


이벤트 루프

JavaScript 비동기 핵심 Event Loop 정리

자바스크립트는 단일 스레드 기반으로 비동기로 동작하나요?

 <https://medium.com/sjk5766/javascript-비동기-핵심-event-loop-정리-422eb29231a8>

JS Concept



JavaScript는 싱글 스레드 기반 언어로 한 번에 하나의 작업만 가능하다.

하지만 비동기로 동작하기 때문에 단일 스레드임에도 불구하고 많은 작업을 수행한다.

하지만 JavaScript 언어 자체가 비동기 동작을 지원하는 것은 아니다. 비동기로 동작하는 핵심 요소는 브라우저가 가지고 있다.

브라우저는 Web APIs, Event Table, Callback Queue, Event Loop 등으로 구성되며 JavaScript 코드가 실행될 때 브라우저와의 동작은 JS → Web API → Calback Queue → Event Loop 순으로 동작한다. 그림으로 표현하면 아래와 같다.

- Heap : 메모리 할당이 발생하는 곳
- Call Stack : 실행된 코드의 환경을 저장하는 자료구조, 함수 호출 시 Call Stack 에 Push 됨
- Web APIs : DOM, AJAX, setTimeout 등 브라우저가 제공하는 API

나머진 아래에서 설명

```
setTimeout(function exec () {  
  console.log('second')  
}, 1000)
```

위 코드가 실행될 때 각 구성요소들의 역할은?

- Web APIs : setTimeout이 Call Stack 에 들어와 실행되면 Browser API인 timer를 호출한다.
- Event Table : 특정 event가 발생했을 때 어떤 callback 함수가 호출돼야 하는지를 알고 있는 자료구조로 위 코드에서 호출된 timer가 종료되면 event 가 발생하는데 이때 exec callback 함수가 실행되어야 한다는 것을 Event Table이 알고 있음
- Callback Queue : 이벤트 발생 시 실행해야 할 callback 함수가 Callback Queue에 추가됨
- Event Loop : Call Stack과 Callback Queue를 감시하고, Call Stack이 비었을 경우, Callback Queue에서 함수를 꺼내 Call Stack 에 추가한다.

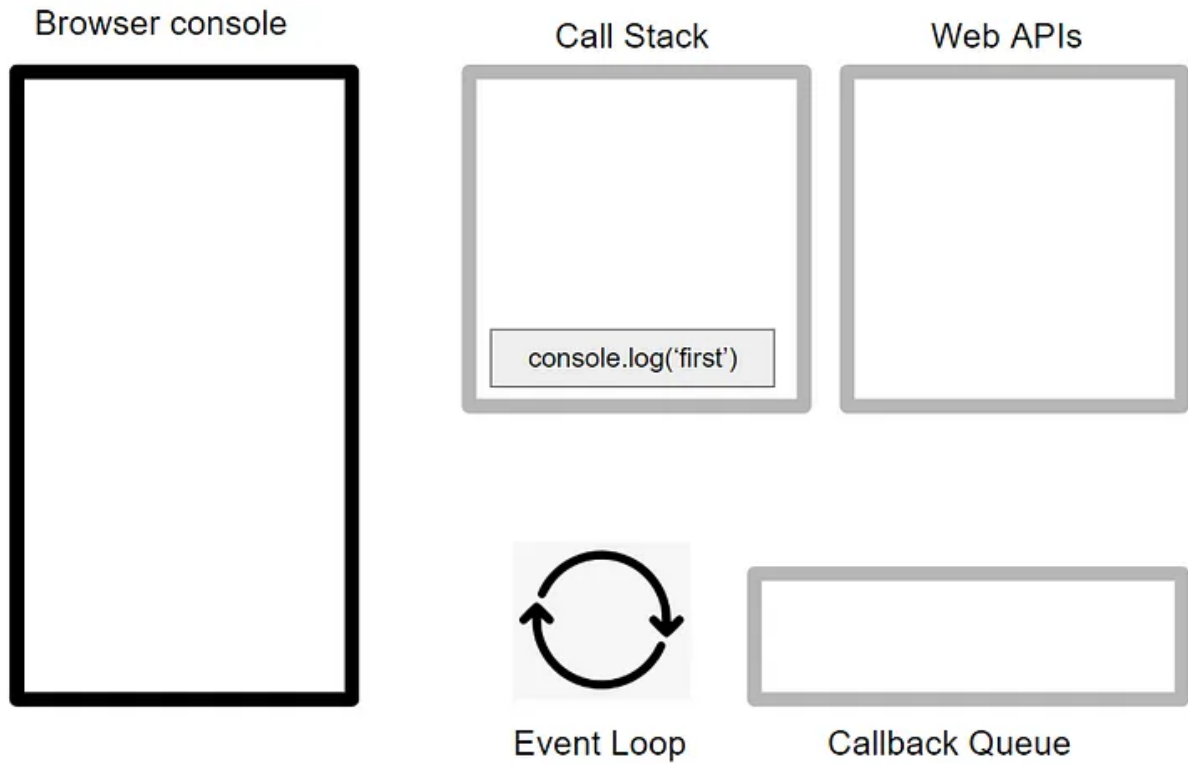
예제를 보면...

```
console.log('first')

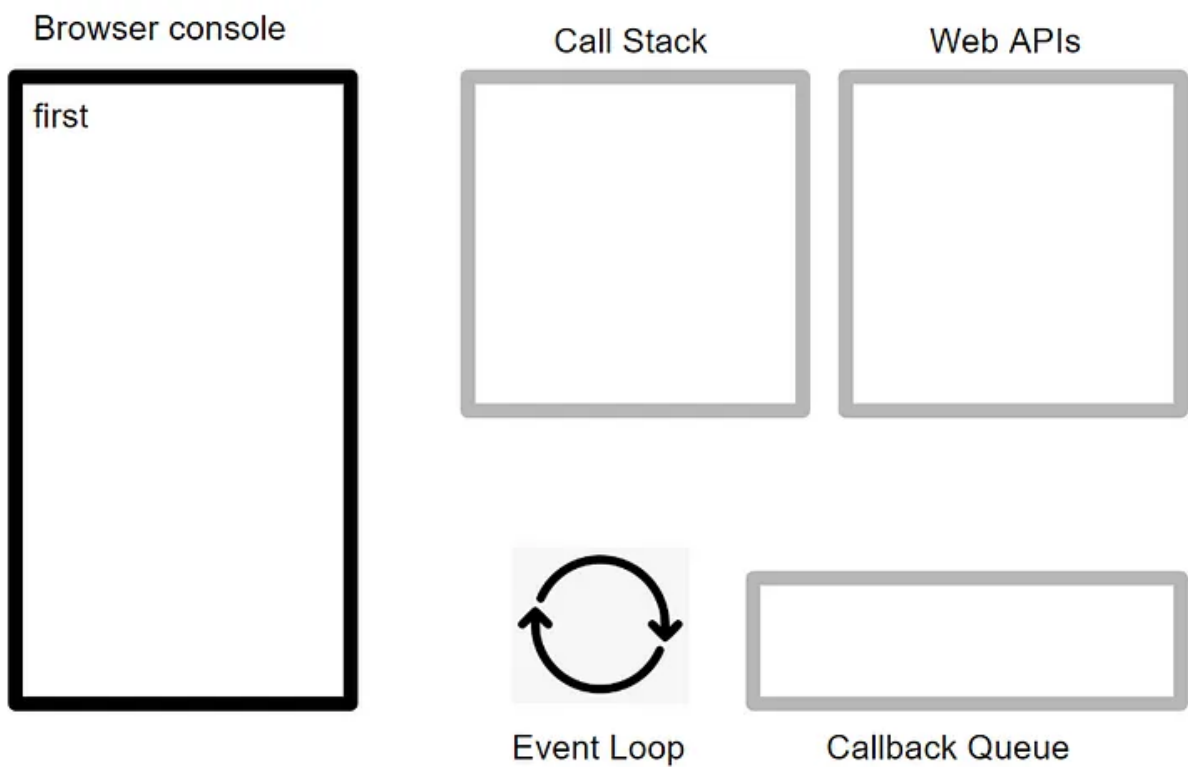
setTimeout(function cb() {
  console.log('second')
}, 1000); // 0ms 뒤 실행

console.log('third')
```

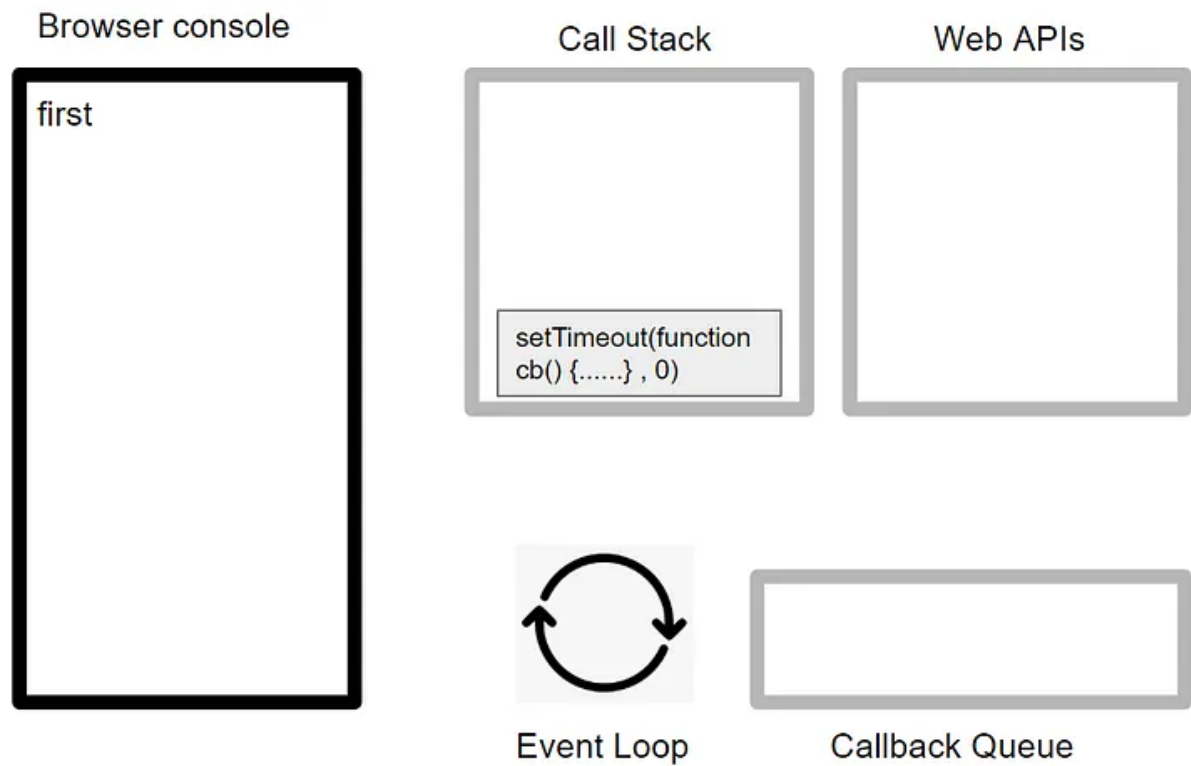
1. console.log('first') 가 call stack에 추가(push)된다.



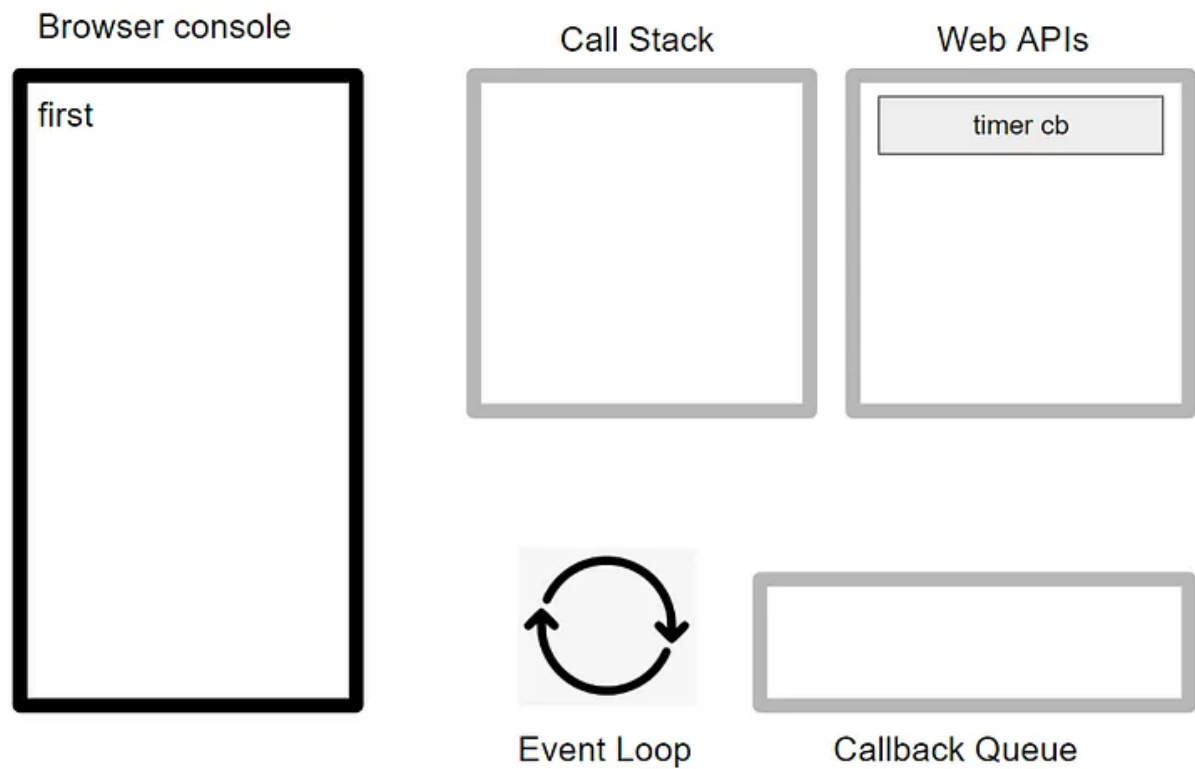
2. console.log('first') 가 실행되어 화면에 출력한 뒤, call stack 에서 제거(pop) 된다.



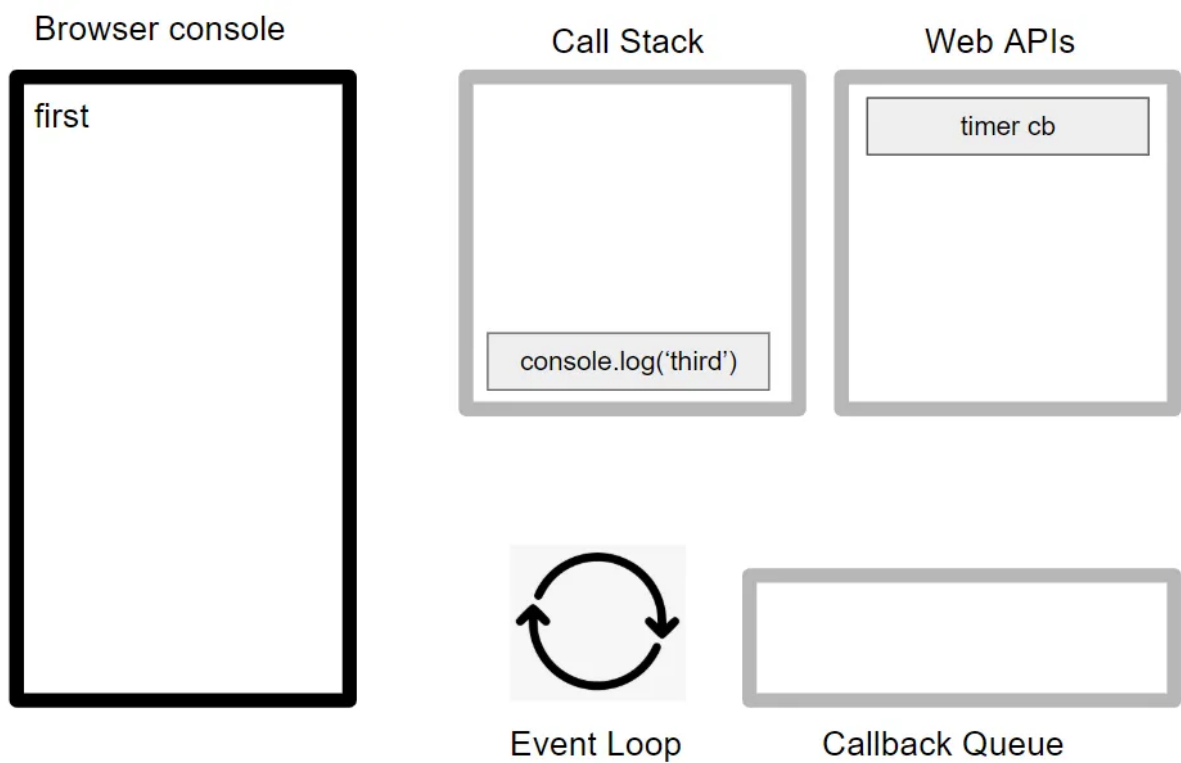
3. setTimeout 함수가 call stack 에 추가된다.



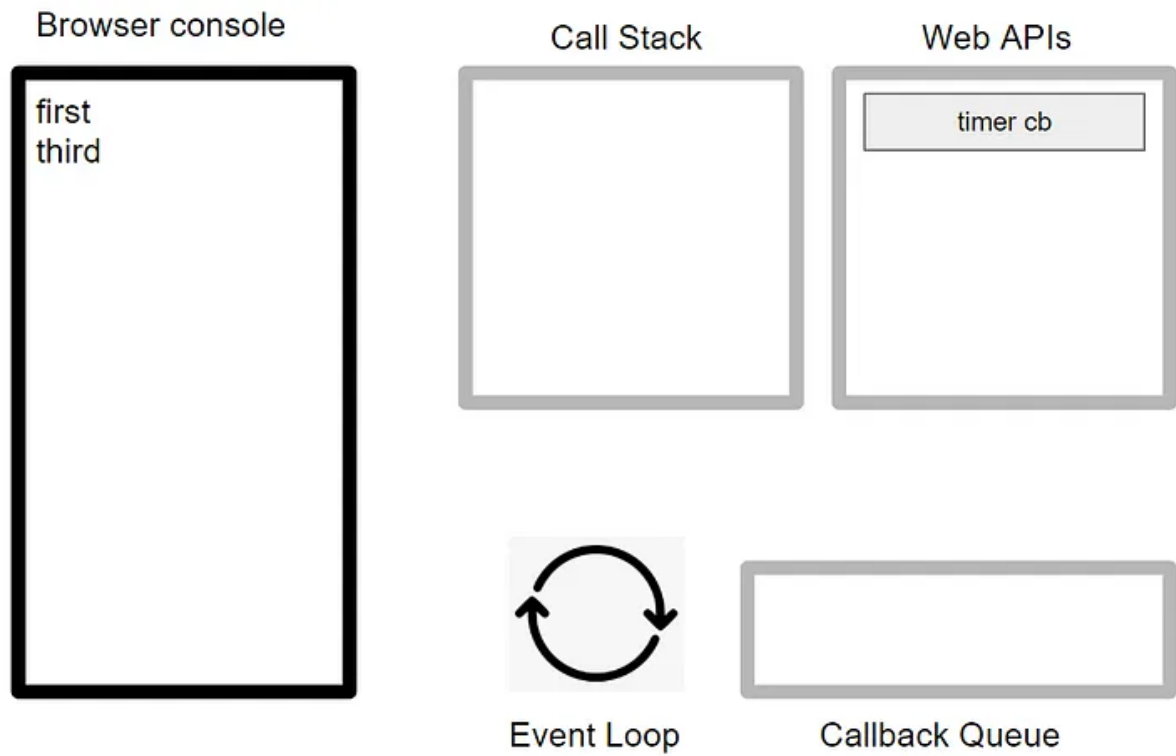
4. setTimeout 함수가 실행되면서 브라우저가 제공하는 timer web API를 호출하고 call stack에서 제거된다.



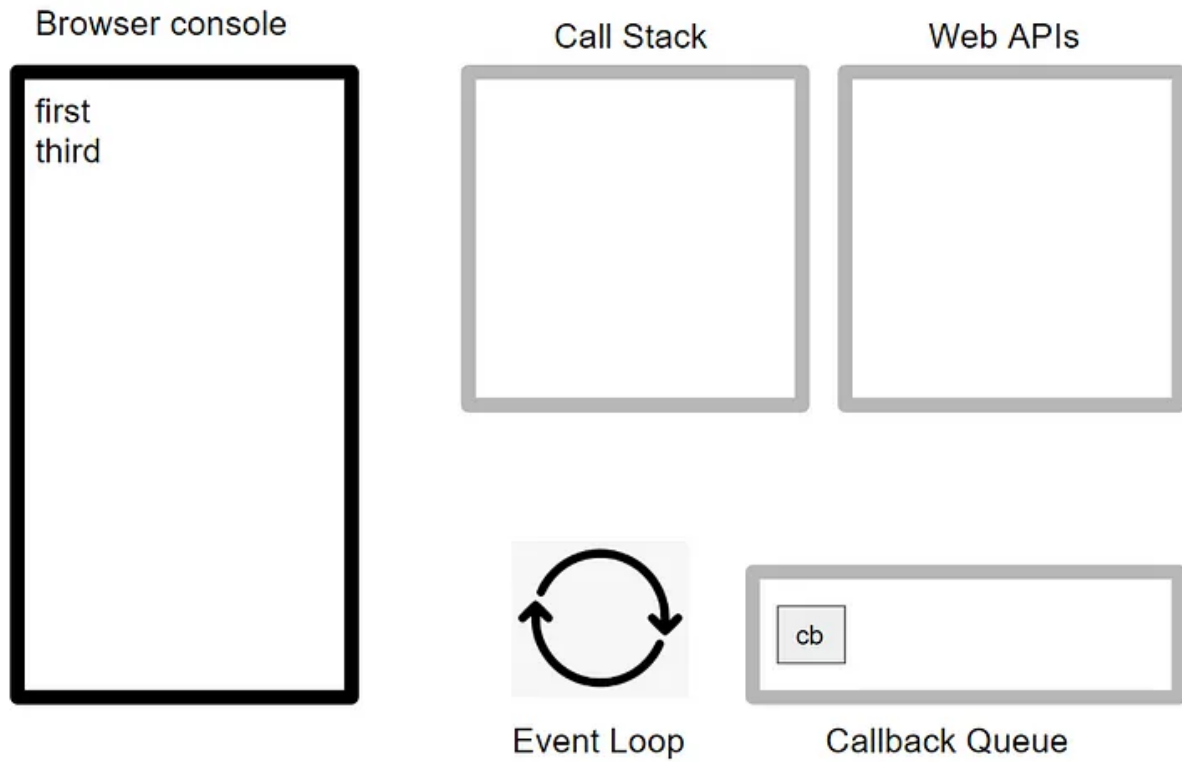
5. `console.log('third')`가 call stack에 추가된다.



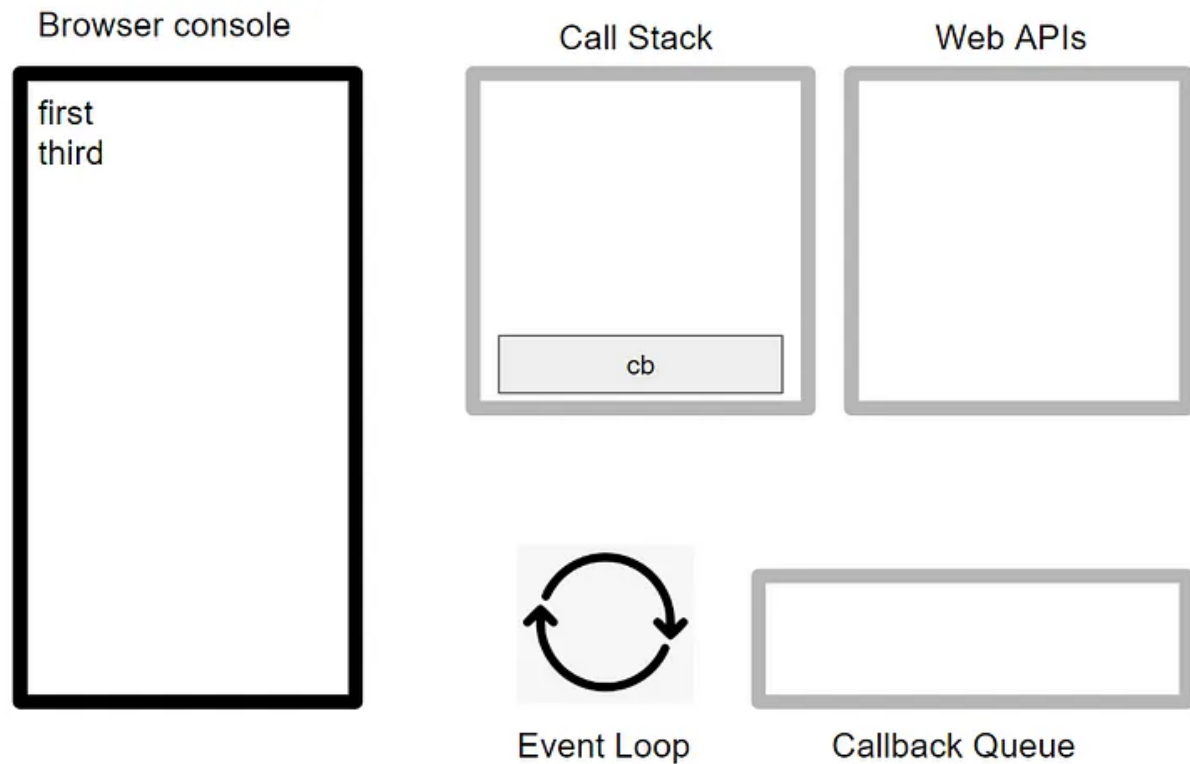
6. `console.log('third')`가 실행되어 화면에 출력되고 call stack에서 제거된다.



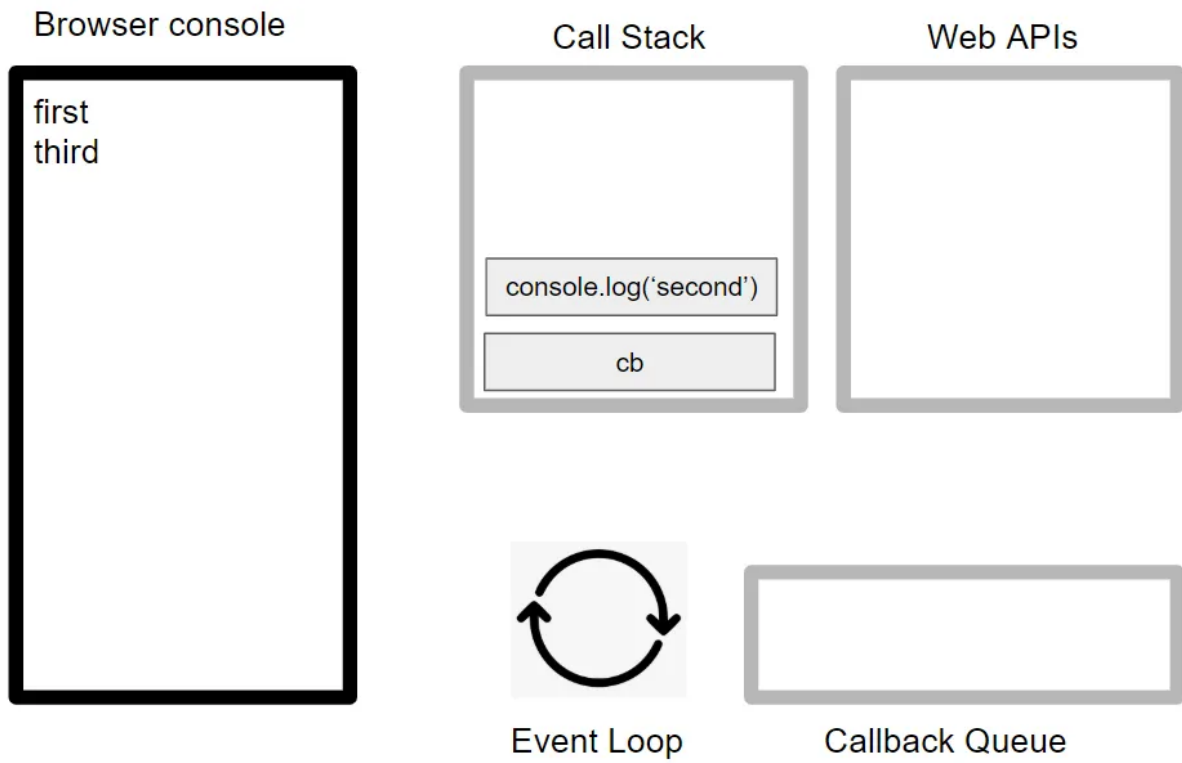
7. `setTimeout` 함수에 전달한 0ms 시간이 지난 뒤 callback으로 전달한 cb 함수가 Callback Queue에 추가된다.



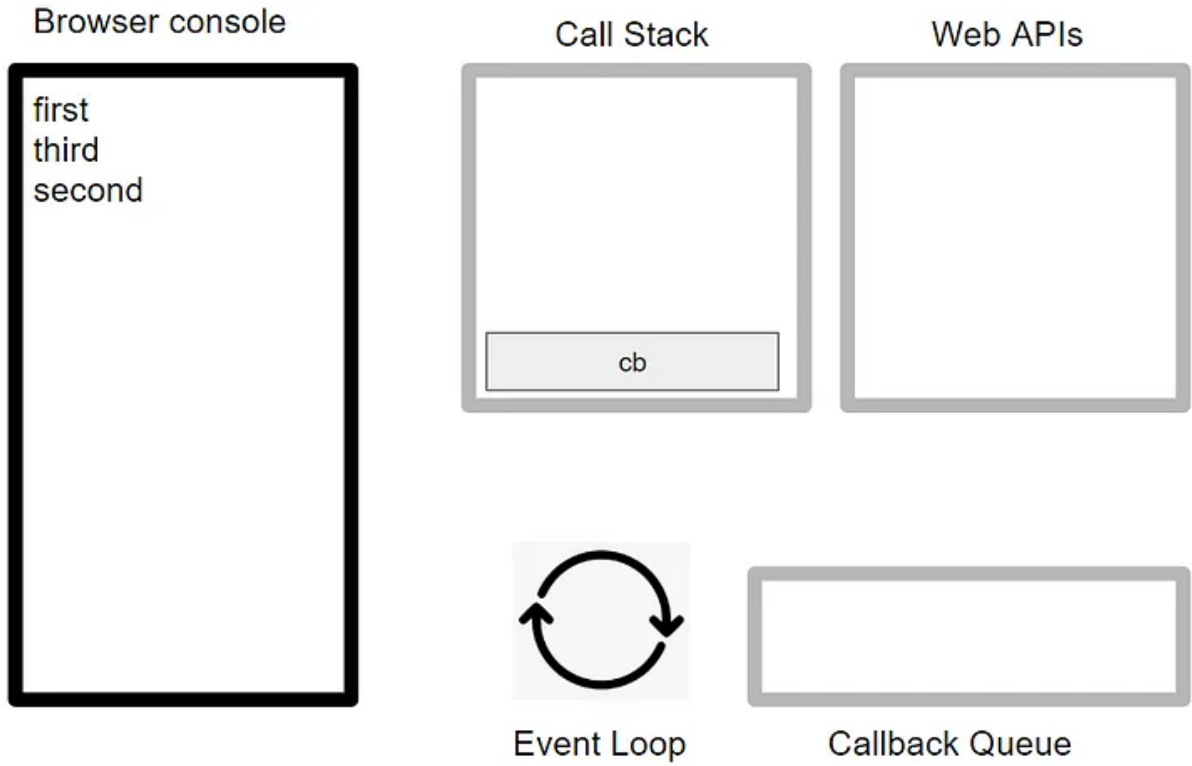
8. Event Loop는 Call stack 이 비어있는 것을 확인하고 Callback Queue를 살펴본다. cb를 발견한 event loop는 Call stack에 cb를 추가한다.



9. cb 함수가 실행되고 내부의 console.log('second')가 call stack에 추가된다.



10. console.log('second')가 화면에 출력되고 call stack에서 제거된다.



11. cb가 call stack에서 제거된다.

