



# Chap42 비동기 프로그래밍

## ▼ 동기 처리와 비동기 처리

자바스크립트 엔진은 단 하나의 실행 컨텍스트 스택을 갖는다.(즉, 싱글 스레드이다.) 이는 함수를 실행할 수 있는 창구가 단 하나이기 때문에 실행 중인 실행 컨텍스트를 제외한 나머지는 모두 대기중인 태스크들이다.

### 따라서 동기 처리를 할 경우에는

**장점** - 태스크들이 순서대로 하나씩 처리되므로 실행 순서가 보장된다.

**단점** - 앞선 태스크가 종료할 때까지 이후 태스크들이 블로킹 된다.

### 그렇다면 비동기 처리를 할 경우에는

**장점** - 종료되지 않는 태스크가 있더라도 곧바로 실행하므로 블로킹이 발생하지 않는다.

**단점** - 태스크의 실행 순서가 보장되지 않는다. 그리고 비동기 함수는 전통적으로 콜백 패턴을 사용하는데 콜백 패턴은 콜백 헬을 발생시켜 가독성을 나쁘게 하고, 에러의 예외 처리가 곤란하며여러개의 비동기 처리를 한 번에 처리하는 데도 한계가 있다.

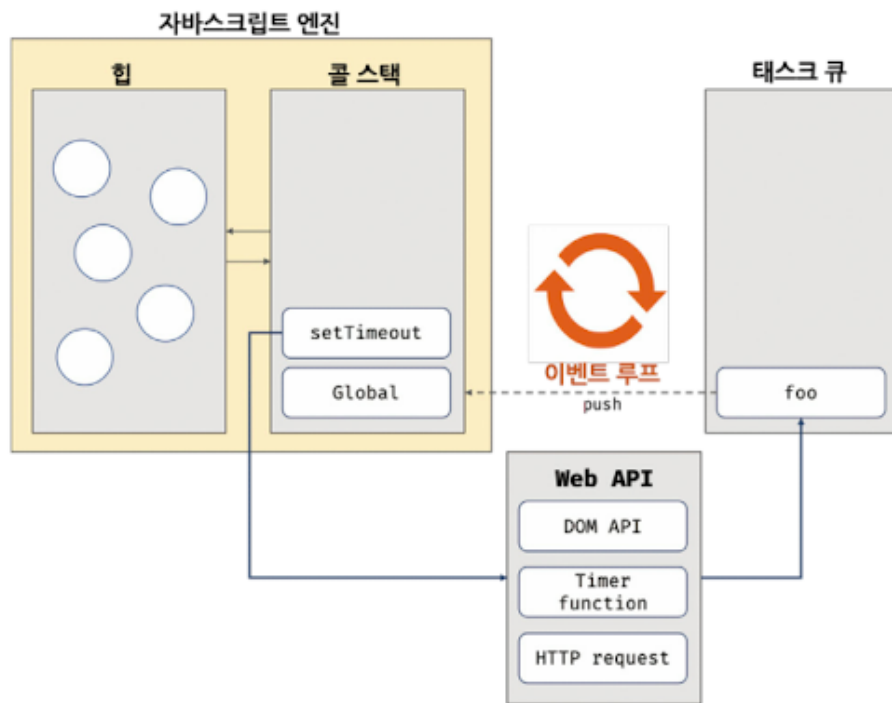
### ▼ 동기 처리

현재 실행 중인 태스크가 종료할 때까지 다음에 실행될 태스크가 대기하는 방식

### ▼ 비동기 처리

현재 실행 중인 태스크가 종료되지 않은 상태라 해도 다음 태스크를 곧바로 실행하는 방식

## ▼ 이벤트 루프와 태스크 큐



비동기 처리에서 소스코드의 평가와 실행을 제외한 모든 처리는 자바스크립트 엔진을 구동하는 환경인 브라우저 또는 Node.js가 담당한다.

따라서 자바스크립트 엔진이 싱글 스레드임에도 불구하고 비동기 동작이 가능한 이유는 브라우저 환경의 이벤트 루프와 태스크 큐 때문이다.

#### ▼ 이벤트 루프

콜스택에 현재 실행 중인 실행 컨텍스트가 있는지, 그리고 태스크 큐에 대기 중인 함수(콜백 함수, 이벤트 핸들러 등)가 있는지 반복해서 확인한다. 만약 콜 스택이 비어있고 태스크 큐에 대기 중인 함수가 있다면 이벤트 루프는 순차적으로 태스크 큐에 대기 중인 함수를 콜 스택으로 이동시킨다. 이 때 콜 스택으로 이동한 함수는 실행된다. 즉, 태스크 큐에 일시 보관된 함수들은 비동기 처리 방식으로 동작한다.

#### ▼ 태스크 큐

`setTimeout`이나 `setInterval`과 같은 비동기 함수의 콜백 함수 또는 이벤트 핸들러가 일시적으로 보관되는 영역이다. 태스크 큐와는 별도로 프로미스의 후속 처리 메서드의 콜백 함수가 일시적으로 보관되는 마이크로태스크 큐도 존재한다.

즉 자바스크립트 엔진은 싱글 스레드로 동작하지만 브라우저는 멀티 스레드로 동작하기 때문에 비동기 동작이 가능한 것이다.

그리고 브라우저는 자바스크립트엔진 외에도 렌더링 엔진과 Web API를 제공한다.