제6장 비동기 처리와 AJAX

목차

1. 비동기 처리

- 동기Synchronous 는 순서대로 처리하는 방식, 비동기Asynchronous 는 순서 없이 동시에 처리하는 방식
- 자바스크립트 비동기 처리는 실행 시간이 다른 함수들을 원하는 순서에 맞게 처리하는 프로그래밍
- 자바스크립트는 단일 스레드 방식인 동기 처리 방식으로 실행하고 비동기 방식으로 처리하기 위해 콜백 함수 사용

비동기 처리 방식	설명
콜백 함수	• 함수안에 또다른 함수를 매개변수로 넘겨서 실행 순서를 제어 • 전통적인 자바스크립트 비동기 처리 방식
프로미스promise	• Promise 객체와 콜백 함수를 사용해서 실행 순서 제어 • ECMAScript 2015 도입
async, await	• async 함수와 await 예약어를 사용해서 실행 순서 제어 • ECMAScript 2017 도입

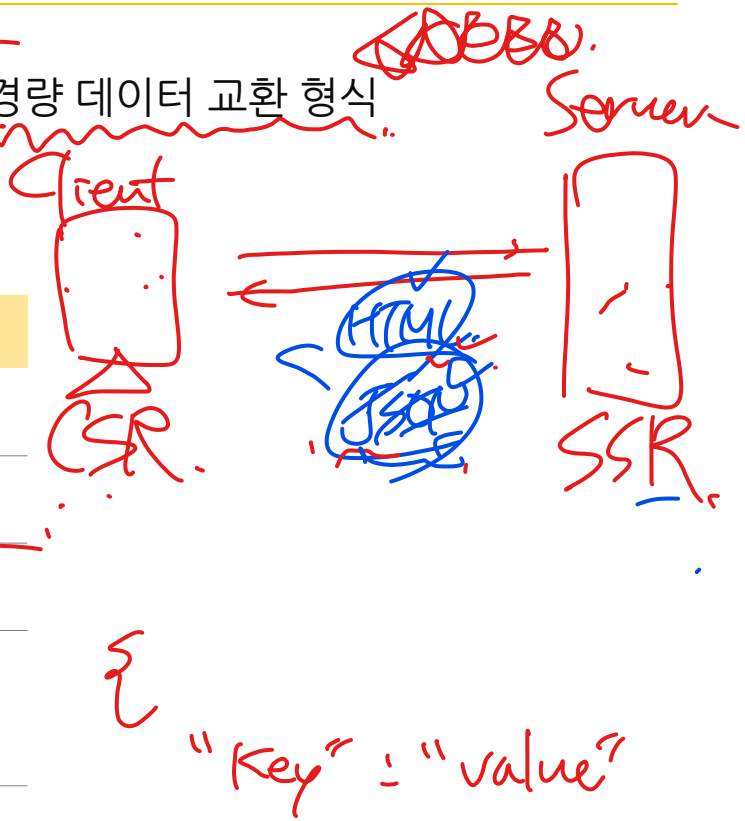
2. Promise

- ECMAScript 2015에서 비동기 방식으로 처리하기 위해 Promise 도입
- Promise는 resolve와 reject 함수에 따라 상태 전이를 통해 비동기 처리를 수행
- 프로미스 체이닝의 복잡해지는 코드를 줄이기 위해 ECMAScript 2017에서 async와 await 문법 추가

주요 함수 및 속성	설명
resolve()	비동기 처리 작업 성공 함수, fulfilled 상태
reject()	비동기 처리 작업 실패 함수, rejected 상태
then()	성공 처리 콜백 연결 함수
catch()	실패 처리 콜백 연결 함수
finally()	최종 결과 콜백 연결 함수
async	비동기 처리 함수 선언 키워드, 프로미스 반환
await	비동기 작업 완료까지 대기

3. JSON

- JSON JavaScript Object Notation 은 데이터를 구조화된 형태로 저장하고 전달하기 위해 사용되는 경량 데이터 교환 형식
- XML 비해 쉽고 특정 프로그래밍에 국한되지 않는 범용적인 데이터 교환 형식으로 사용



주요 특징	설명
텍스트 기반 형식	JSON은 텍스트 형식으로 데이터를 표현하므로, 사람이 쉽게 읽고 이해
언어 독립적	JSON은 대부분 프로그래밍 언어에서 JSON을 생성하고 파싱할 수 있는 라이브러리를 제공
키-값 쌍의 데이터 구조	JSON은 데이터를 키-값 쌍(key-value pairs)으로 표현
JSON 함수	stringify() - JSON 객체를 JSON 문자열로 변환 parse() - JSON 문자열을 JSON 객체로 변환
JSON 예시	{ "name": "John Doe", "age": 30, "courses": ["Math", "Science", "History"], "address": "Busan, Korea" }

{ "Key" : "value" }

4. AJAX

- AJAX Asynchronous Javascript And XML 는 비동기 방식으로 JSON 또는 XML 데이터를 교환하기 위한 통신 기술
- Javascript는 AJAX 통신을 위해 XMLHttpRequest 객체 사용
- XMLHttpRequest 객체를 개선한 fetch 함수는 프로미스를 반환하는 비동기 처리 함수