

15장. let, const 키워드와 블록 레벨 스코프

15.1 var 키워드로 선언한 변수의 문제점

1. 변수 중복 선언 허용
2. 오로지 함수의 코드 블록만을 지역 스코프로 인정
 - 함수 외부에서 var로 선언한 변수는 코드 블록 내에서 선언해도 모두 전역 변수가 됨
3. 변수 호이스팅에 의해 변수 선언문이 스코프의 선두로 끌어 올리진 것처럼 동작함

15.2 let 키워드

이름이 같은 변수를 중복 선언하면 문법 에러 발생

모든 코드 블록을 지역 스코프로 인정하는 블록 레벨 스코프를 따름

변수 호이스팅이 발생하지 않는 것처럼 동작

→ let 키워드는 선언 단계와 초기화 단계가 분리되어 진행

- 일시적 사각지대란?

스코프의 시작 지점부터 초기화 시작 지점까지 변수를 참조할 수 없는 구간

15.3 const 키워드

상수를 선언하기 위해 사용(재할당이 금지된 변수)

const 키워드로 선언한 변수는 반드시 선언과 동시에 초기화해야 한다

let 키워드처럼 블록 레벨 스코프를 가지며 변수 호이스팅이 발생하지 않는 것처럼 동작

→ const 키워드로 선언된 변수에 원시 값을 할당한 경우 재할당 불가능

→ const 키워드로 선언된 변수에 객체를 할당한 경우 값 변경 가능

가능한 이유: const 키워드는 재할당을 금지할 뿐 "불변"을 의미하는 것은 아님

15.4 var vs. let vs. const

-ES6를 사용한다면 var 키워드 사용하지 않기

-재할당이 필요한 경우에 한정해 let 키워드를 사용한다. 이때 변수의 스코프는 최대한 좁게 만들기

-변경이 발생하지 않고 읽기 전용으로 사용하는 원시 값과 객체에는 const 키워드를 사용

→ const 키워드는 재할당을 금지하므로 var, let 키워드보다 안전