

변수는 프로글매의 어느 곳에서도 선언하여 사용할 수 있다. 다만 변수는 선언된 위치에 따라 그 변수가 영향을 미치는 범위가 한정된다. 이와 같이 변수가 적용되는 범위를 유효범위(scope)라고 한다. 변수는 유효 범위에 따라 '전역 변수'와 '지역 변수'로 구분된다.

① 双열問令

함수 내부에 선언된 변수를 지역변수라고 한다. 함수 안에서만 적용되는 변수이다. 따라서 함수가 실행될때 생성되고, 함수의 실행이 종료되면 변수의 유효성이 사라진다. 따라서 지역 변수는 함수 밖에서 호출할수 없으며 함수가 실행될때마다 변수의 값이 초기화되다. 지역 변수를 사용하기 위해서는 함수 안에 'var'을 붙여서 새롭게 선언해야한다. 이렇게 선언하지 않으면 전역 변수를 참조하는 경우가 된다.

📆 실습하기

① 전역변수

전역 변수는 함수 밖에 변수를 정의하여 함수 밖은 물론 함수 내에도 영향을 미치는 변수이다. 전역변수와 지역 변수가 같은 이름으로 명시되면 함수 내에서도 지역 변수가 우선적으로 적용된다.

📆 실습하기

O Ver

특징 및 단점

- 1. 호이스팅 문제: var로 선언된 변수는 호이스팅되어 변수 선언이 코드의 최상단으로 끌어올려지지만, 초기화는 그대로 유지된다. 이로 인해 선언 이전에 변수를 참조할 경우 undefined가 반환되어 혼란을 초래할 수 있다.
- 2. 블록 스코프 없음: var는 if 문이나 for 루프와 같은 블록 내에서 선언된 변수도 블록 외부에서 접근할 수 있다. 이로 인해 예기치 않은 동작이 발생할 수 있다.
- 3. var 키워드 생략: 함수 내에서 var 키워드 없이 사용하면 전역 변수가 생성된다. 이는 다른 코드와 충돌할 수 있으며, 전역 변수가 예상치 못한 변경을 겪을 수 있다.
- 4. 재선언 가능: 같은 스코프 내에서 동일한 이름으로 변수를 여러 번 선언할 수 있다. 이로 인해 코드의 가독성이 떨어지고, 의도치 않은 오류가 발생할 수 있다.

1 let, const

특징 및 개선

- 1.호이스팅되지만 초기화 전 접근 시 ReferenceError가 뜬다.
- 2. 블록 스코프: 블록은 {}로 바인딩된 코드 묶음인데 , if문 for 루프 포함하여 중괄호 속에서 변수 선언되었다면 해당 범위에서만 유효하다.
- 3. 선언위치 : 전역 위치에서 선언, 블록 위치에서 선언, 함수 위치에서 선언. 선언위치에 따라 유효범위가 달라진다.
- 4. 재선언 불가능: 같은 스코프 내에서 동일한 이름으로 변수를 여러 번 선언할 수 없다.

D Ref

var	let	const
호이스팅 0	호이스팅 0	호이스팅 0
변수 선언	변수 선언	상수 선언 (동시에 초기화 필수)
업데이트 가능	업데이트 가능	업데이트 부분적 가능 (개체는 불가능, 속성은 가능)
전역/지역 스코프	전역/지역/블록 스코프	전역/지역/블록 스코프
같은 범위 재선언 가능	같은 범위 재선언 불가능	같은 범위 재선언 불가능
선언 이전 접근 가능 (undefined)	선언 이전 접근 불가능 (ReferenceError)	선언 이전 접근 불가능 (ReferenceError)