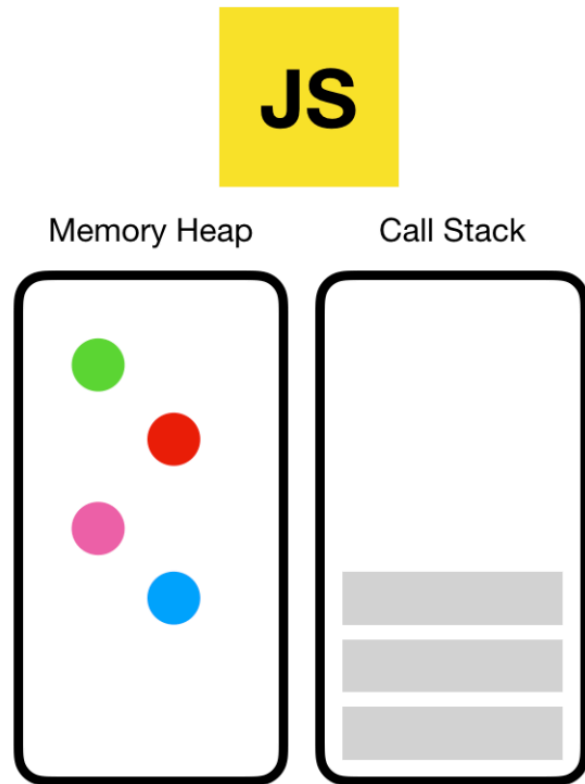


Javascript

호출 스택 & 이벤트루프

자바스크립트 엔진

가장 대중적인 자바스크립트의 엔진은 구글의 V8엔진
V8 엔진은 크롬과 노드 안에서 동작



메모리 힙(Memory Heap)

변수와 객체에 대한 모든 메모리 할당은 여기서 발생

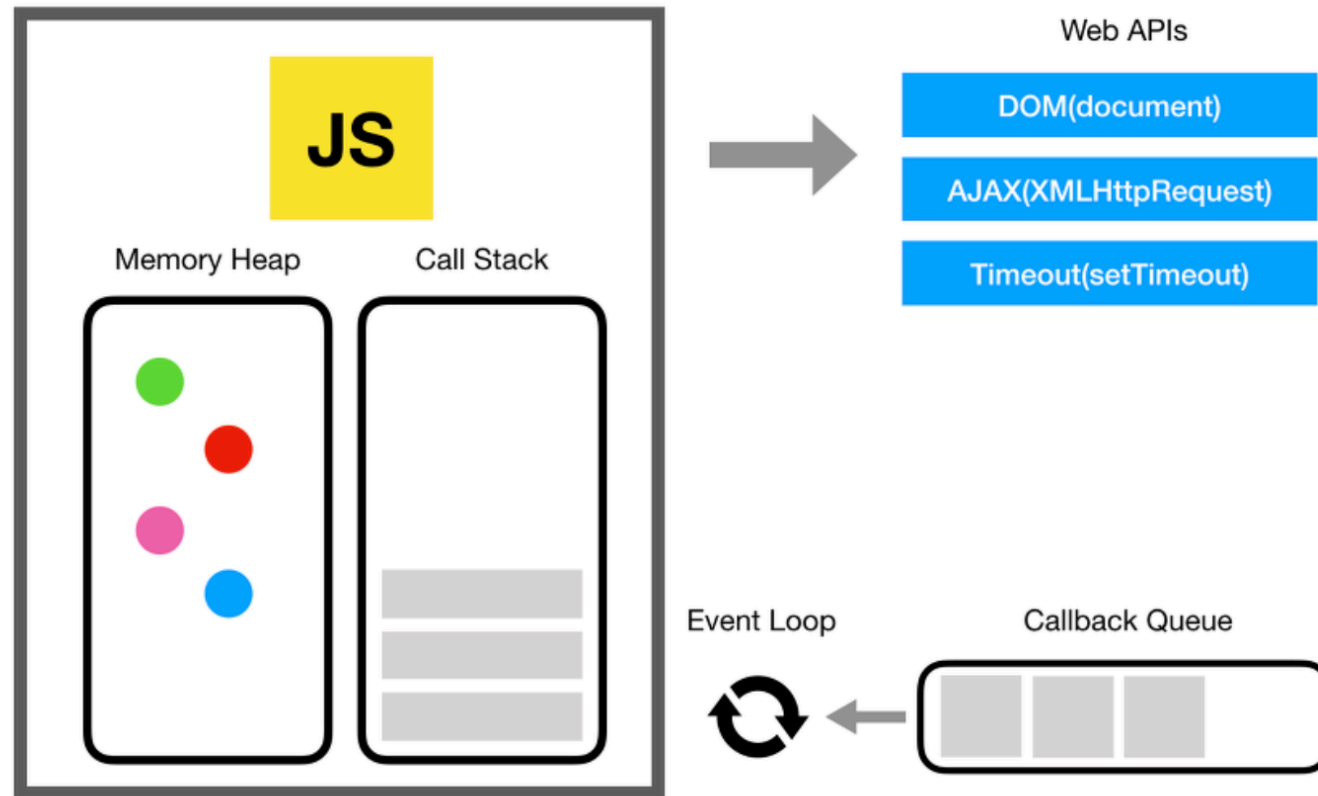
호출 스택(Call Stack)

코드가 실행될 때 호출 스택이 쌓임

실행 환경(Runtime)

브라우저는 엔진 / Web API / Event Queue와 Event Loop 등으로 구성

Web API : <https://developer.mozilla.org/ko/docs/Web/API>



호출 스택(Call Stack)

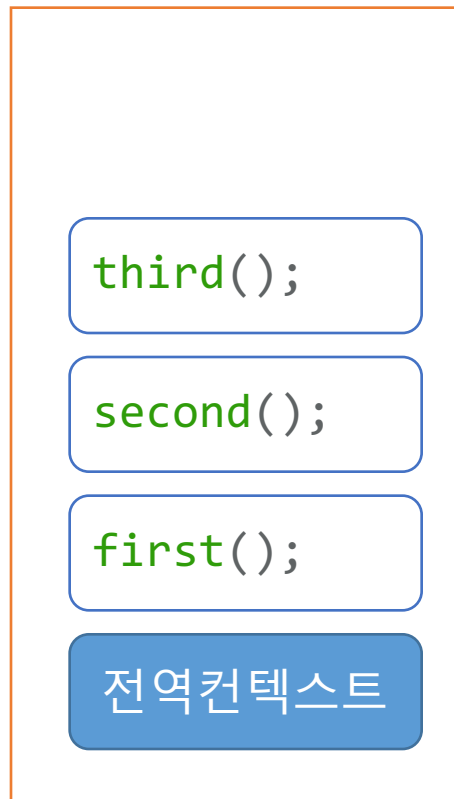
LIFO(마지막에 들어온 것이 먼저 나감) 구조

호출 스택이 가득 찼을 때 발생하는 에러 (보통 만 개 정도)
Uncaught RangeError: Maximum call stack size exceeded

```
function first() {  
  second();  
  console.log('첫 번째');  
}  
  
function second() {  
  third();  
  console.log('두 번째');  
}  
  
function third(){  
  console.log('세 번째');  
}  
  
~first();  
~third();
```

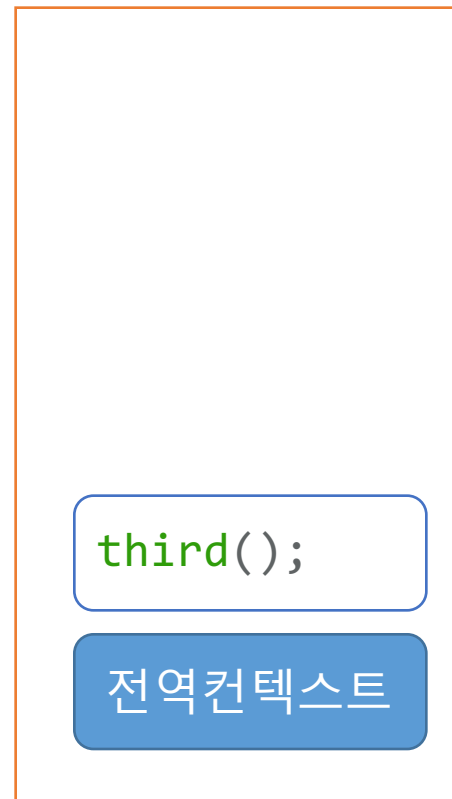


호출 스택(Call Stack)



~first() 완료

호출 스택(Call Stack)



~third() 완료

호출 스택(Call Stack)



실행완료

이벤트 루프(Event Loop)

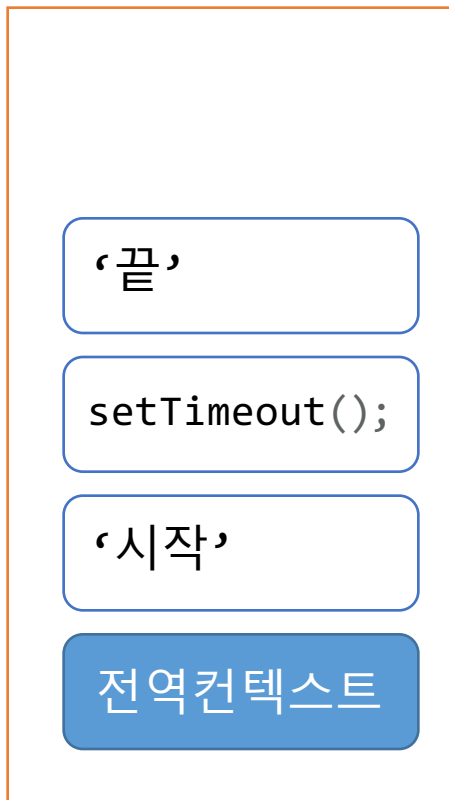
자바스크립트는 보통 싱글 스레드라고 불리는데,
바로 메인 스레드인 이벤트 루프가 싱글 스레드이기 때문이다.

```
function run () {  
  console.log('동작');  
}  
  
console.log('시작');  
  
setTimeout(run, 3000);  
  
console.log('끝');
```

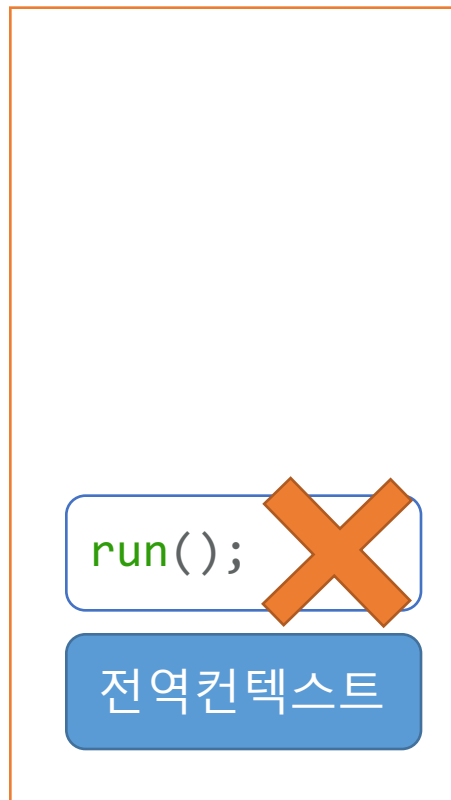


시작
끝
동작

호출 스택(Call Stack)



호출 스택(Call Stack)



3초 후

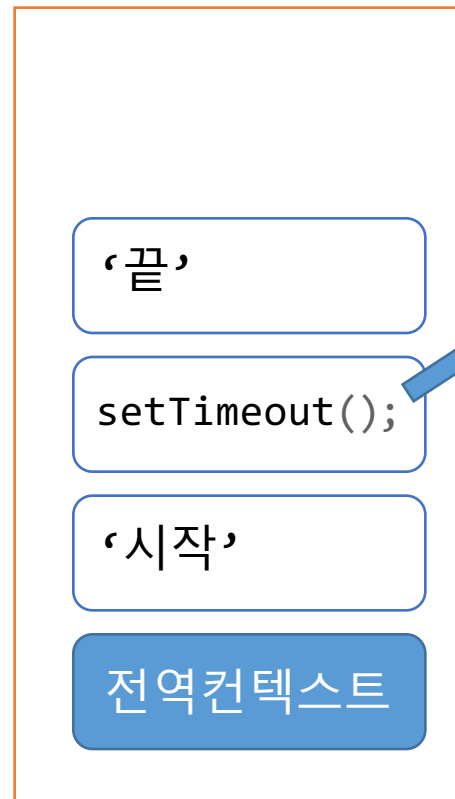
이벤트 루프(Event Loop)

호출 스택이 비워지면(전역 컨텍스트 종료되면)
태스크 큐에서 함수를 하나씩 호출 스택으로 밀어 올림.

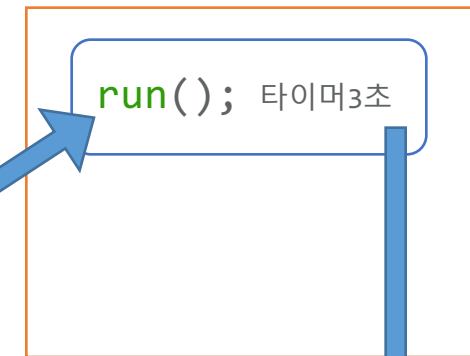
```
function run () {  
  console.log('동작');  
}  
  
console.log('시작');  
setTimeout(run, 3000);  
console.log('끝');
```



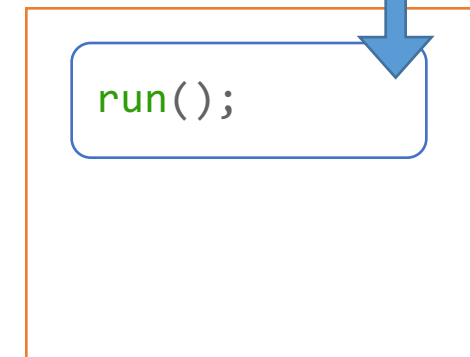
호출 스택(Call Stack)



백그라운드



태스크 큐



이벤트 루프

setTimeout 0도 기본적으로 4ms의 지연 시간이 있음
자바스크립트 기본 제공 메소드 중 콜백 함수를 사용하는 것들은
백그라운드를 사용하는 경우가 많다

참고

- 제로초블로그 : <https://www.zerocho.com/category/JavaScript/post/597f34bbb428530018e8e6e2>
- 마스터누누 블로그 : <https://new93helloworld.tistory.com/358>
- 자바스크립트 33가지 개념 : <https://github.com/leonardomso/33-js-concepts>