**React Axios 들어가기에 앞서**

**동기화(Synchronous) vs 비동기화(Asynchronous)**

**: 순서대로 실행하는 것과 그렇지 않은 것들**

1. **동기화(Synchronous)**

동기는 동시에 일어난다는 뜻으로 요청(request)을 하면 요청에 대한 결과 즉,응답( response)이 올 때 까지 기다린다는 뜻이다.

:다른 작업이 실행되는 되는 동안 다른 작업을 할수없는것을 **블로킹방식**이라한다.

taskB

taskA

taskC

만약, taskA(0.2초) , taskB(10초) , taskC(0.4초) 걸리는 작업이라면 taskB작업시간이 너무 길어 성능이 저하된다.

이럴때, 비동기화가 필요하다.

1. **비동기화(Asynchronous)**

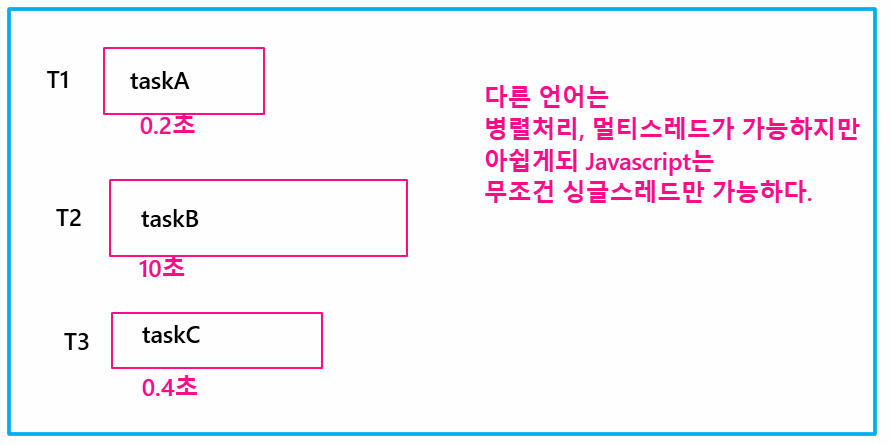
비동기는 “동시에 일어나지 않는다”는 뜻으로 요청(request)를 하고 응답(response)이 올 때까지 기다리지 않고 다른 작업을 할 수 있도록 하는 것이다.

여러개의 작업을 동시에 실행시킨다. 즉 먼저 작성한 코드의 결과를 기다리지 않고 다음 코드를 바로 실행한다.–**논블로킹방식**

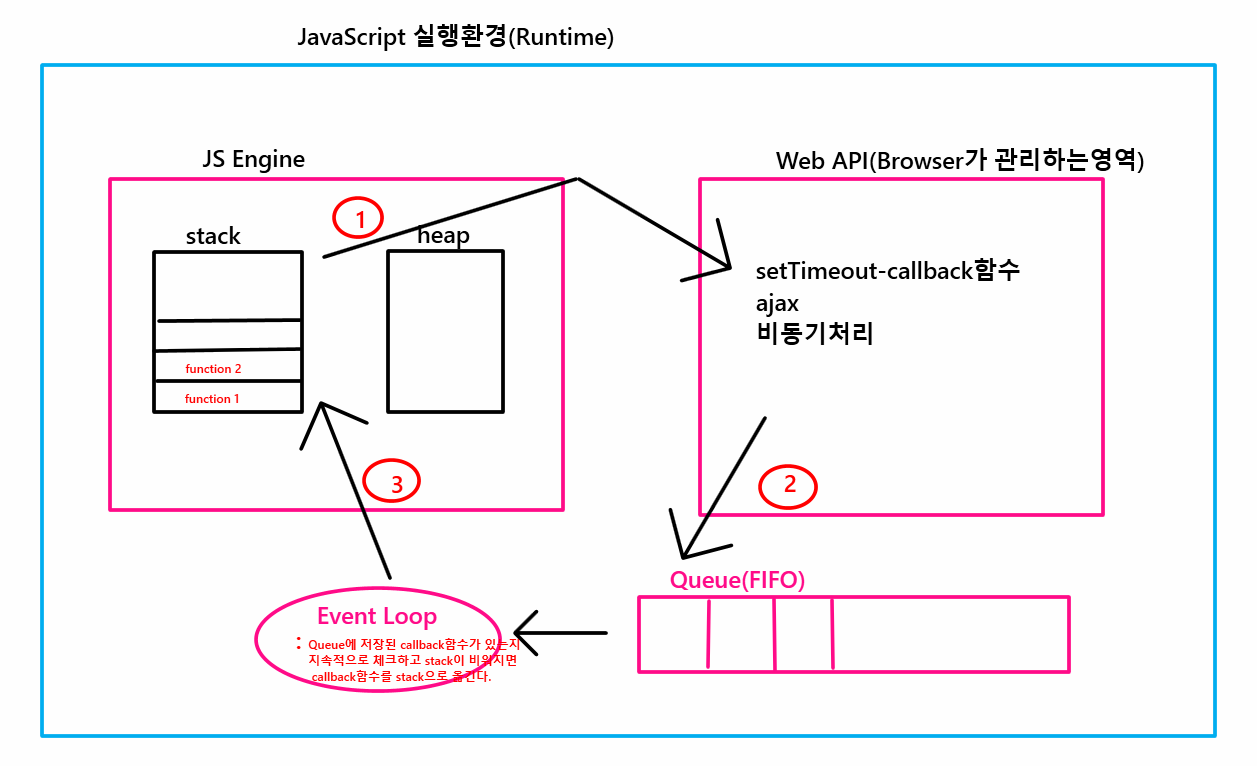
**동기방식**은 설계가 매우 간단하고 직관적이지만 결과가 주어 질 때까지 아무것도 못하고 대기해야하는단점이 있고,  
**비동기방식**은 동기보다 복잡하지만 결과가 주어지는데 시간이 걸리더라도 그 시간 동안 다른 작업을 할 수 있으므로 자원을 효율적으로 사용할 수 있는 장점이 있다

**“자바스크립트는 싱글스레드로 동작한다”**

**: 멀티스레드가 불가하기 때문에 비동기작업이필요.**



**그렇다면, JavaScript가비동기처리를어떻게?**



**비동기화(Asynchronous)처리 방법**

1. **순수 JavaScript 기반**

:소스가 길어지고 특정 작업이 끝난 후에 다른 작업(콜백함수)을 실행하고 또 그 작업이 끝 난 후에 또 다른 작업(콜백함수)을 실행 해야 하는 경우에 콜백지옥의 문제점이 발생한다.

가독성이 떨어지고 깔끔하지 못한 코드의 문제점.

1. **Promise**

**:** 비동기 처리를 효율적으로 작업 할 수 있도록 도와주는 자바스크립트 내장객체

:순수 자바스크립기반의 비동기화 처리의 문제점 Callback hell에서 탈출 하기 위한 JS 비동기를 돕는 함수이다.

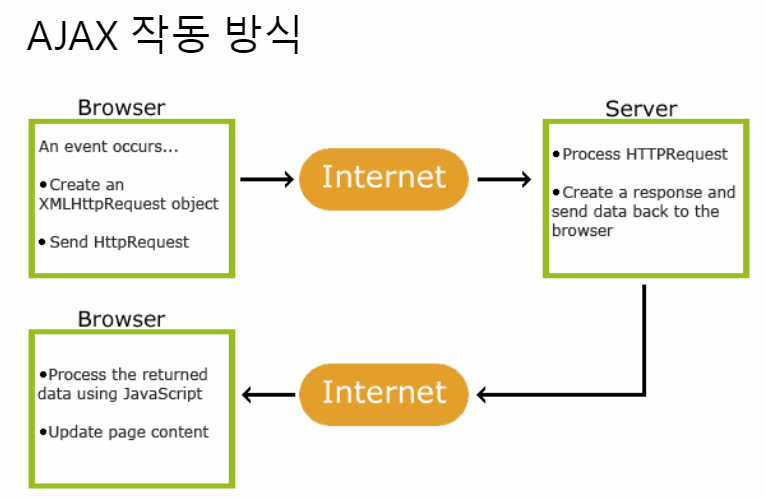
1. **Async& await**

**:**: Promise를 좀더 세련되고 직관적으로 처리 할수 있도록 돕는다.

**Ajax(A**synchronous **J**avaScript **A**nd **X**ML**)란?**

Ajax는 프로그래밍 언어가 아니라 기술적 용어로, 웹 애플리케이션과 서버 간에 데이터를 주고받는 데 사용된다. Ajax는 클라이언트와 서버 사이에서 비동기적으로 데이터를 교환하여 웹 페이지를 새로고침하지 않고도 일부 내용을 업데이트할 수 있게 해준다.

즉, 페이지가 로드된 후 페이지 전체를 새로고침하지 않고도 일부만 갱신할 수 있는 기술로, 브라우저에 내장된 XMLHttpRequest 객체를 이용해 웹 서버에 데이터를 요청하고, 그 결과를 XML 또는 JSON 형태로 받아 JavaScript 기반으로 화면에 반영한다.



요즘, 웹은 한 페이지에서 다양한 요청을 보내고 그 결과를 받아야 하는 경우가 많기 때문에 특정 요청이 올 때까지 다른 요청을 할 수 없으면 사용자 입장에서는 로딩이 느리고 화면이 제대로 랜더링되지 않는다.

그러므로, Ajax 기술을 이용하면 한 페이지에서 전체 새로고침 없이 화면의 일부분만 데이터를 갱신할 수 있기 때문에 훨씬 효율적이다.

**비동기통신**

애플리케이션의 사용자 경험을 향상시키며 서버로부터 응답이 오는 동안 사용자가 다른 작업을 계속할 수 있다.

**그렇다면,**

**JavaScript에서 비동기 통신을 어떻게?**

1. **Fetch 함수 – IE에서 지원이잘 안됨.**

: 원격 API를 간편하게 호출 할 수 있도록 브라우져에서 제공하는 fetch()함수 - window.fetch(url, option);

1. **외부 lib 사용(Promise기반의 Http 비동기 통신라이브러리)**

**: 원격 API를 호출 하는 라이브러리**

- jQuerylib

- Axios lib

**이를 통해 개발자들은 비동기 작업의 성공 또는 실패에 따라 콜백 함수를 체인으로 연결할 수 있다.**

**여기서 잠깐,**

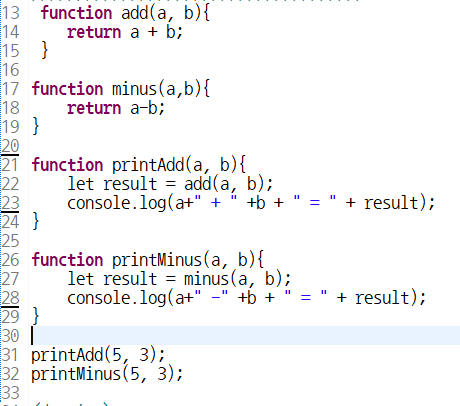
**콜백함수(Callback Function)란?**

: 함수를 파라미터(매개변수)로 전달받는 함수를 뜻함.

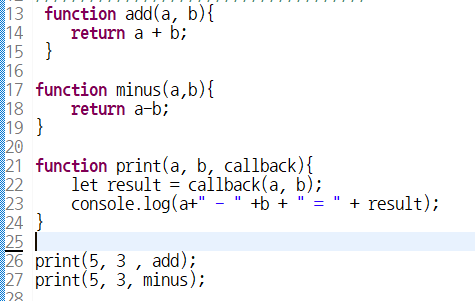
: 함수에서 특정 작업이 완료되었을 때,추가로 실행 해야 하는 작업을 기술 한 함수를 만들 때 많이 사용한다.

ex) **setTimeout(콜백함수 ,ms);  / Ajax와같은Network작업**

**-콜백함수를 이용하지 않은 코드**



**-콜백함수 이용한 코드**



**☞콜백함수의 장점**

1. 함수를 인자로 받기 때문에 필요에 따라 함수의 정의를 달리해 전달할 수 있다.
2. 함수를 굳이 정의하지 않고 익명 함수로도 전달 할 수 있다.
3. 비동기(Asynchronous) 처리 방식을 좀더 간결하게 할 수 있다.

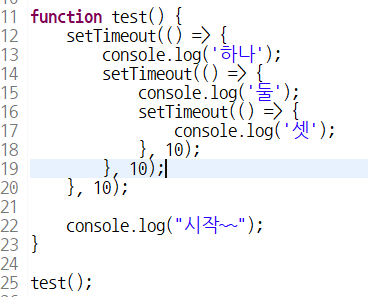
**☞콜백함수의 단점**

1. 콜백함수를 너무 남용하면 코드의 가독성이 떨어진다**. (콜백 지옥)**
2. 에러 처리가 어렵다.

**콜백지옥란?**

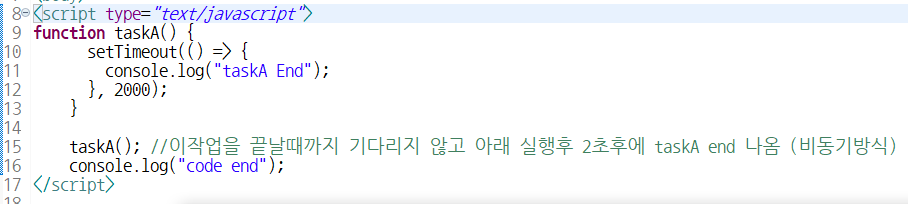
콜백지옥(callback hell)이란 콜백함수를 익명함수로 전달하는 과정에서 또 다시 콜백안에 함수 호출이 반복되어 코드의 들여쓰기 수준이 감당하기 힘들 정도로 깊어지는 현상을 말한다.

**콜백지옥 코드 예**

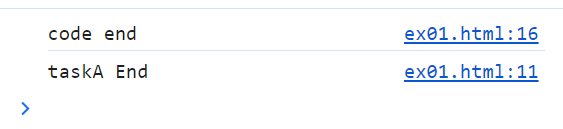


**Ex) 비동기 코드를 만들어보자.**

ex01.html

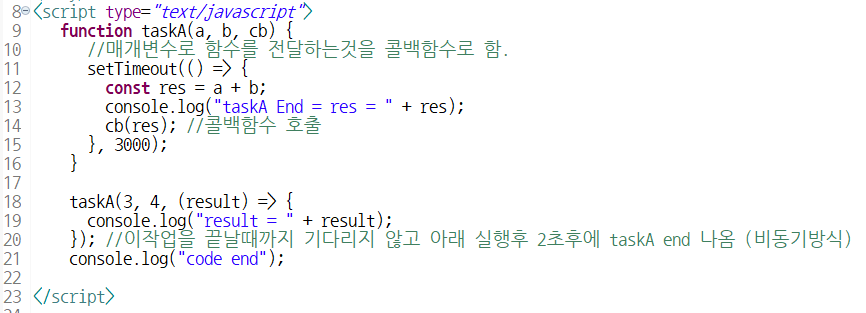


실행결과

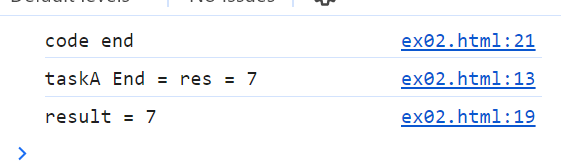


**Ex) 비동기처리의 결과 값을 확인하기 위한 콜백함수 전달해보자.**

**ex02.html**

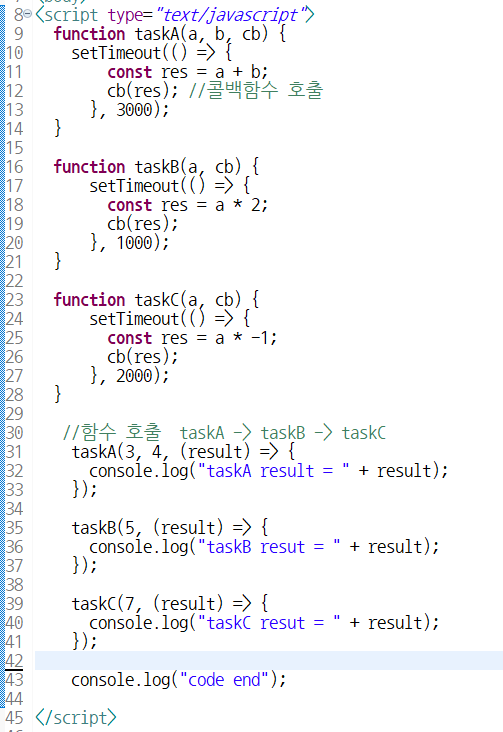


-실행결과

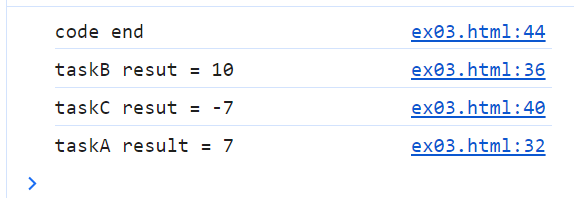


**Ex) 작업 여러 개를 만들어서 A->B->C 순으로 실행 해보자.**

ex03.html



* 실행결과

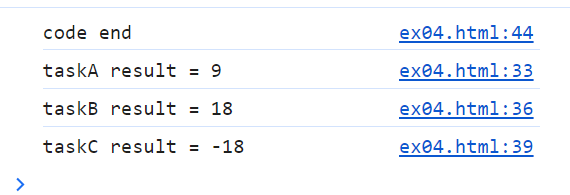


**Ex) 위 코드는 동기적 작업이 안된다.콜백 함수의 결과를 받아서 다른 작업의 인자로 전달하여 호출 해보자.**

Ex04.html



-실행결과



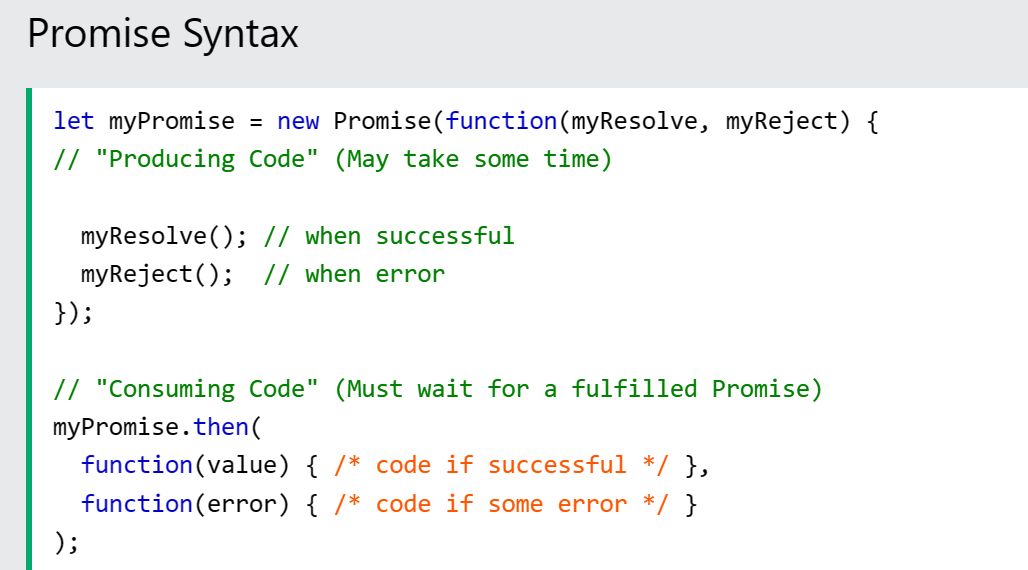
위 코드가 복잡하다 – 콜백지옥(callback hell)

: JS에서 콜백지옥을 벗어나기 위해서 promise를 지원한다.

Ex)위 코드를 Promise로 변경해보자.

ex04.html

**<Promise문법>**



**Axios란?**

* **https://axios-http.com/docs/api\_intro**
* Axios는 node.js와 브라우저를 **위한 Promise 기반 HTTP 비동기 통신 라이브러리이다**.
* 백엔드 서버와 프론트엔드 서버간의 통신을 위해 사용한다.
* 서버 사이드에서는 네이티브 node.js의 http 모듈을 사용하고, 클라이언트(브라우저)에서는 XMLHttpRequests를 사용한다.
* **HTTP 요청과 응답을 JSON 형태로 반환해준다.**
* 주요 HTTP 요청 유형(GET, POST, PUT, DELETE 등)을 지원이를 통해 웹 애플리케이션은 서버와 다양한 방식으로 통신할 수 있다.
* **Axios는 더 간결하고 명확한 오류 처리 방식을 제공한다.**
* Promise 기반 구조를 활용하여 .catch 메서드를 사용할 수 있다.
* 일부 오래된 브라우져에서는 작동하지 않을 수 있다.

**Axios(config)**

**axios({**

**method:,**

**url :**

**data:**

**})**

**.then((result)=>{**

**//성공**

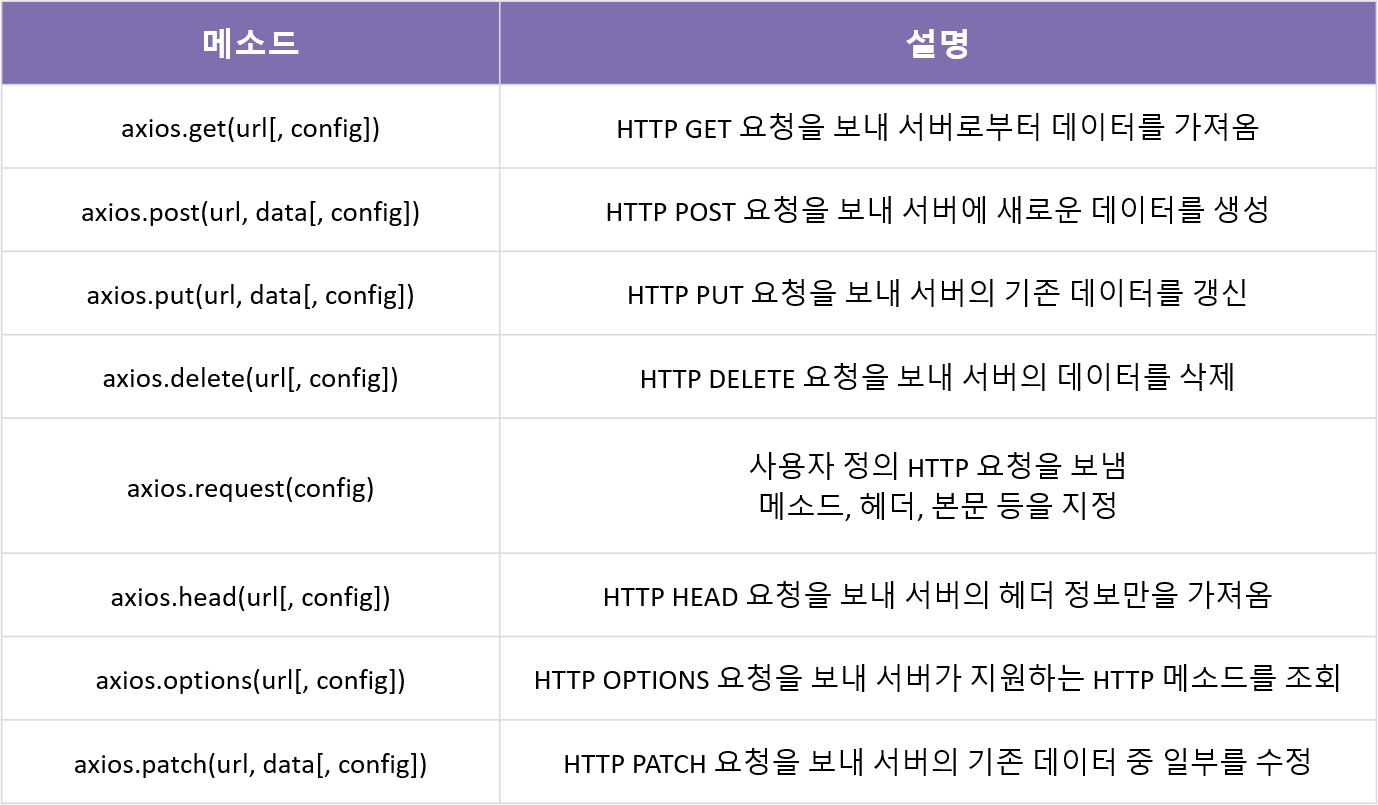
**})**

**.catch((error)=>{**

**//실패**

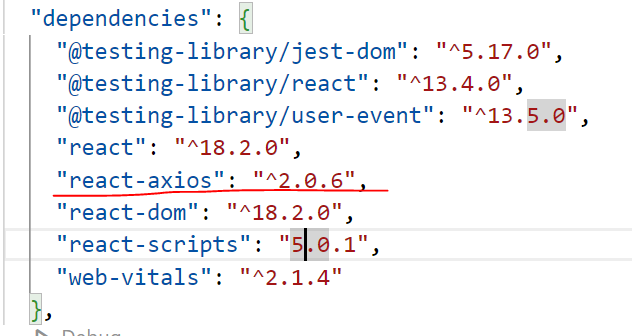
**});**

**}**

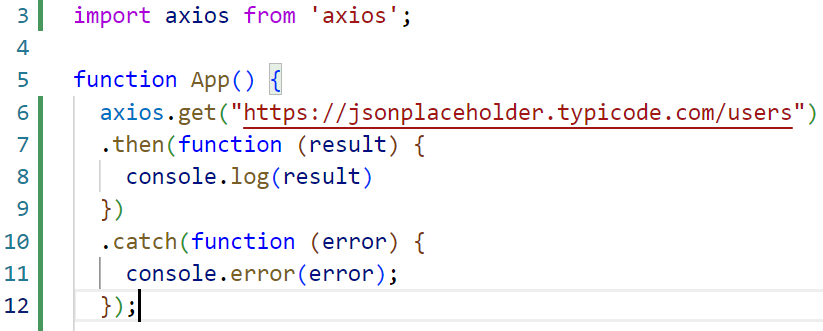


**이제 ReactAxios라이브러를 사용하여 비동기통신 해보자.**

1. VS\_CODE tool을 열고 Terminal을 open한다.
2. npm create vite@latest my-axios 프로젝트 생성
3. cd my-axios 폴더 이동
4. npm i
5. **npm install react-axios설치**
6. package.json 확인



**코드예시)**



**실행 결과)**

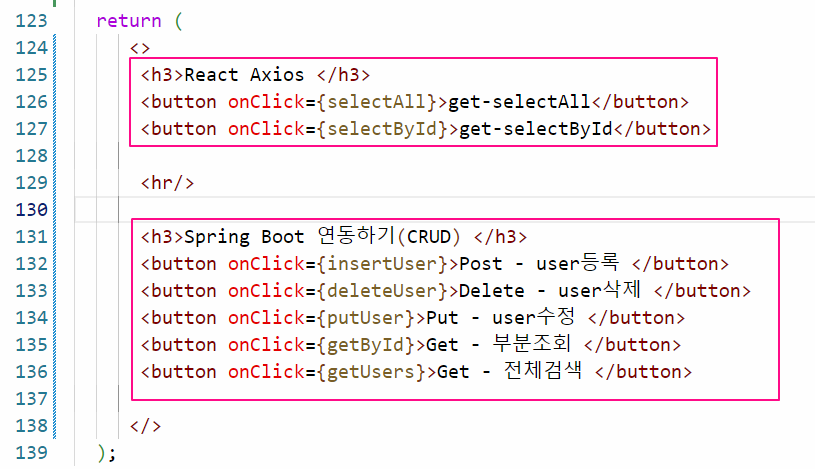


**☞ 샘플 데이터(가상데이터)를 제공하는 무료 온라인 REST API서비스**

<https://jsonplaceholder.typicode.com/>

**실습해보기**

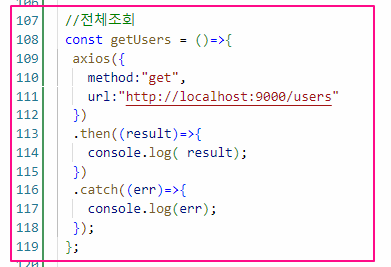
**App.jsx파일**

****

****

****

****

****