

---

# Ajax

## (Asynchronous JavaScript & XML)

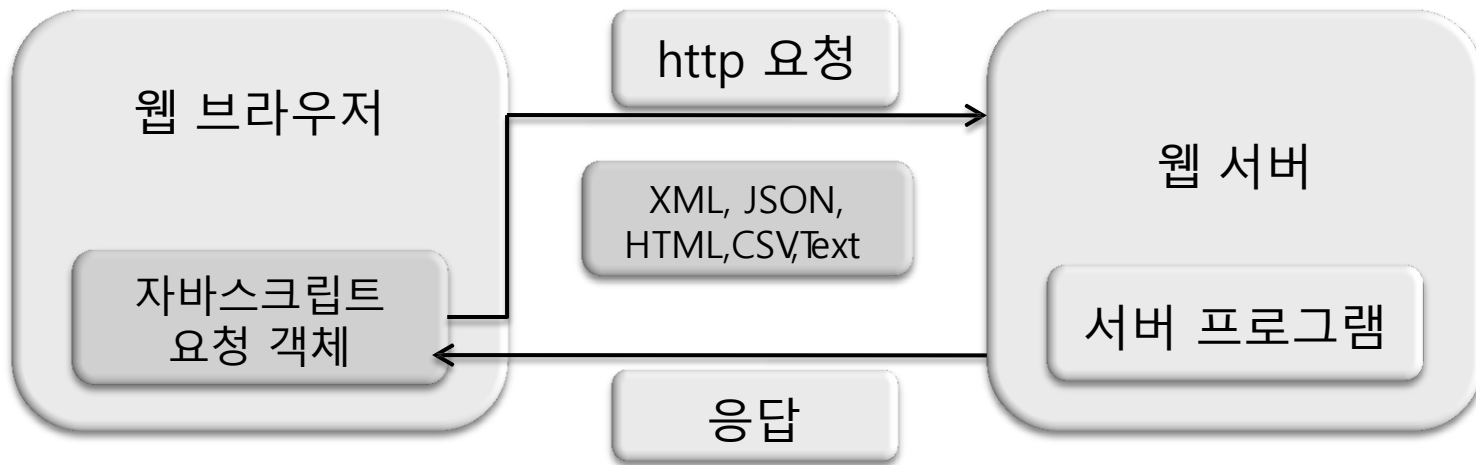
## Ajax란? (1/2)

---

- ▶ Asynchronous JavaScript & XML
  - Java Script에 의한 비동기적인(Asynchronous) 통신으로 XML기반인 데이터를 클라이언트인 웹 브라우저와 서버 사이에서 교환하는 방식 의미
  - 주 목적
    - ▶ 웹 어플리케이션을 데스크톱 어플리케이션처럼 빠른 응답을 갖는 구조로 개발하고자 함

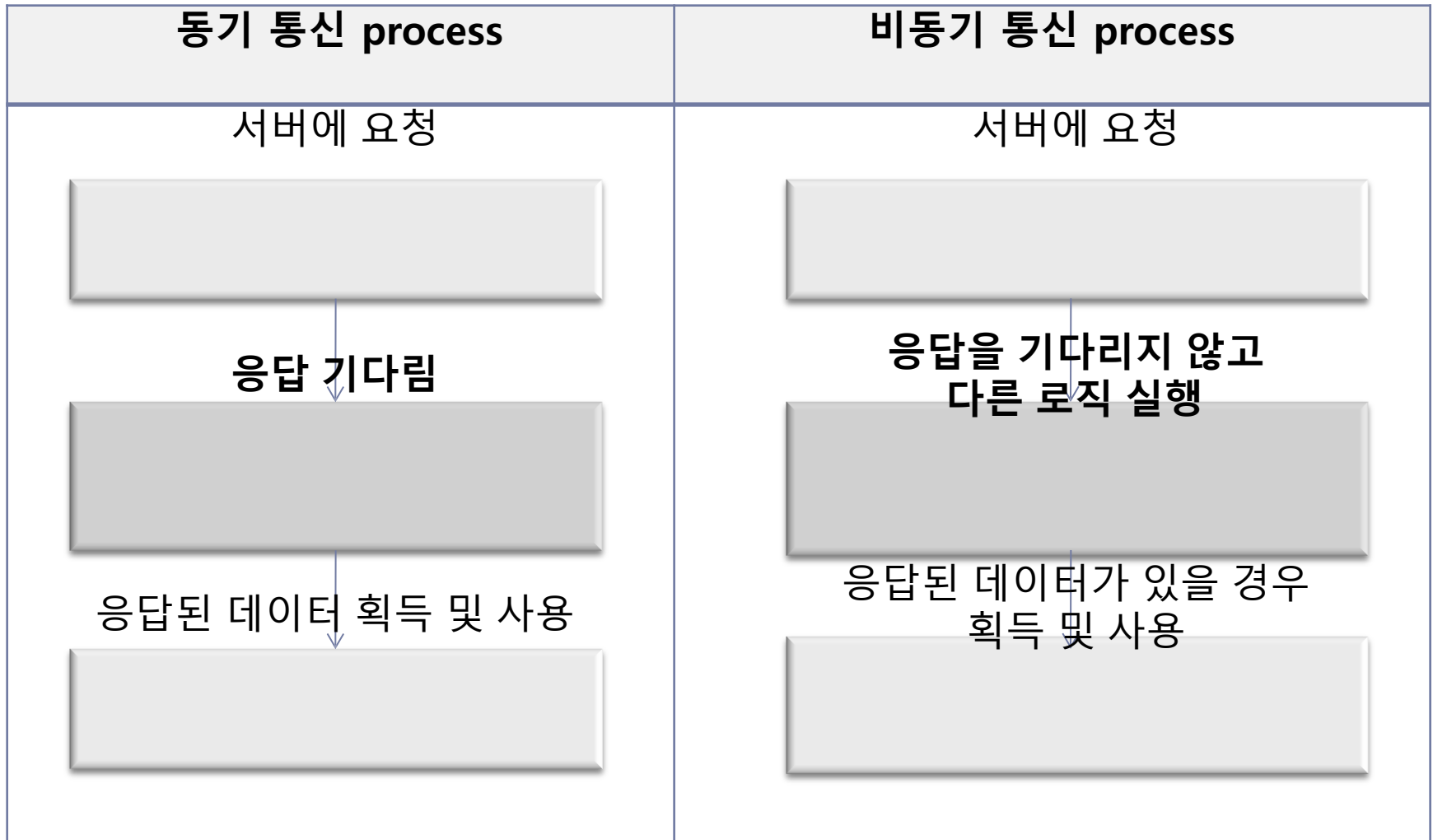
## Ajax란? (1/2)

---



- ▶ 자바스크립트로 HTTP 요청을 보내고 XML, JSON, Text 등으로 응답받아 사용하는 비동기 통신 개발 기술

# 동기 통신 : 비동기 통신



# 동기 통신 : 비동기 통신

동기 통신 특징	비동기 통신 특징
<p>발생 가능한 문제점</p> <ul style="list-style-type: none"> <li>- 서버의 응답이 지연될 경우 그 시스템에서 계속 딜레이가 발생</li> </ul>	<p>개발 코드</p> <ul style="list-style-type: none"> <li>- 응답을 기다리지 않고 다른 일을 할 수도 있기 때문에 비 동기적으로 응답 이벤트를 처리할 수 있는 핸들러 개발</li> </ul> <p>웹 브라우저</p> <ul style="list-style-type: none"> <li>- 서버로 부터 전체 페이지를 반환 받지 않기 때문에 페이지의 이동 및 새로고침 현상이 없음</li> </ul> <p>발생 가능한 문제점</p> <ul style="list-style-type: none"> <li>- 웹 페이지와 서버 간에 벌어지는 대부분의 상호 작용들은 사용자에게 보이지 않음. 100% 신뢰 할수 없다는 단점</li> <li>- 요청 후 응답 유무와 상관없이 다른 로직을 실행할 수 있기 때문에 순차적으로 무언가 하기에는 부적합</li> </ul>

# Ajax 활용 분야

- ▶ 페이지의 이동 없이 웹 브라우저의 화면을 동적으로 변경
- ▶ 사례1 - 포털 사이트 검색 시 데이터 실시간 전송
  - 검색하고자 하는 단어의 일부 입력시 입력된 일부 단어로 시작되는 검색어가 드롭다운 박스에 표시
- ▶ 사례2 - Google의 지도 서비스인 GoogleMap
  - 페이지 이동 없이 마우스 조작만으로 지도 표시 영역의 변경이 가능



# Ajax의 장점

---

- ▶ 웹 어플리케이션을 데스크톱 어플리케이션처럼 빠르고 편한 응답을 갖는 구조로 개발이 가능
- ▶ 페이지의 이동 없이 웹 브라우저의 화면을 동적으로 변경
- ▶ 사용자가 서버로 요청 후 요청된 데이터가 응답되기 전에도 다른 작업을 할 수 있음
- ▶ 응답 형식이 데이터만 들어가 있는 XML, JSON, Text 형식이므로 전통적인 웹 애플리케이션에 비해서 서버 측 처리 속도도 빠르고 전송 데이터양도 감소
- ▶ 응답으로 받은 데이터들을 자바스크립트로 조작 및 검색하거나 변환을 할 수 있고, 작업 실행 속도도 무척 빠름
- ▶ 불필요한 데이터 요청을 최소화할 수 있고, 많은 일이 클라이언트인 웹 브라우저에서 처리될 수 있기 때문에 서버측 부담 감소

## Ajax 단점

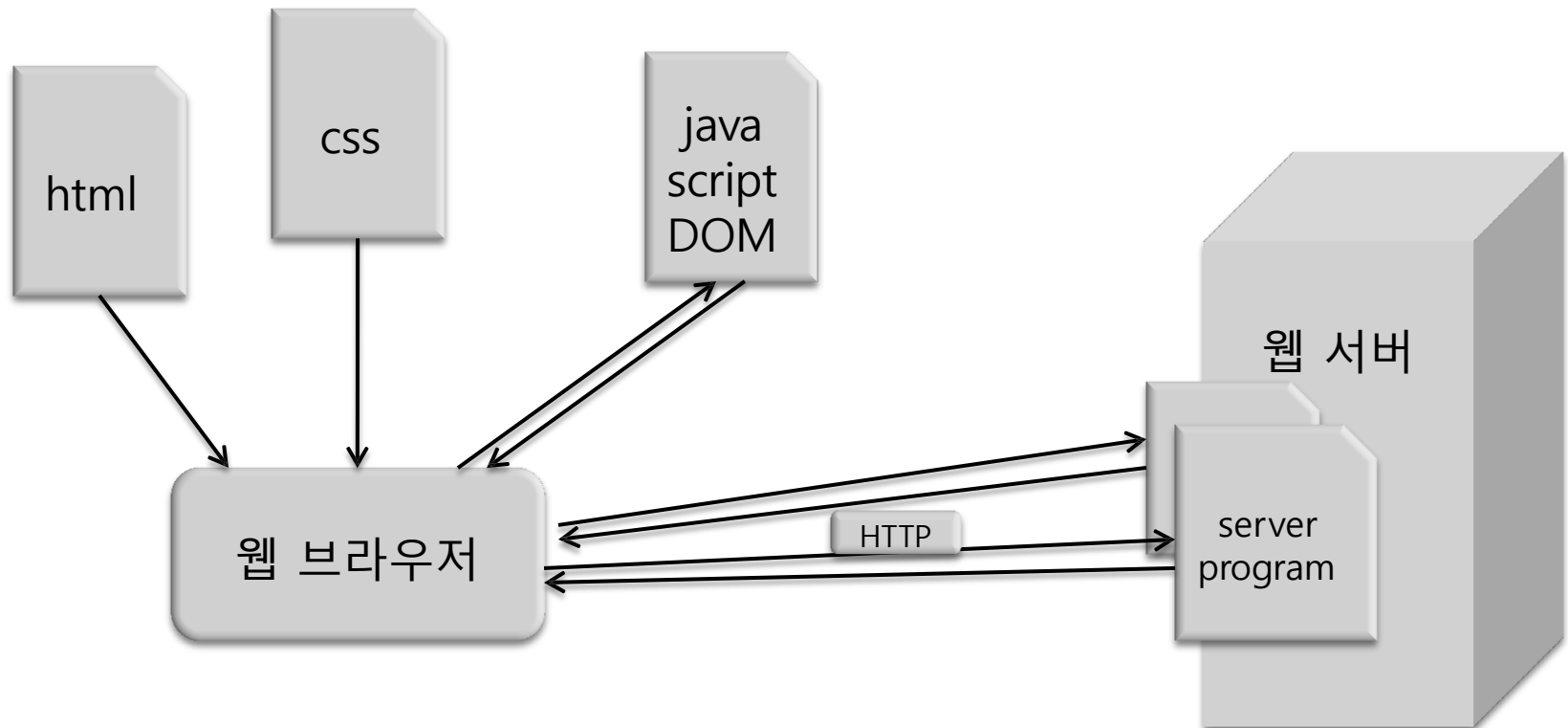
---

- ▶ Ajax가 포함된 HTML 페이지가 속한 서버가 아닌 다른 서버로 요청을 보낼 수 없음(cross-origin)
- ▶ 클라이언트 PC의 파일에 접근할 수 없음



# Ajax 적용 기술

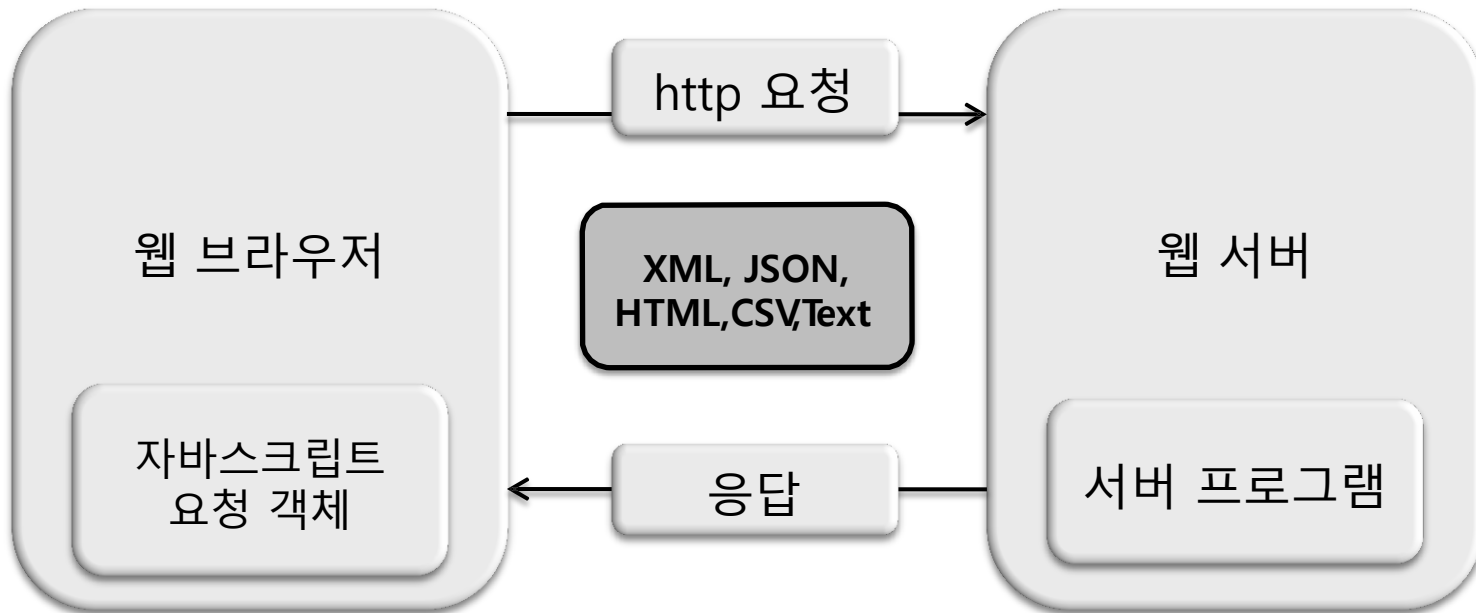
- ▶ 이미 사용되고 있는 기술들을 모아서 재구성
- ▶ HTML+CSS, DOM, XMLHttpRequest, JavaScript, 프로그래밍 언어등의 기술을 통합해서 사용되어 지는 기술



---

# Ajax Data

## Ajax에서 사용되는 데이터 형식(1/2)



## Ajax에서 사용되는 데이터 형식(2/2)

데이터 포맷	설명
XML	확장 가능한 마크업 언어로 데이터를 구조화 할 수 있는 언어
JSON	JavaScript를 기반으로 만들어졌으나, 프로그래밍 언어 및 이종 시스템간 데이터 교환시에 사용 할 수 있는는 이상적인 text 포맷
HTML	웹 브라우저상에 문자나 이미지 등을 표현하는 마크업 언어
CSV	데이터를 , 표기를 기준으로 구분한 text 데이터
TEXT	일반 문자열

# JSON의 개요

---

- ▶ JavaScript Object Notation
- ▶ JSON 은 텍스트 기반의 경량(lightweight) 데이터 변환 포맷
- ▶ 사용자가 읽고, 쓰기 쉬워 데이터 처리가 쉬움
- ▶ JavaScript를 기반으로 만들어졌으나, 프로그래밍 언어에 독립적인 text 형식
- ▶ 이기종간의 데이터 교환에 적합

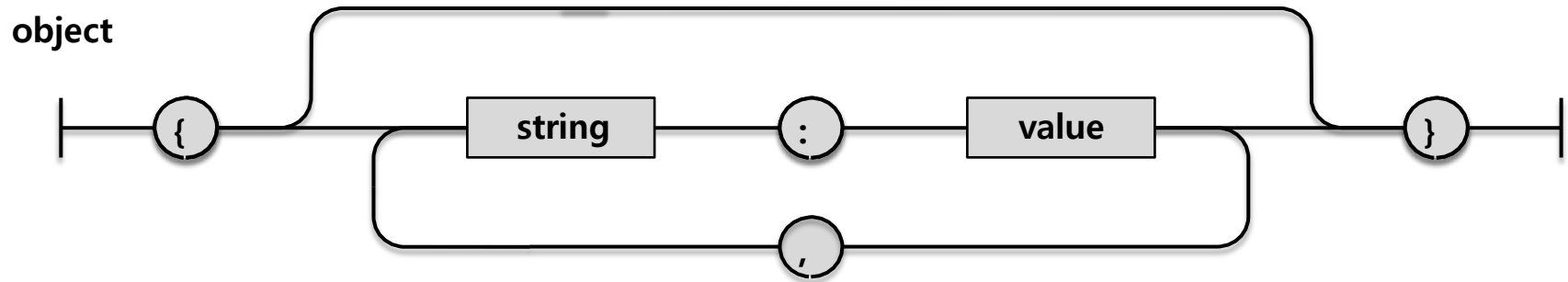
## JSON 데이터 형식(1/4)

---

- ▶ 컬렉션데이터 구조
  - 이름/값 쌍(pair)으로 이루어진 데이터 집합
  - 중괄호 내에 key와 value 로 구성
  - key와 value 사이에는 : 표기로 구분
  - key는 문자열 타입, value는 JSON 데이터
  - 다수의 데이터 존재할 경우 key와 value 값들은 , 표기로 구분

## JSON 데이터 형식(2/4)

### ▶ 컬렉션데이터 구조



```
{  
  "name1" : value1,  
  "name2" : value2,  
  ...  
}
```

문법

```
{  
  "id" : id,  
  "pw" : pw  
}
```

예 1

```
{  
  "conversation": { "decimal": "103",  
                    "hyper": "&0x67"  
  }  
}
```

예 2

## JSON 데이터 형식(3/4)

---

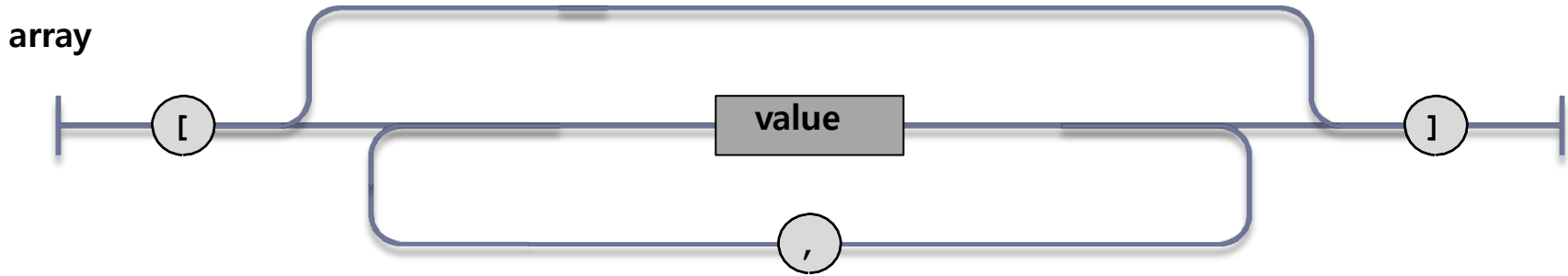
### ▶ 배열 구조

- 순서가 있는 데이터들의 목록
- [(left bracket) 으로 시작해서 ](right bracket)으로 표현
- 각 데이터들은 , 표기로 구분



# JSON 데이터 형식(4/4)

## ▶ 배열 구조



[value1, value2, ...]

문법

var arrayName=[id, pw]

예 1

```
[  
  [0, -1, 0],  
  [1, 0, 0],  
  [0, 0, 1]  
]
```

예 2

## JSON 데이터 사용(1/2)

---

### ▶ eval() 함수

- JSON 형식의 문자열 데이터를 자바스크립트 객체로 변환
- 이 함수 생략시 접근자( . ) 및 [ ]로 특정 데이터 access 불가
- 문법

```
eval( "(" + JSON형식의 데이터 + ")" );
```

## JSON 데이터 사용(2/2)

- ▶ [ ] 브래킷 연산자 와 . 접근자를 활용하여 JSON 데이터 access
  - [] 연산자
    - ▶ 예 : 변수['key']
  - . 접근자
    - ▶ 예 : 변수.key

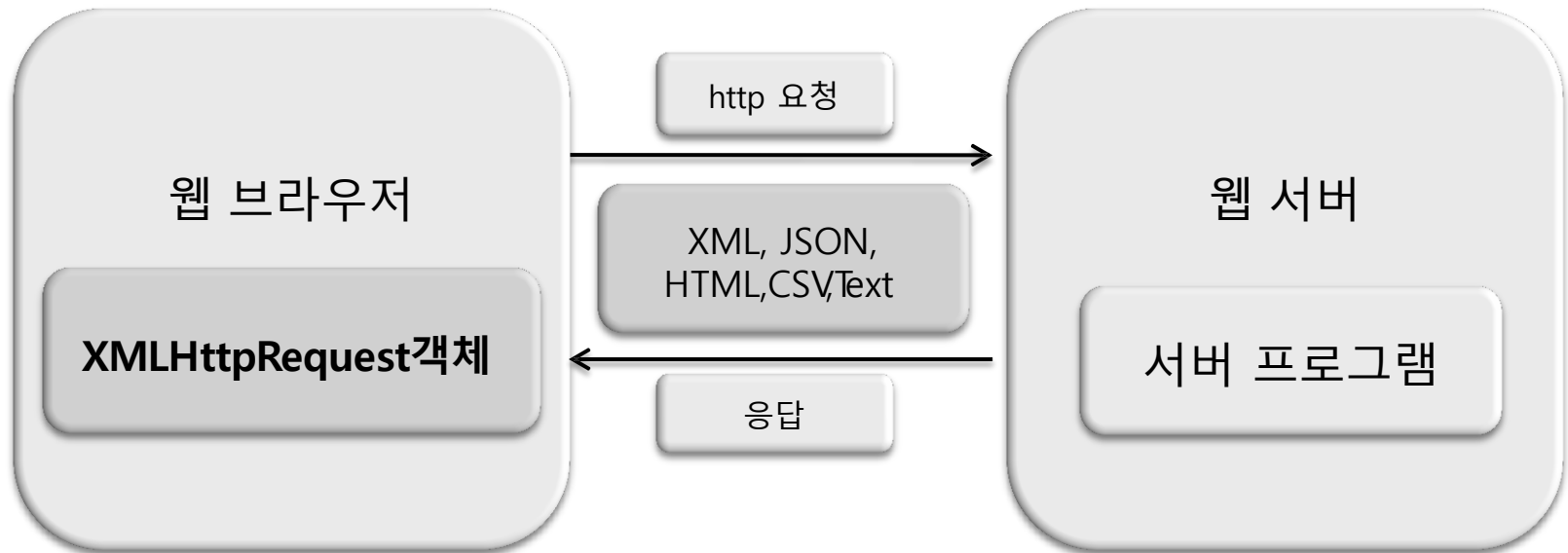
```
var jsonData = eval( "({ ko:'대한민국', fr:'프랑스', en:'미국', koValue:100, frValue:50 })" );  
  
var firstCountry = jsonData['ko']; //대한민국  
var frValue = jsonData.frValue; //50
```

---

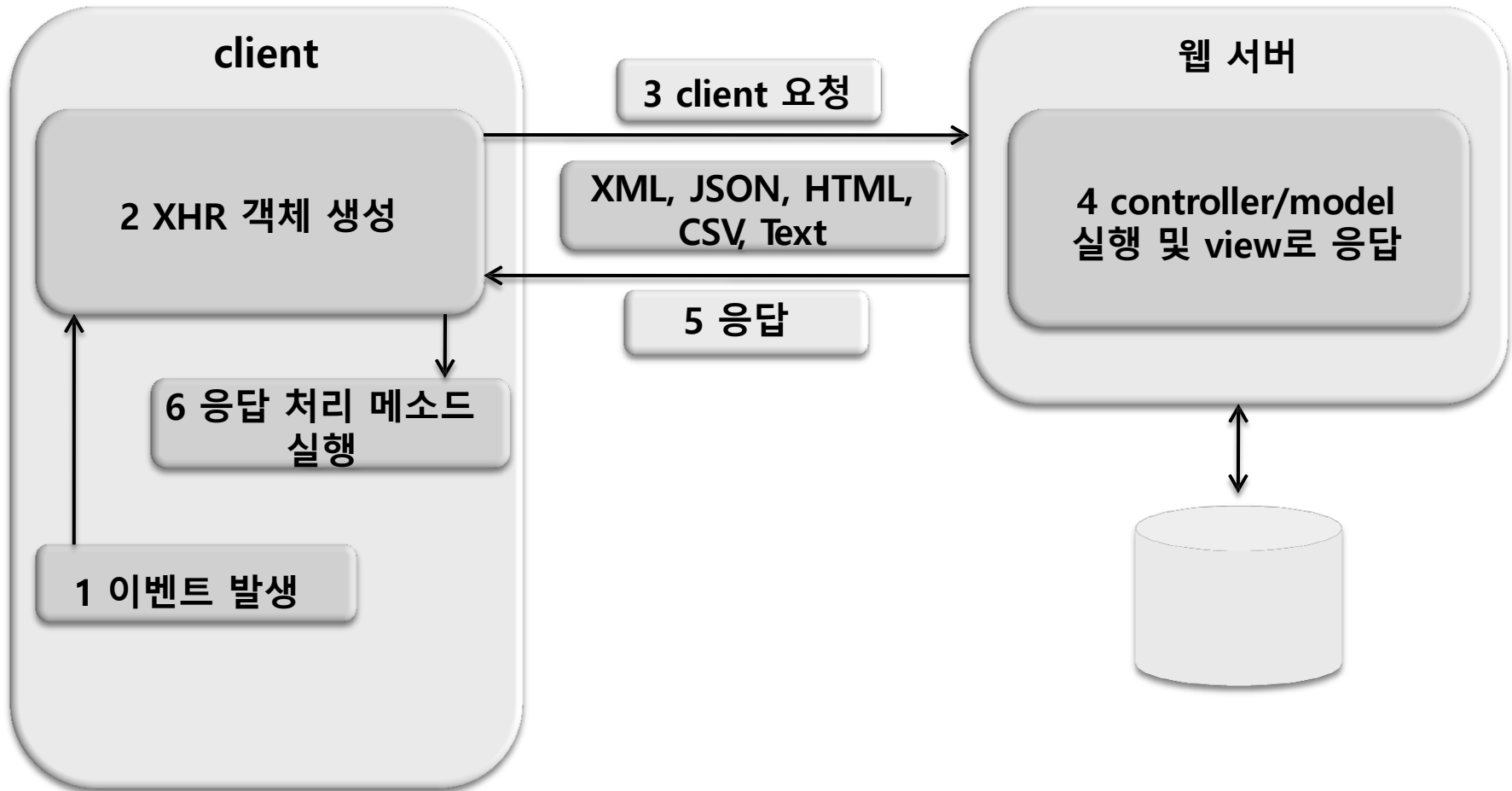
# Ajax Programming

## Ajax 통신 원리

- ▶ 폼 전송 방식이 아닌 자바스크립트 객체 (XMLHttpRequest)를 이용한 요청과 응답 처리
- ▶ 웹 브라우저가 비동기 요청 받으면 콜백 함수 실행



# Ajax 실행 process

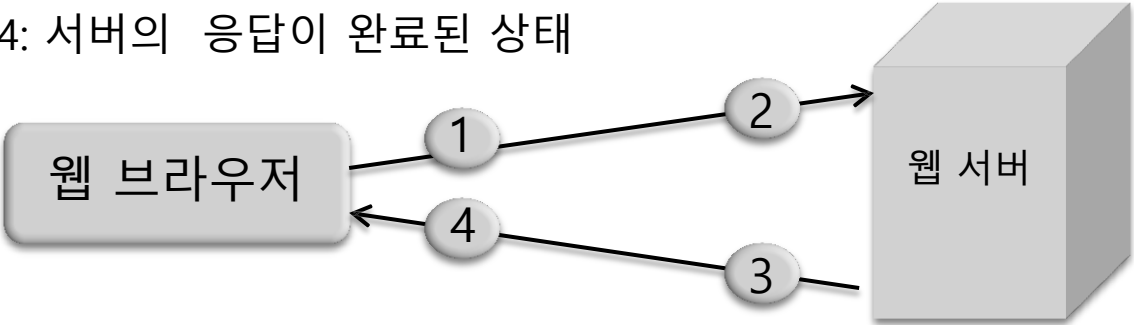


# XMLHttpRequest(XHR) 객체의 개요

---

- ▶ Ajax에서 서버와 클라이언트 간에 통신을 담당하는 주요 객체
- ▶ 요청과 응답을 처리하는 필수 객체
- ▶ 전체 페이지의 새로 고침 없이 서버와의 통신을 가능하게 함
- ▶ 기원
  - ActiveX의 구성요소 중의 하나인 웹 브라우저인 인터넷 익스플로러(IE)에서 가장 먼저 구현
  - W3C의 비표준
    - ▶ 웹 브라우저별, 브라우저 버전별 객체 생성 및 사용하는 방법이 다름  
(크로스 브라우저 개발 방법 적용 필요)

# XHR 객체의 속성(1/3)

속성	설명	읽기/쓰기
readyState	<p>AJAX 객체의 요청, 응답의 상태를 나타내는 숫자</p> <p>-로딩중, 처리중, 처리완료등의 상태 표현</p> <p>0: request 가 초기화 되지 않음(open() 호출 전)</p> <p>1: request 가 셋업되고 보내지 않음(send() 호출 전)</p> <p>2: request 가 보내지고, 진행 중 일 경우(send() 호출후)</p> <p>3: response 에서 부분적인 데이터를 받으며, 완료가 아직 안 된 경우</p> <p>4: 서버의 응답이 완료된 상태</p>  <pre> graph LR     Browser[웹 브라우저] -- 1 --&gt; Server[웹 서버]     Server -- 2 --&gt; Browser     Browser -- 3 --&gt; Server     Server -- 4 --&gt; Browser     </pre>	읽기 전용



## XHR 객체의 속성(2/3)

속성	설명	읽기/쓰기
status	서버로부터 응답된 HTTP 상태 코드를 나타내는 숫자 200 : 정상적으로 응답을 받은 경우 404 : 페이지를 찾지 못한 경우 405 = 요청 방식 다름 500 = 서버 오류 발생	읽기 전용
onreadystatechange	요청에 대한 응답을 받을때마다(readyState 속성이 바뀔때마다) 호출됨 콜백 함수 지정 속성	읽기/쓰기

## XHR 객체의 속성(3/3)

속성	설명	읽기/쓰기
statusText	서버로부터 받은 응답의 상태를 나타내는 문자열 정상적으로 응답을 받으면 'OK' 파일을 찾지 못하면 'Not Found'	읽기 전용
responseText	서버로부터 응답된 문자열 데이터를 보유	읽기 전용
responseXML	서버로부터 응답된 xml 데이터를 보유 DOM 객체로 파싱할 수 있음	읽기 전용

## XHR 객체의 메소드(1/2)

메소드	설명
<code>void open(String method, String url, [boolean async])</code>	<p>AJAX 요청을 초기화하면서 요청 방식, 주소, 동기화 여부를 지정</p> <p>[매개변수]</p> <ol style="list-style-type: none"><li>1. method<ul style="list-style-type: none"><li>- http 요청 방식을 나타내며 "get" 또는 "post" 방식을 사용</li></ul></li><li>2. url<ul style="list-style-type: none"><li>- 요청할 페이지의 주소를 지정</li></ul></li><li>3. aysnc<ul style="list-style-type: none"><li>- 비동기 통신 여부를 나타내며 true 또는 false</li><li>- default값 true</li></ul></li></ol>

## XHR 객체의 메소드(2/2)

메소드	설명
void send(Object content)	사용자의 요청을 서버에 보냄 인자에는 요청과 함께 서버로 보낼 내용을 지정 GET방식 = null POST방식 = 쿼리스트링
String getAllResponseHeaders()	응답을 받은 모든 헤더 정보를 문자열로 반환
String getResponseHeader (String header)	응답을 받은 경우 header 인자로 지정한 이름의 헤더 정보 값을 문자열로 반환
void setResonseHeader (String header, Sring value)	요청을 보내기 전에 header 헤더 정보의 값을 value로 설정
void abort()	send() 메소드로 보낸 현재의 요청을 중단