**웹 시스템 설계 – 7주차 학습 보고서 2 (2020년 10월 14일)**

|  |  |  |
| --- | --- | --- |
| 성명 : 유 상 정 | 학과 : 사이버보안학과 | 학번 : 201620641 |

※ 본 학습보고서를 작성함에 있어 다른 학생의 문서로부터 일부 또는 전체를 복사하였습니까?

예( ) 아니오(✓) (복사 하였다면 예에 체크하고 아니라면 아니오에 체크하시오)

**I. Objectives**

* Promise 객체를 사용하여 Asynchronous 한 task를 정의할 수 있다.

**II. Exercises**

* 다음 Promise를 사용한 코드에서 밑줄 친 (resolve 함수와 reject함수) 코드의 역할에 대해서 설명하시오.

|  |
| --- |
| function countdown(seconds) {  return new Promise(function(resolve, reject) {  for(let i=seconds; i>=0; i--) {  setTimeout(function() {  if(i===13) return reject(new Error("DEFINITELY NOT COUNTING THAT"));  if(i>0) console.log(i + '...');  else resolve(console.log("GO!"));  }, (seconds-i)\*1000);  }  });  } |

|  |  |
| --- | --- |
| **resolve 함수의 역할** | Promise 객체 생성자는 매개 변수로 실행 함수를 받는다. 이때, 첫번째 함수, resolve의 역할은 비동기 작업이 성공적으로 완료되어 결과를 값으로 반환한다. 또한 Promise가 처음 생성되면 상태가 Pending인데, resolve함수를 수행하면 Fulfilled 상태로 바꾼다. 그래서 for문을 다 돌고서, i값이 0이되면 resolve 함수를 수행하여, promise상태를 ‘fulfilled’로 바꾸고, “Go!”를 호출한 후, 결과를 값으로 반환한다. |
| **reject 함수의 역할** | Promise 객체 생성자는 매개 변수로 실행 함수를 받는다. 이때, 두 번째 함수, reject의 역할은 비동기 작업이 실패하면 오류의 원인을 반환하고 호출한다. 또한 비동기 작업이 실패하여 reject를 호출하면 promise의 상태를 ‘rejected’로 변경한다. 그래서 for문에 의해 i값이 13이 되면, Promise의 상태를 변경하고 Error문을 console창에 출력한다. |