

# TDZ(Temporal Dead Zone)

👤 배정	
▼ 상태	8월
🕒 속성	@2021년 8월 14일 오후 11:47
▼ 언어	

## TDZ(Temporal Dead Zone)란 무엇일까?

그 전에 우선 변수 선언의 3단계를 알아보자.

### 변수 선언의 3단계

변수는 선언, 초기화, 할당 이렇게 3단계를 거쳐 생성된다.

## 변수의 라이프사이클

선언 단계(Declaration phase)

초기화 단계 (Initialization phase)

할당 단계(Assignment phase)

- **선언 단계(Declaration phase)** : 변수를 실행 컨텍스트의 변수 객체에 등록한다.  
→ 스코프가 참조하는 대상
- **초기화 단계(Initialization phase)** : 실행 컨텍스트에 존재하는 변수 객체에 선언 단계의 변수를 위한 메모리를 만든다.  
→ undefined로 초기화된다.

- 할당 단계(Assignment phase) : 사용자가 undefined로 초기화된 메모리에 다른 값을 할당한다.

## TDZ(Temporal Dead Zone)의 정의

TDZ는 스코프의 시작 지점부터 초기화 시작 지점까지의 사각지대 구간을 뜻한다. 비유하자면 선언 단계와 초기화 단계 사이에 잠시 머물러 있는 것이라고 할 수 있다.

## var, let/const 변수의 라이프사이클

### var 변수의 라이프사이클

### var 변수의 라이프사이클

변수 이름을 food 라고 짓는다고 가정해본다.

선언 단계(Declaration phase)  
초기화 단계 (Initialization phase)

food === undefined

초기화 된 상태  
(Initialized state)

food = '고기'

할당 단계(Assignment phase)

food === '고기'

할당된 상태  
(Assigned state)

1. var는 변수 선언 전에 선언 단계와 초기화 단계를 동시에 실행한다.

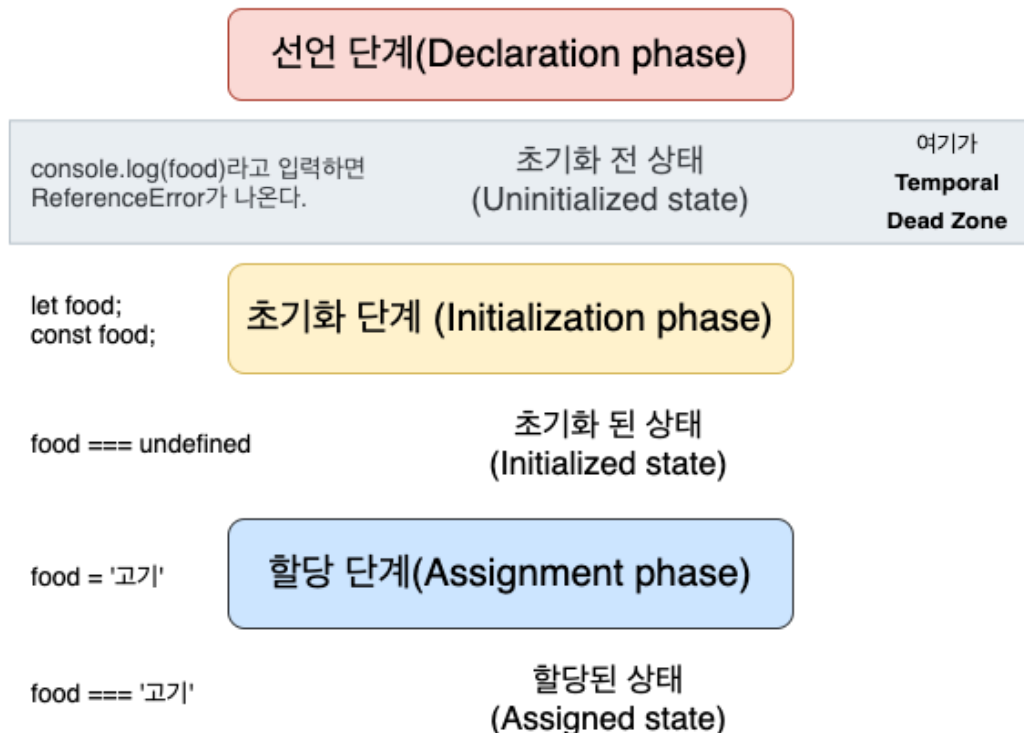
2. 선언과 초기화가 동시에 실행되기 때문에 실행 컨텍스트 변수 객체를 등록 후 바로 그 메모리를 undefined 한다.

→ 그래서 선언하기도 전에 `undefined` 로 호출이되는 호이스팅이 발생한다.

## let/const 변수의 라이프사이클

### let / const 변수의 라이프사이클

변수 이름을 food 라고 짓는다고 가정해본다.



여기서 `let food;` 와 `const food;` 는 food 라는 변수를 각각 `let`과 `const`로 선언했을 때를 분리했을 때를 가정한것이다.

`let`과 `const`는 `var`와는 다르게 선언단계와 초기화 단계가 따로 분리되어 실행된다.

그래서 선언 단계와 초기화 단계 사이에서는 실행 컨텍스트에는 변수를 등록했지만 메모리가 할당되지 않은 상태라 `ReferenceError` 가 나오는 것이다.

이런 사각지대를 **TDZ(Temporal Dead Zone)**라고 하는 것이다.

## 결론

1. TDZ는 변수가 선언되고 초기화되기 사이의 사각지대이다.
2. var는 변수의 선언단계와 초기화단계가 동시에 실행되어 TDZ가 존재하지 않기 때문에 호이스팅이 일어나는 것이다.
3. let과 const는 초기화 단계에 오면 TDZ 상태가 해제된다.