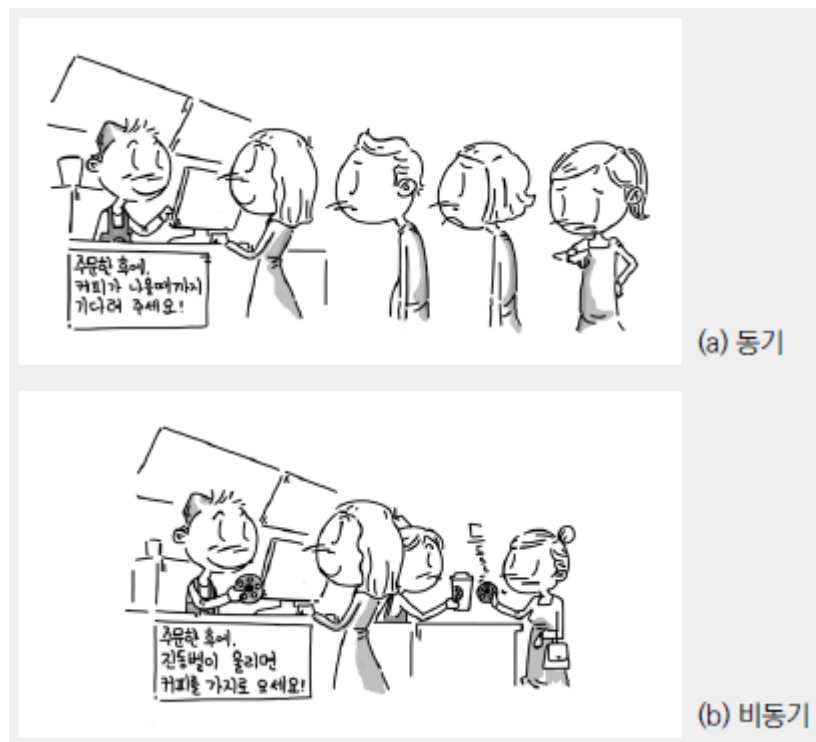


# 비동기 프로그래밍(8월 7일)

|      |                      |
|------|----------------------|
| 👤 배정 |                      |
| ▼ 상태 | 8월                   |
| 🕒 속성 | @2021년 8월 7일 오후 9:36 |
| ▼ 언어 |                      |

## 동기 & 비동기 처리

### 😊 동기



<동기와 비동기의 차이를 알수있는 예시도>

자바스크립트 엔진은 단 하나의 실행 컨텍스트 스택을 갖는다.

이것은 함수를 실행할 수 있는 창구가 단 하나이며 동시에 2개 이상의 함수를 실행 할 수가 없다는 것을 의미한다. 대기중인 태스크들은 현재 실행 중인 실행 컨텍스트가 팝되어 실행 컨텍스트 스택에서 제거되면, 다시 말해 현재 실행중인 함수가 종료하면 비로소 실행되기 시작한다는 것이다.

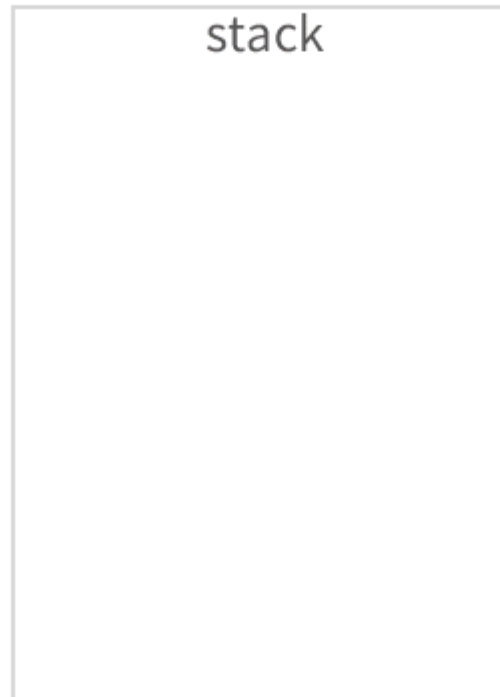
- 자바스크립트는 한번에 하나의 태스크만 실행할 수 있는 **싱글스레드** 방식이다.
- 싱글스레드는 한번에 하나의 태스크를 실행하는 **블로킹**이 발생된다.

```
function multiply (a, b) {
  return a * b;
}

function square (n) {
  return multiply(n, n);
}

function printSquare (n) {
  var squared = square(n);
  console.log(squared);
}

printSquare(4);
```



이런 구조로 함수가 호출 되기 때문에, 만약 잘못해서 무한호출 되는 재귀함수를 실행시킨다면 스택의 함수 호출이 계속해서 쌓이다가 최대 콜스택 사이즈 초과(Maximum call stack size exceeded) 가 된다. 그런 경우엔 대기시간이 길어질 수밖에 없다. 다음 작업이 실행되기 위해서 전작업이 끝날때까지 기다려야 하기 때문이다.

이런것처럼 다른 작업을 실행하기 위해서 전 작업이 완료될때까지 기다려야 하는 현상을 '블로킹' 이라고 한다.

= 동기 처리 , synchronous

장점 : 실행 순서 보장

단점 : 앞의 태스크 종료까지 이후 태스크들이 블로킹됨

예시 ) sleep 함수

😊 비동기

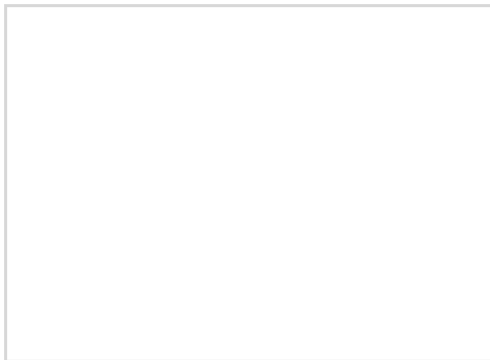
현재의 실행태스크가 종료되지 않더라도 바로 다음 태스크가 실행이 된다. '블로킹'이 되지 않으므로, 작업 중단이 되지 않는 '논-블로킹'이라 한다.

```
console.log(1);
```

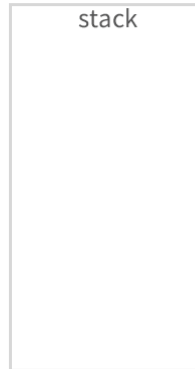
```
setTimeout(function cb () {  
  console.log(2);  
}, 5000);
```

```
console.log(3);
```

console



stack



web APIs



event loop

task queue

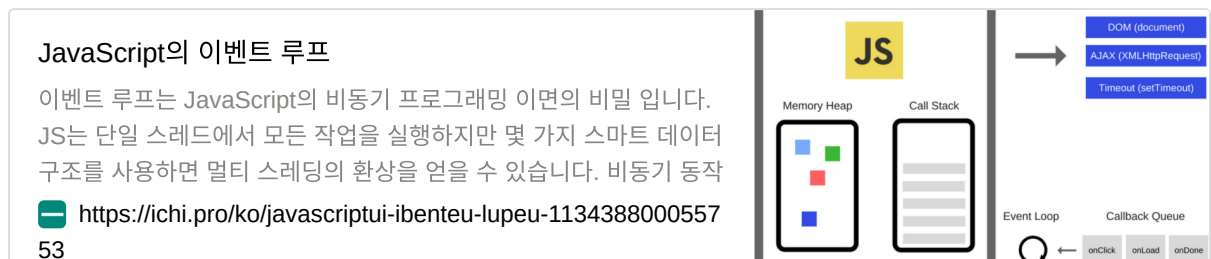
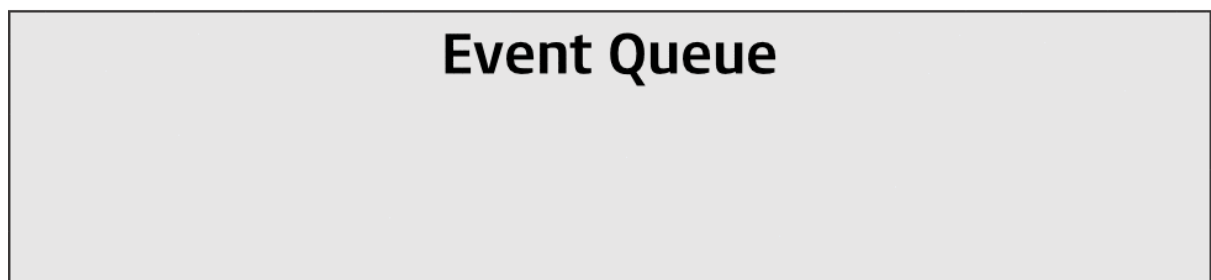
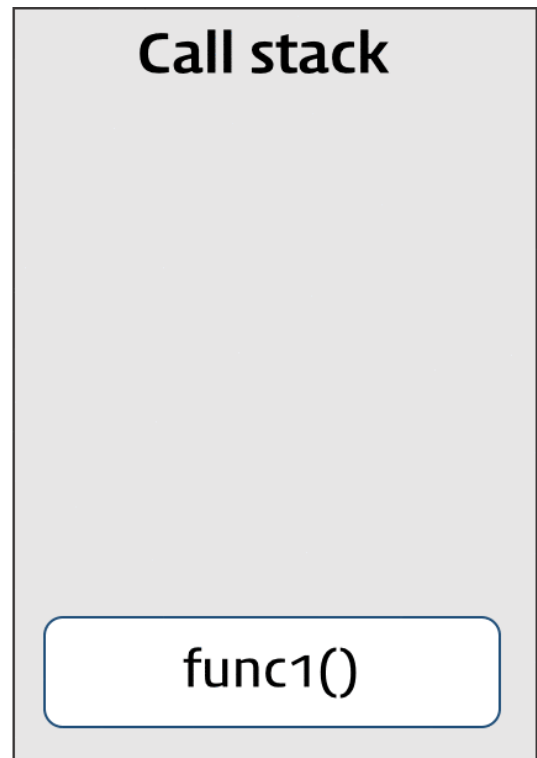
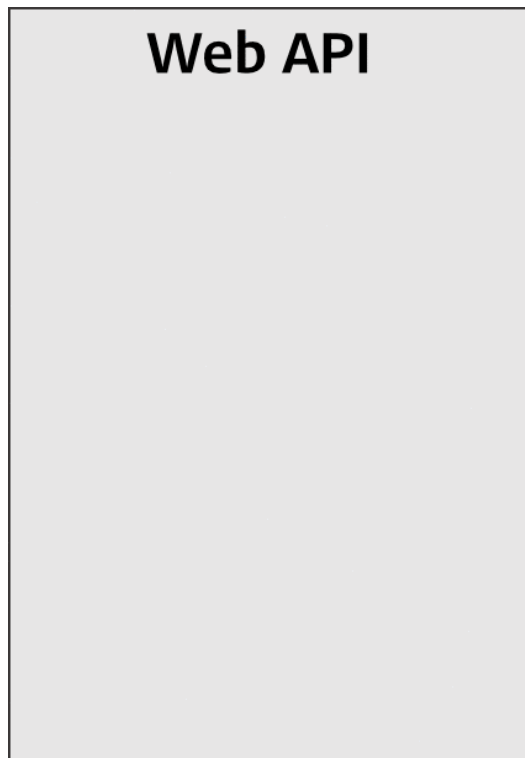


장점 : 현재 실행 중인 태스크가 종료되지 않았다고 해도 다음 태스크를 곧바로 실행하므로 블로킹이 발생하지 않음(Non-blocking)

단점 : 태스크의 실행 순서 보장X, 콜백지옥 유발 가능성 있음

예시 ) setTimeout, setInterval, HTTP 요청, 이벤트 핸들러

## 😄 이벤트 루프와 태스크 큐



### ▼ 콜 스택

평가 과정에서 생성된 실행 컨텍스트가 추가되고 제거되는 스택 자료구조인 실행 컨텍스트 스택이 바로 '콜 스택'이다.

자바스크립트엔진은 단 하나의 콜 스택을 사용하기 때문에 최상위 실행 컨텍스트가 종료되어 콜 스택에서 제거되기 전까지는 다른 어떤 태스크도 실행 되지 않는다.

### ▼ 힙

힙은 객체가 저장되는 메모리 공간이다. 콜 스택의 요소인 실행 컨텍스트는 힙에 저장된 객체를 참조한다. 메모리 값을 저장하려면 먼저 값을 저장할 메모리 공간의 크기를 결정해야 한다.

또 객체는 원시값과는 다르게 크기가 정해져있지 않기에 할당해야 할 메모리 공간의 크기를 런타임에 동적으로 결정 해야 한다. 따라서 객체가 저장되는 메모리 공간인 힙은 구조화 되어 있지 않다는 특징이 있음.

#### ▼ 태스크 큐(= **task queue**, **event queue**, **callback queue**)

setTimeout이나 setInterval 같은 비동기함수의 콜백 함수 혹은 이벤트 핸들러가 일시적으로 보관됨

참고) macrotask queue : 프로미스의 후속 처리 메서드의 콜백 함수가 임시적으로 보관됨

#### ▼ 이벤트 루프

이벤트 루프는 콜스택 현재 실행중인 실행 컨텍스트가 있는지, 그리고 태스크 큐에 대기중인 함수(콜백 함수, 이벤트 핸들러 등) 가 있는지 확인한다. 만약 콜 스택이 비어 있고 태스크 큐에 대기중인 함수가 있다면 이벤트 루프는 순차적 태스크 큐에 대기중인 함수를 콜스택으로 이동시킨다.

콜스택으로 이동된 함수는 실행이 된다. 즉, 태스크 큐에 임시 보관된 함수들은 비동기 처리 방식으로 동작한다.

자바스크립트는 싱글 스레드 방식으로 동작한다. 하지만 싱글스레드 방식으로 동작한다는 것은 브라우저가 아닌 브라우저에 내장된 '자바스크립트 엔진' 이라는 것이다. 만약 모든 자바스크립트 코드가 자바스크립트 엔진에서 싱글 스레드 방식으로 동작한다면 자바스크립트는 비동기로 동작할 수 없다. 즉, 자바스크립트 엔진은 싱글 스레드로 동작하지만 브라우저 멀티 스레드로 동작한다.