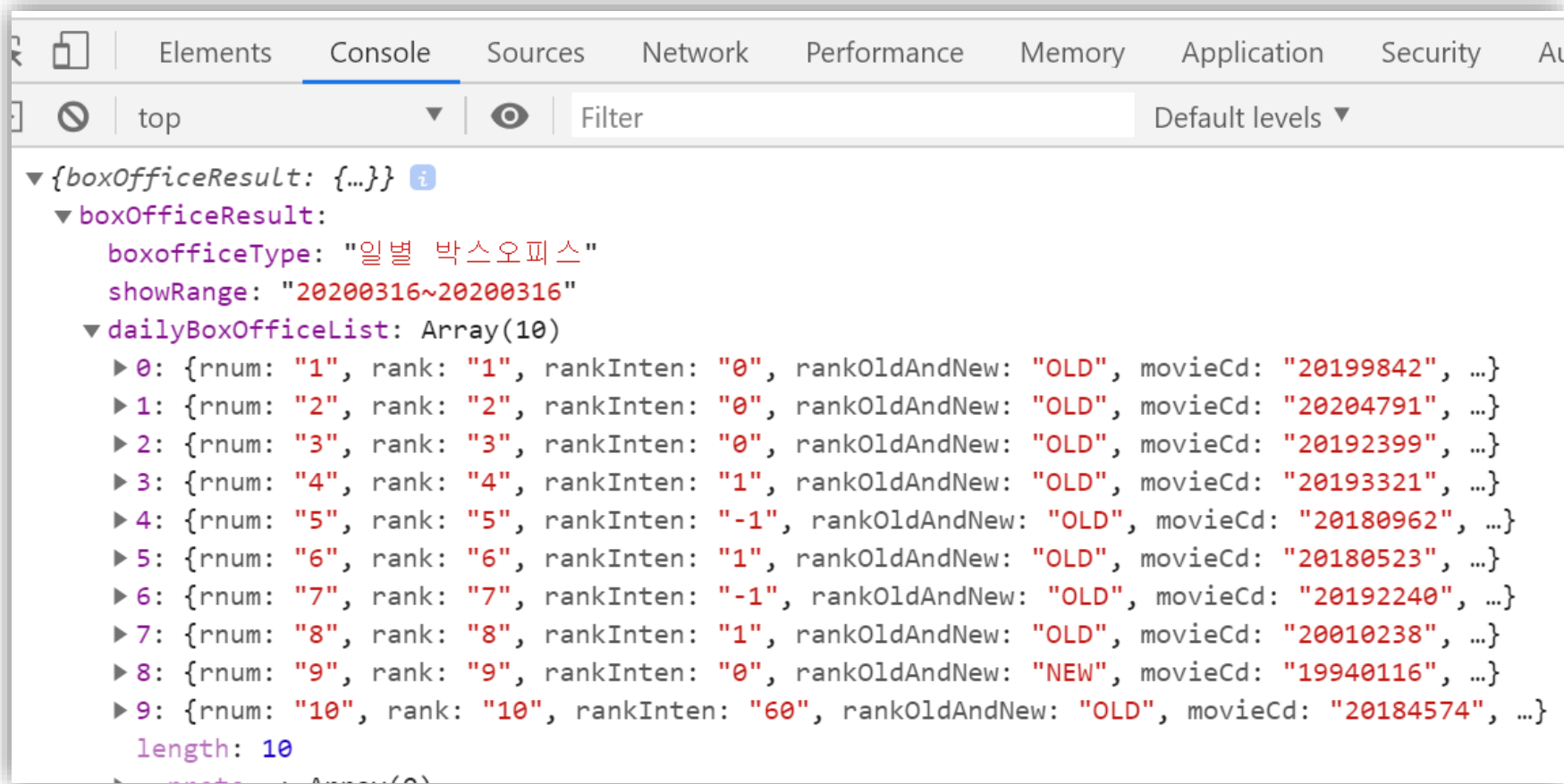
[  
 {"name": "홍길동", "age": 12},  
 {"name": "김수지", "age": 20},  
 {"name": "박병관", "age": 20},  
 {"name": "차현석", "age": 35}  
]

→ JSON Object

## 제공 서비스

영화관입장권통합전산망이 제공하는 오픈API 서비스 모음입니다.  
사용 가능한 서비스를 확인하고 서비스별 인터페이스 정보를 조회합니다.

 <b>1</b> 박스오피스	▪ 일별 박스오피스    ▪ <a href="#">주간/주말 박스오피스</a>
 <b>2</b> 공통코드조회	▪ 공통코드 조회
 <b>3</b> 영화정보	▪ 영화목록    ▪ 영화 상세정보
 <b>4</b> 영화사정보	▪ 영화사목록    ▪ 영화사 상세정보
 <b>5</b> 영화인정보	▪ 영화인목록    ▪ 영화인 상세정보



Open API를 통해 영화 데이터를 웹 페이지에 출력하시오.

순위	영화명	개봉일
1	천박사 퇴마 연구소: 설경의 비밀	2023-09-27
2	1947 보스톤	2023-09-27
3	거미집	2023-09-27
4	더 년 2	2023-09-27
5	극장판 엉덩이 탐정: 미스터리 가면 ~최강의 대결	2023-09-28
6	잠	2023-09-06
7	가문의 영광: 리턴즈	2023-09-21
8	오픈하이머	2023-08-15
9	아르고 원정대: 꼬마 영웅 패티의 대모험	2023-09-21
10	80일간의 세계일주	2023-09-27



# API

(Application Programming Interface)

프로그램과 프로그램 사이를 연결하여 데이터를 주고 받을 수 있도록 약속된 규칙





# fetch()

Client와 Server 사이에 데이터를 주고받기 위해  
브라우저에서 제공하는 비동기 통신 함수



# `try{...}catch(error){...}`

JavaScript에서 예외처리를 위한 구문



# await

비동기 함수가 처리될 때까지 기다렸다가 다음 코드를 실행하게 해주는 키워드  
비동기 함수를 동기 함수처럼 작동하게 해준다  
(ex. `await axios.get(url)`)



# async

해당 함수가 비동기 작업을 포함하고 있음을 명시하기 위한 키워드  
함수의 선언 앞에 붙여서 사용 (함수 내부에 await 키워드가 있는 경우)  
(ex. const func = **async** () => {})