

블록/논블록 동기/비동기

에스이랩 박연구

2022-08-04

블록/논블록

- 함수 호출에 대한 것
- 블로킹 (blocking)
 - 함수A 를 호출했는데, 수행이 모두 끝날 때 까지 기다린다.
 - 함수A가 끝나고 리턴되면, 함수A 호출 다음 부분부터 이어서 진행된다.
- 논-블로킹(non-blocking)
 - 함수A를 호출했는데, 함수A의 실행을 요청하고 바로 리턴된다

블록/논블록

- 블로킹 (blocking)

```
console.log("count 1");  
console.log("count 2");  
console.log("count 3");
```

```
function doWork() {  
  let count = 1;  
  while (count < 10) {  
    console.log(count);  
    count ++;  
  }  
}
```

```
console.log("start");
```

```
doWork();
```

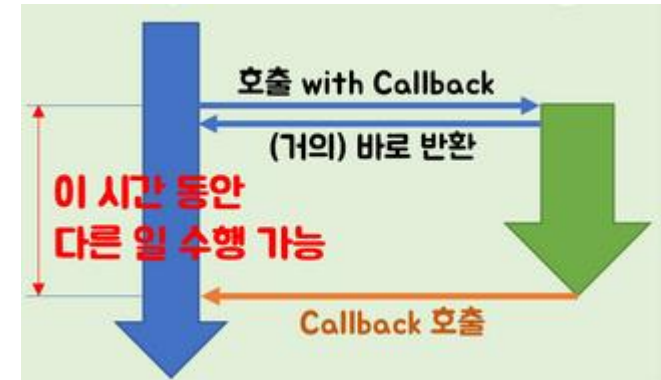
```
console.log("end");
```

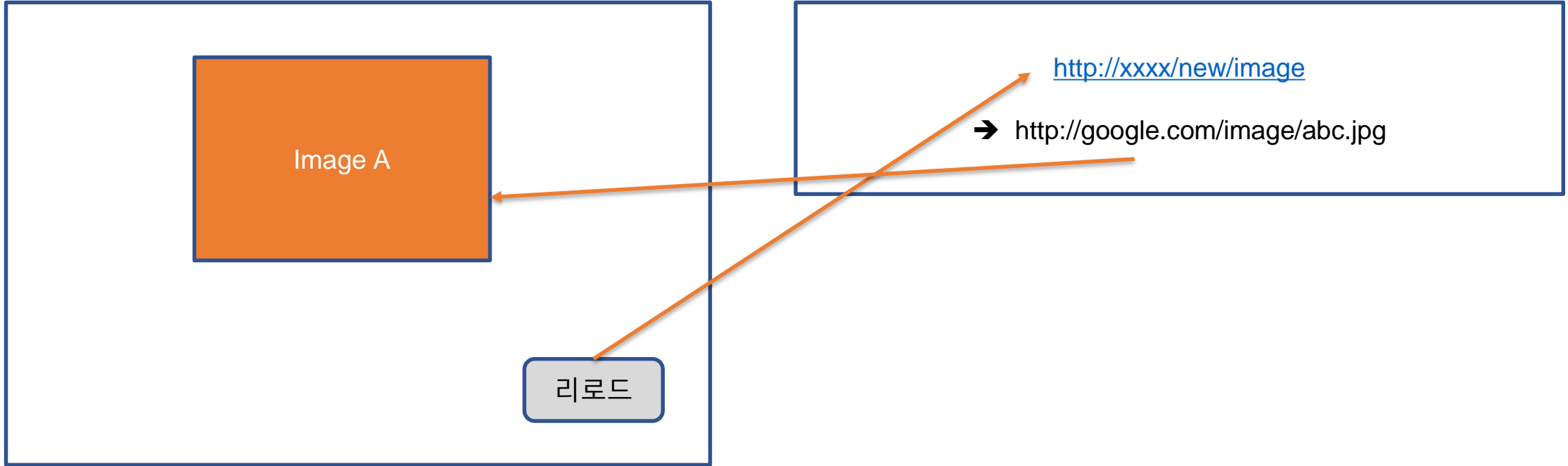


블록/논블록

- 논-블록킹(non-blocking)

```
function getJob(id, callback) {  
  $.ajax({  
    url: "/api/job/job/" + id,  
    type: "get",  
    success: function(result) {  
      JobStoreManager.prototype._jobMap.delete(id);  
      callback(result);  
    }  
  }); // end of ajax  
}  
  
console.log("start");  
  
getJob('jobId1', (result) => {  
  console.log(result);  
});  
  
console.log("end");
```





동기/비동기

- 행위에 대한 것
- 동기 (Synchronous)
 - 작업A가 먼저 모두 처리되고 나서 작업 B가 처리되기 시작
 - 하나씩 순차적 처리
- 비동기 (Asynchronous)
 - 작업A와 작업B가 동시에 처리
 - A, B 사이에 인과관계가 있어야 함

동기/비동기

Operations to be executed

FETCHING DATA
FROM API

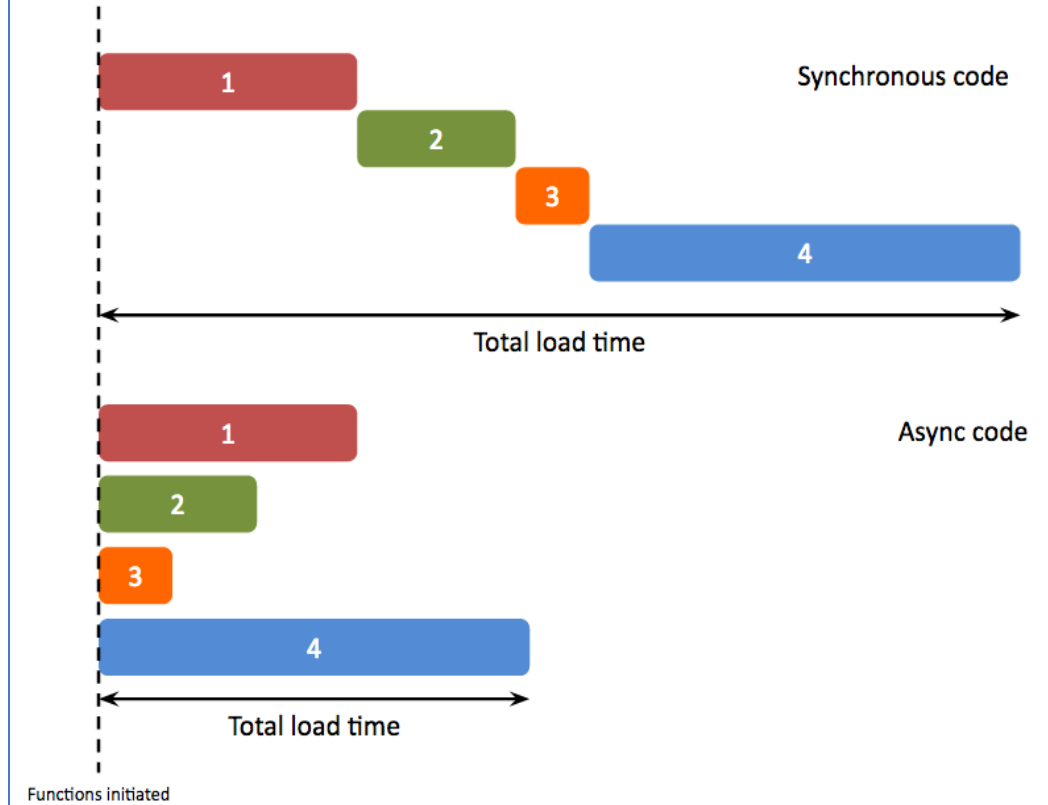
SET TIMEOUT
FOR 5 SECONDS

DATABASE
OPERATION

Synchronous



Asynchronous



참고 링크

- 블록/논블록, 동기/비동기
<https://velog.io/@taehee-kim-dev/%EB%B8%94%EB%A1%9D%EB%85%BC%EB%B8%94%EB%A1%9D-%EB%8F%99%EA%B8%B0%EB%B9%84%EB%8F%99%EA%B8%B0>
- 블럭,논블럭,동기,비동기 이야기
<https://hamait.tistory.com/930>
- Node.js: 비동기 프로그래밍 이해
<https://www.nextree.co.kr/p7292/>
- Blocking/Non-Blocking + Sync/Async 조합 정리
<https://velog.io/@kjh3865/BlockingNon-Blocking-SyncAsync-%EC%A1%B0%ED%95%A9-%EC%A0%95%EB%A6%AC>

끝