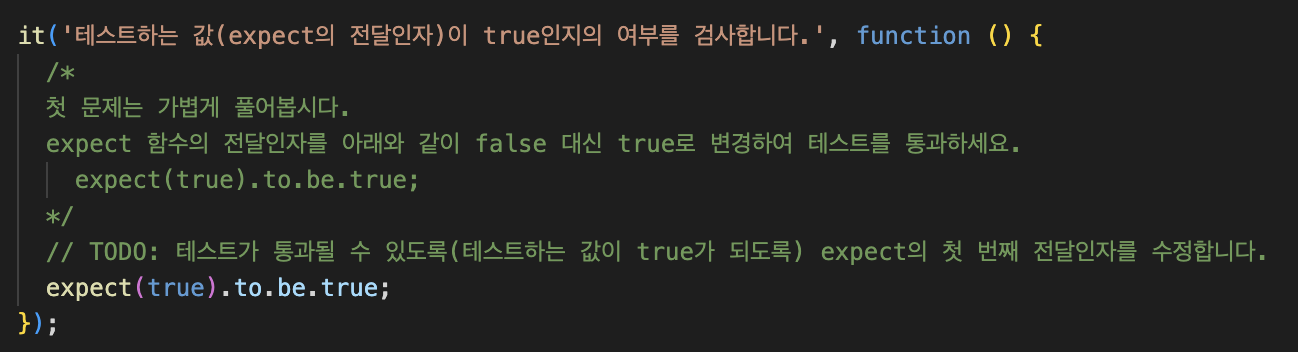
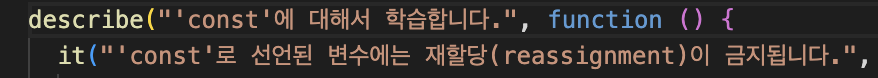
[2023.03.06]-Node.js 기반 테스트프레임워크 mocha를 활용한 과제



이번 과제는 테스트 케이스를 활용해 주어진 문제들의 빈칸에 적절한 답을 입력하고, 그 결과를 테스트해보면서 지금까지 배워온 여러 가지 개념들을 체크해보는 시간이었다. 이 과제를 통해 점검해 볼 수 있었던 개념은 크게 8가지였다.

1. **타입**
2. **변수**
3. **스코프와 클로저**
4. **화살표 함수**
5. **원시 자료형과 참조 자료형**
6. **배열과 객체**
7. **전개연산자**
8. **구조 분해 할당**



https://blog.kakaocdn.net/dn/bvfpFG/btr2sXmlNpO/xCawiEeWN7UFeTHKvX1xG0/img.png

위 이미지에 쓰여있는 코드처럼 아주 기초적인 개념부터 다시 차근차근 되새겨봄과 동시에, 이해하기 어렵고 어떻게 작동하는지 알아채기 쉽지 않은 부분들까지 공부하면서 지금까지 배운 내용을 머릿속에 정립해가는 과정이었다.

*// \*함수 선언문으로 작성된 코드는 변수 호이스팅이 발생한다\*//*

함수 표현식으로 사용하는것이 바람직하다!

(회사별 코드 컨벤션에 따라 조금씩 다르지만)

### 3. 스코프와 클로저

클로저는 함수와 함수가 선언된 어휘적 환경의 조합을 말한다.

이 환경은 클로저가 생성된 시점의 유효 범위 내에 있는 모든 지역 변수로 구성된다.

여기서의 키워드는 "함수가 선언"된 "어휘적(lexical) 환경"입니다.

특이하게도 자바스크립트는 함수가 호출되는 환경와 별개로, 기존에 선언되어 있던 환경 - 어휘적 환경 - 을 기준으로 변수를 조회하려고 합니다.

유어클레스 영상에서 언급되는 "외부함수의 변수에 접근할 수 있는 내부함수"를 클로져 함수로 부르는 이유도 그렇습니다.

클로저는 내부(inner) 함수가 외부(outer) 함수의 지역 변수에 접근할 수 있습니다.

.

스코프와 클로저 개념은 아직도 정확하게 이해가 되지 않는 난해한 개념이다. 함수 내부에서 선언된 변수를 외부에서 사용할 수 없다는 것과 그 반대인 외부에서 선언된 변수를 내부에서 사용할 수 있다는 점, 변수가 선언되는 위치에 따라 갖는 의미가 다르다는 것, 전역 변수의 위험성과 로컬 변수의 우선 순위 등 실제로 코드를 실행시켜 보면서 확인해보아야 확실하게 이해할 수 있을 것 같은 부분이 많다.

### 4.  화살표 함수

it('화살표 함수를 이용해 클로저를 표현합니다', function () {

const adder = x => {

return y => {

return x + y

}

}

expect(adder(50)(10)).to.eql(60)

const subtractor = x => y => {

return x - y

}

expect(subtractor(50)(10)).to.eql(40)

화살표 함수는 함수 스코프가 아니라 블록 스코프로 취급된다는 점을 기억하고, 상황에 따라 리턴이나 소괄호 등을 생략하는 용법에 익숙해지도록 계속 사용해보아야겠다.

### 5. 원시 자료형과 참조 자료형

Object 자료형은 데이터는 heap에 저장되고, 변수에 할당을 할 경우 변수에는 주소가 저장된다.

1) [1, 2, 3]; *// [1, 2, 3]이라는 데이터가 heap에 저장되지만 변수 할당이 되지 않아 주소는 어디에도 저장되지 않는다.*

2) const num1 = [1, 2, 3]; *// // [1, 2, 3]이라는 데이터가 heap에 저장되고, 그 주소가 변수 num1에 저장된다.*

3) const num2 = [1, 2, 3]; *// // [1, 2, 3]이라는 데이터가 heap에 저장되고, 그 주소가 변수 num2에 저장된다.*

1), 2), 3)에서 말하는 주소는 전부 다른 주소입니다.

원시 자료형과 참조 자료형의 특징을 저번에 배웠지만, 참조 자료형을 여러 가지 메서드나 속성을 통해 다뤄보면서, 같은 값을 가지는 두 변수나 주솟값이 같거나 다른 변수, 참조 자료형이 할당된 변수의 복사(얕은 복사와 깊은복사) 등 디테일한 부분에서 굉장히 헷갈리는 요소가 많은 부분이었다. 자바스크립트에서 객체는 빼놓을 수 없는 핵심 요소이기 때문에 참조 자료형을 충분히 많이 다루어보아야 할 필요성을 한 번 더 느낄 수 있었다.

### 6. 배열과 객체

배열과 객체에 대한 기본 개념 및 특징을 다시 한 번 복습한 후에 배열과 객체를 함수의 전달인자로 전달하는 경우, 주솟값을 참조한다는 중요한 개념을 알 수 있었다. 또한 this라는 개념을 접하게 되었는데 조금 심화된 내용이지만 확실하게 정리해두어야 하는 개념이라는 생각이 들었기에 필요한 자료를 더 찾아보게 되었다.

### 7. 전개연산자

### 

8. 구조 분해 할당

