

AJAX

▶ AJAX

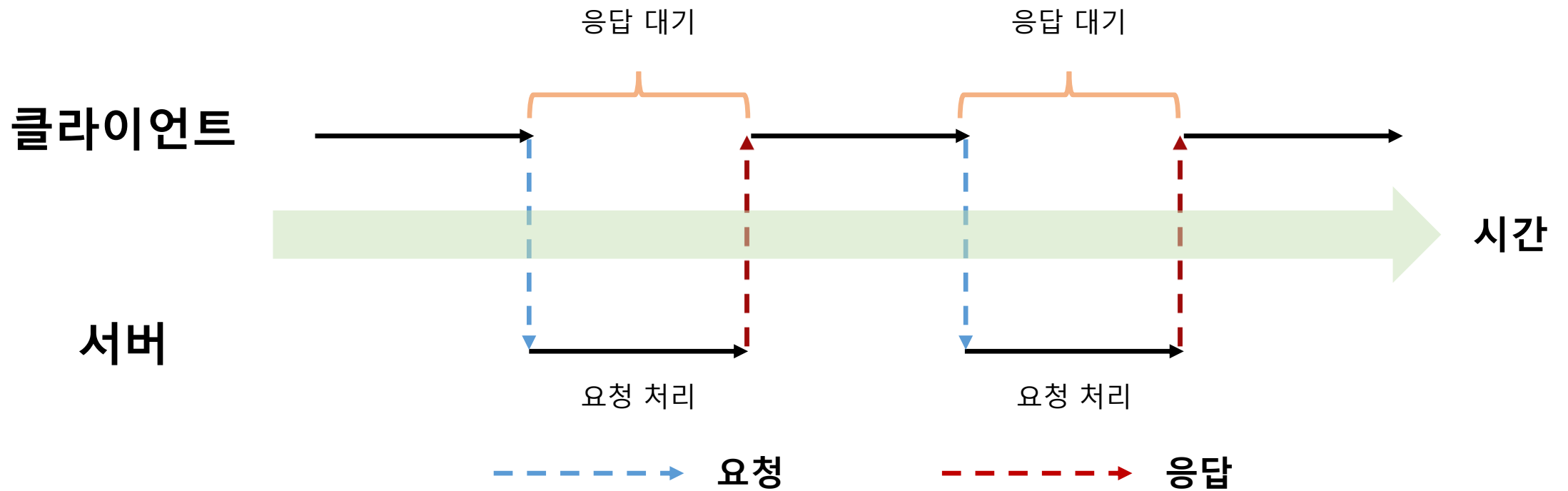
✓ AJAX란?

- Asynchronous JavaScript and XML의 약자
- JavaScript를 이용하여 비동기식으로 클라이언트와 서버가 데이터(XML)를 주고받는(통신) 방식
- 데이터 형식은 XML 뿐만 아닌 Text, HTML, **JSON**, CSV 등 다양한 형식 사용 가능

▶ AJAX

✓ 동기식 데이터 통신

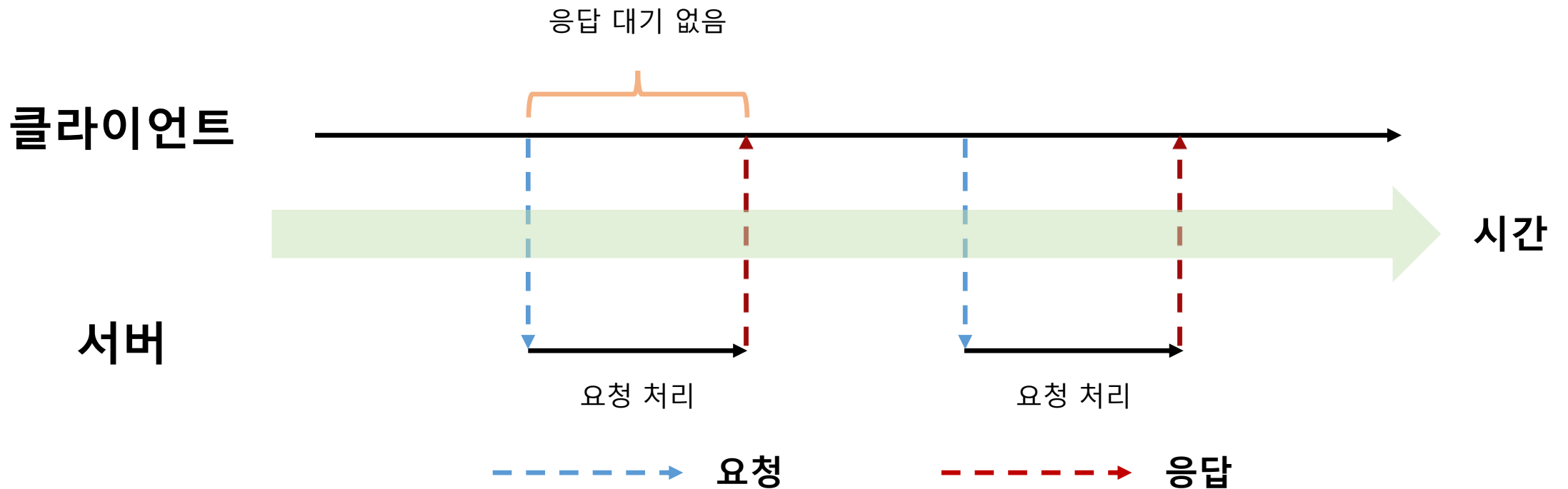
클라이언트가 서버로 데이터를 요청하면 응답이 올 때 까지 다른 작업은 대기



▶ AJAX

✓ 비동기식 데이터 통신

클라이언트가 서버로 데이터 요청 후 응답을 기다리지 않고 다른 작업 수행 가능.
추후 요청에 대한 응답이 오면 응답에 관련된 작업을 진행.



▶ AJAX 특징

- 전체 페이지를 갱신하지 않고 일부분만 업데이트 가능
- 사용자에게 즉각적인 반응과 풍부한 UI 경험을 제공 가능
- ActiveX나 플러그인 프로그램 설치 없이 이용 가능
- JavaScript 방식, jQuery 방식으로 구현 가능

▶ AJAX 단점

- AJAX는 JavaScript이므로 브라우저에 따른 크로스 브라우저 처리가 필요함
- 오픈 소스로 차별화가 어려움
- 연속적인 데이터 요청 시 서버 부하 증가하여 페이지가 느려짐
- 페이지 내 복잡도가 증가하여 에러 발생 시 디버깅이 어려움

JSON

▶ JSON

✓ JSON

- JavaScript Object Notation (자바스크립트 객체 표현법)
- 간단한 포맷
 - 괄호 {} 내에 key : value 쌍으로 구성 → { "key" : value }
 - key → 반드시 문자열 사용 (쌍 따옴표("") 표기 필수)
 - value → String, Number, Boolean, Array, Object, null 데이터 저장 가능
(단, char 데이터는 저장 불가)
- 객체{} 또는 배열[] 데이터를 효율적으로 표시 가능

▶ JSON

✓ JSON 특징

- AJAX 통신에서 Object 타입의 데이터 전송 시 XML 대비 용량이 작고 속도가 빠름
→ 경량 데이터 교환 방식
- 간단한 포맷을 가지고 있어 이해하기 쉬움
- 순수 TEXT 기반
 - 구조화된 TEXT 형식
 - 대부분의 프로그래밍 언어에서 JSON 포맷 데이터를 핸들링 할 수 있는 라이브러리를 제공 → 시스템간 객체 교환에 용이

JavaScript방식 AJAX

▶ JavaScript방식 AJAX – 방법 1 : XMLHttpRequest

※ AJAX는 브라우저 내장 객체인 **XMLHttpRequest**를 이용하여 비동기식으로 데이터를 송수신함
(IE 브라우저는 ActiveXObject 객체 사용)

1. 객체 생성 → script문에 요청을 위한 **XMLHttpRequest**객체 생성
2. 서버 응답 처리 함수 생성 및 지정 → **onreadystatechange**에 함수 지정
 - 지정되는 함수에 포함될 내용
 - 1) 응답 상태 확인 → **readyState**(서버 응답 상태 확인) , **status**(Http 응답 상태 코드 확인)
 - 2) 응답 완료(서버 응답 데이터 접근) → **responseText** / **responseXML**
3. 요청 방식, 대상(서버), 동기/비동기 지정 → **open()** 메소드 호출
4. 대상(서버)에 전송 → **send()** 메소드 호출

▶ JavaScript방식 AJAX – 방법 2 : **fetch()** API

JS의 **Promise 객체**를 이용한 비동기 요청 방법으로 JS에서 기본 제공함.

※ **Promise(약속) 객체** : 비동기 요청에 대한 처리가 완료되면 바로 결과를 제공할 것이라고 약속
-> 비동기 요청 시 동시에 다른 JS 코드가 실행되고 있음

※ **fetch(가지고 오다) API** : 비동기 요청이 끝나면 응답 결과를 가져와서
현재 화면에서 다룰 수 있는 방법을 제공 하는 API

▶ JavaScript방식 AJAX – 방법 2 : **fetch()** API

[GET방식 요청 작성 방법]

`fetch("요청주소")` // 지정된 주소로 **GET**방식 비동기 요청(ajax)

// 전달하고자 하는 파라미터를 주소 뒤 **쿼리스트링**으로 추가

`.then(response => response 파싱)` // 요청에 대한 **응답 객체(response)**를 필요한 형태로 파싱

`.then(response 파싱 데이터 => {});` // 첫 번째 then에서 파싱한 데이터를 이용한 동작 작성

`.catch(e => {})` // 예외 발생 시 처리할 내용을 작성

▶ JavaScript방식 AJAX – 방법 2 : **fetch()** API

[POST, PUT, DELETE 방식 요청 작성 방법]

```
fetch("요청주소", { // 지정된 주소로 비동기 요청(ajax)
  method : "POST", // 데이터 전달 방식을 POST로 지정
  headers: {"Content-Type": "application/json"}, // 요청 데이터의 형식을 JSON으로 지정
  body : JSON.stringify( {K:V, K:V} ) // JS객체를 JSON 형태로 변환하여 Body에 추가
})
.then(response => response 파싱 ) // 요청에 대한 응답 객체(response)를 필요한 형태로 파싱
.then(response 파싱 데이터 => {}); // 첫 번째 then에서 파싱한 데이터를 이용한 동작 작성
.catch( err => {} ) // 예외 발생 시 처리할 내용을 작성
```

▶ JavaScript방식 AJAX – 방법 2 : **fetch()** API

응답 객체(response)가 사용 가능한 속성, 메서드

.status : HTTP 상태코드

- 200 번째: 응답 성공
- 400 : 파라미터가 잘못됨
- 403 : 서버로 부터 접속을 제한당함
- 404 : 요청 주소를 찾을 수 없음
- 405 : 요청 방식(method)가 일치하지 않음

.ok : 200번째 HTTP 상태코드인 경우 true, 아니면 false

.body : 응답 내용

.text() : 문자열 형태의 응답 데이터를 텍스트 형식으로 반환

.json() : JSON 형식(JS 객체 문자열 표기법) 응답 데이터를 JS 객체 형식으로 반환

jQuery방식 AJAX

▶ jQuery방식 AJAX

✓ jQuery방식 AJAX 장점

- JavaScript 방식보다 구현 방법이 간단함 → 코드 길이 감소
- 직관적이며 다양한 방법의 코딩 가능
- **크로스 브라우저 처리를 jQuery가 자동으로 해결**
→ JavaScript방식 XMLHttpRequest 객체 생성 시 브라우저 검사 생략 가능

▶ jQuery방식 AJAX

✓ jQuery방식 AJAX 구현 방법

- \$.ajax(url [, settings])

→ jQuery에 내장되어 있는 **AJAX** 통신을 위한 가장 기본적인 함수

[구현방법]

\$.ajax({

url : "요청이 전송되는 url이 포함된 문자열" // 필수 구현 속성

[,settings] // 선택 속성(다수 속성 선택 가능)

});

▶ jQuery방식 AJAX

✓ jQuery방식 AJAX 주요 속성

속성명	설명
url	요청(request) 데이터를 전송할 URL
data	서버로 전송할 요청 Parameter 설정
type	Http 요청 방식 지정 (GET / POST)
datatype	서버의 응답(response)데이터의 형식(xml,text,json,html 등)을 지정 미작성 시 자동으로 판단하여 지정.
success(data)	ajax 통신 성공 시 호출되는 함수를 지정 매개변수로 응답 데이터를 받음(data)
error	ajax 통신 실패 시 호출되는 함수를 지정
complete	ajax통신 성공여부와 관계없이 통신 완료 후 실행되는 함수 지정
async	비동기(true) / 동기(false) 지정

jQuery.ajax() API → <http://api.jquery.com/jqueryajax/>