

JS 비동기 프로그래밍



❖ 학습해야 할 내용

- Blocking – Non Blocking 함수 이해.
- Promise 객체 사용법의 이해.

1. JS 에는 non-blocking 한 함수들이 존재한다. 이는 JS 가 싱글 스레드로 동작 하기 때문이다. blocking 함수와 non-blocking 함수의 차이를 약속하고 만약 JS 의 모든 함수들이 blocking 하다면, 어떤 일이 일어날지 작성하시오.
2. Promise 는 non-blocking 한 작업들을 처리할 때, 지금 당장이 아닌 미래에 어떤 값을 return 하기 위해 사용한다. 이런 작업들은 끝날 때까지 결과를 알 수 없지만, 결과가 무조건 성공/실패 둘 중 하나라는 것은 확신할 수 있다. 이 때, 아래 코드를 다음 키워드를 포함하여 설명하시오.
[성공(원하는 대로), 실패(그 외의), resolve, reject, then, catch, 익명함수]

```
function sendXHR(url) {
  return new Promise((resolve, reject) => {
    XHR = new XMLHttpRequest();
    XHR.open('GET', url);

    /*
     * XHR.onload 에 함수를 저장하면,
     * XHR 요청의 응답이 도착했을 때 해당 함수가 실행된다.
     */
    XHR.onload = () => {
      // XHR.status 에는 응답의 HTTP 상태 코드가 담겨있다.
      if (XHR.status === 200) {
        // XHR.responseText 에는 응답으로 받은 내용이 담겨있다.
        resolve(XHR.responseText);
      } else {
        reject(XHR.status)
      }
    };

    XHR.send()
  })
}

sendXHR('http://SOME_URL.com')
  .then(result => console.log(`Good: ${result}`))
  .catch(statusCode => console.log(statusCode));
```