2023.04.13

변수 선언
데이터 할당

이도하와 김도경

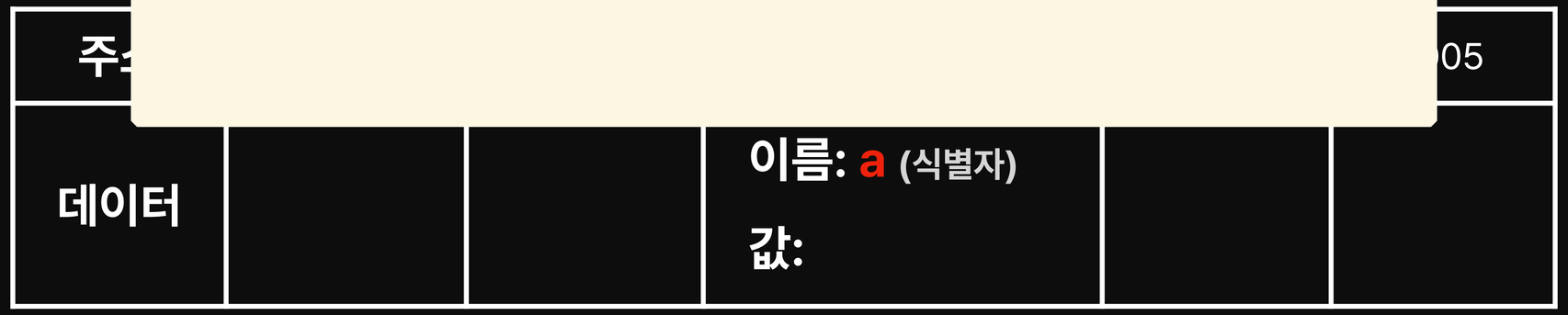
변수 선언이란?

변수를 생성하

변수를 선언하

변수를 선언하

```
var a;    // 변수에 식별자를 a로 지정

a = 'abc'; // 변수 a에 데이터 할당
```



변수 선언이란?

변수를 생성하는 것을 말한다.

변수를 선언하기 위해서는 var, let, const 키워드를 사용한다.

변수를 선언하면 메모리의 할당은 어떻게 될까?

주소	...	1002	**1003**	1004	1005
데이터			이름: a (식별자) 값:		

둘은 정확히 동일하게 동작한다.

Why?

선언 단계와 초기화 단계, 이렇게 2단계를 거쳐 수행하기 때문.

변수 선언은 소스코드가 순차적으로 실행되는 시점인 런타임 이전에 먼저 실행,

값의 할당은 소스코드가 순차적으로 실행되는 시점인 런타임에 실행된다.

실행방법은 호이스팅을 알면 된다. (뒤에서 계속 😉)

var a = 'abc' 는 선언 단계와 할당까지 한번에 수행한다. (변수 초기화)

선언 단계에서 변수 영역에 메모리 공간을 확보하고

초기화 단계에서 별도의 메모리 공간을 다시 확보해서 데이터를 저장한다.

```
var a;

a = 'abc';

// 변수 a에 데이터 할당

var a = 'abc';

// 선언과 할당을 한문장으로 표현
```

변수 데이터 할당

데이터 할당 과정

주소	...	1002	1003	1004	1005
데이터			이름: a 값: @5003		

데이터의 위치를 가리킴

주소	...	5002	5003	5004	5005
데이터			'abc'		

왜 변수 영역에 바로 대입하지 않고 따로 저장하는 걸까?

1. 데이터 변환을 자유롭게 할 수 있게 한다.

2. 동시에 메모리를 더욱 효율적으로 관리하기 위함이다.

주소	...	1002	**1003**	1004	1005
데이터			이름: a 값: @5003		

주소	...	5002	**5003**	5004	5005
데이터			abc'		

선언 단계에서 변수 영역에 메모리 공간을 확보하고

초기화 단계에서 별도의 메모리 공간을 다시 확보해서 데이터를 저장한다.

변수 데이터 재할당

이전 값이 저장되어 있

새로운 메모리 공간을

이전 값이 저장되어 있

```
var a = 'abc';

a = 'New';

// 새로운 친구 !
```

주소					1005
데이터					

이 변경

주소					5005
데이터			'abc'	‘New’	

변수 데이터 재할당

이전 값이 저장되어 있던 메모리 공간을 지우고 그 메모리 공간에 재할당 값을 새롭게 저장하는 것이 아니라

새로운 메모리 공간을 확보하고 그 메모리 공간에 새로운 값을 저장한다.

이전 값이 저장되어 있던 메모리 공간은 **가비지 콜렉터**에 의해 자동 해제된다.

주소	...	1002	**1003**	1004	1005
데이터			**이름: a** **값: @5004**		

데이터를 가리키는 곳이 변경

주소	...	5002	5003	**5004**	5005
데이터			**'abc'**	**'New'**	

모든 변수가 이렇게 재할당이 될까?

변경이 불가능한 const

변수에 저장된 값을 변경할 수 없고, 값을 재할당 할 수 없는 변수를 상수(constant)라 한다.

const 키워드를 사용해 상수를 표현 할 수 있다.

```
const animal = "capybara";
```

```
animal = "elephant";
```

변경이 불가능한 const

변수에 저장된 값을 변경할 수 없고, 값을 재할당 할 수 없는 변수를 **상수(constant)**라 한다.

const 키워드를 사

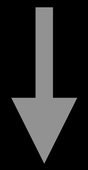
> Uncaught TypeError: Assignment to constant variable.
> at <anonymous>:1:8

```
const animal = "capybara";
```

```
animal = "elephant";
```

호이스팅

```
test = 10;
console.log(test);

var test;
```



```
Console
10
```

호이스팅

: 변수와 함수의 메모리 공간을 선언 전에 미리 할당하는 것
(변수의 선언과 초기화를 분리한 후, 선언만 코드의 최상단으로 옮기는 것)

var로 선언한 변수는 호이스팅으로 인해 선언을 나중에 해도 10이 출력된다.

호이스팅

```
test = 10;
console.log(test);

let test;
```

```
ation
2e-fc45-39bd-39322dbc4128:1
```

NO!

let과 const로 선언한 변수도 호이스팅이 일어난다!

왜 에러가 나는지 알기 위해서는 **TDZ(Temporal Dead Zone)**에 대해서 알아야 한다.

하

다.

그럼 let 키워드로 선언하면 호이스팅이 일어나지 않는걸까?

호이스팅

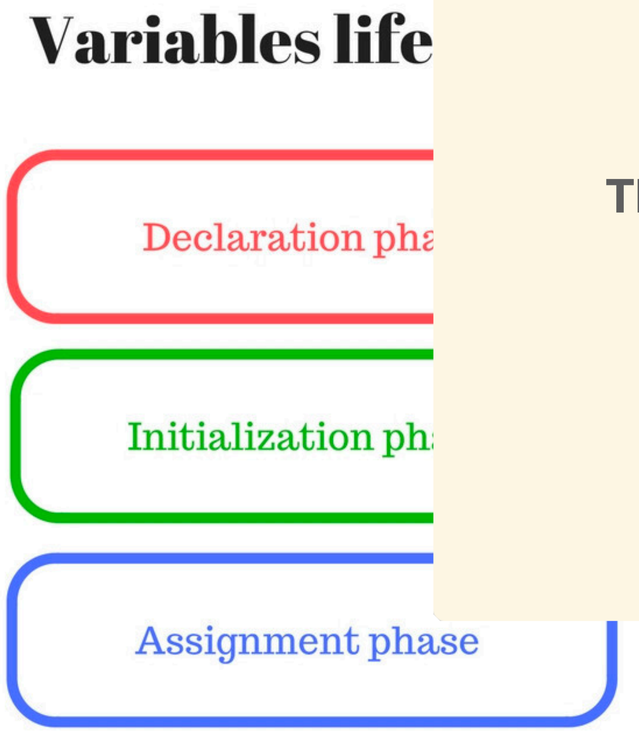
```
test = 10;
console.log(test);

let test;
```

```
Uncaught ReferenceError: Cannot access 'test' before initialization
 at https://cdpn.io/cpe/boomboom/pen.js?key=pen.js-706a8e58-7a2e-fc45-39bd-39322dbc4128:1
```

하지만 let으로 선언한 test는 참조 에러(reference error)가 발생한다.

그럼 let 키워드로 선언하면 호이스팅이 일어나지 않는걸까?

TDZ



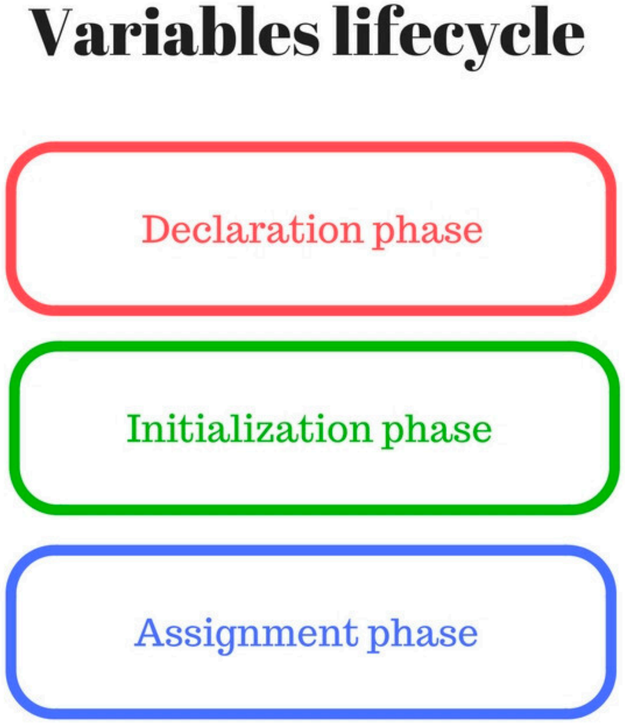
TDZ(Temporal Dead Zone) :: 사각지대

: 스코프의 시작 지점부터 초기화 시작 지점까지의 구간을 의미

할당 단계(Assignment phase)

: 사용자가 undefined로 초기화된 메모리의 다른 값을 할당하는 단계이다.

TDZ

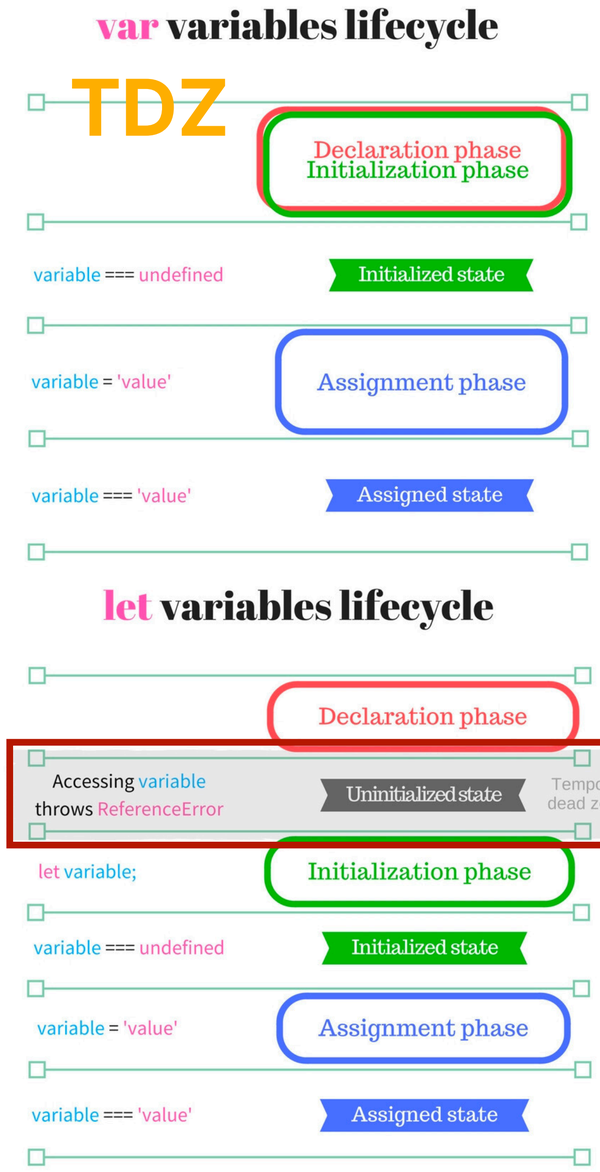


JavaScript에서 변수는 3단계에 걸쳐서 생성된다

선언 단계(Declaration phase)
: 변수를 실행 컨텍스트의 변수 객체에 등록하는 단계를 의미한다. 이 변수 객체는 스코프가 참조하는 대상이 된다.

초기화 단계(Initialization phase)
: 실행 컨텍스트에 존재 하는 변수 객체에 선언 단계의 변수를 위한 메모리를 만드는 단계이다. 이 단계에서 할당된 메모리에는 undefined로 초기화 된다.

할당 단계(Assignment phase)
: 사용자가 undefined로 초기화된 메모리의 다른 값을 할당하는 단계이다.



두 키워드의 라이프사이클을 살펴보면

var 키워드는 **선언단계와 초기화 단계가 함께 진행**되지만
let const 등의 키워드는 **선언단계와 초기화 단계가 분리되어 진행**된다.

실행 컨텍스트에 **변수 등록은 되었지만(선언은 되었지만)**
메모리가 할당(초기화) 되지 않아서 참조 에러가 발생하는 것이다.

그래서 우리는 호이스팅이 되지 않는다고 오해하게된 것.

TDZ(Temporal Dead Zone) :: 사각지대

: **스코프의 시작 지점부터 초기화 시작 지점까지의 구간**을 의미

let과 const는 TDZ의 영향을 받아서 **할당 하기전에는 사용할 수 없음**

* 참고로 function 키워드는 선언, 초기화, 할당 단계를 동시에 진행한다.

* 가비지 콜렉터

: 할당한 메모리 공간을 주기적으로 검사하여
더 이상 사용되지 않는 메모리를 해제 하는 기능

C언어와 같은 **언매니지드 언어**의 경우
개발자가 명시적으로 할당하고 해제해줘야 하는데,

자바스크립트의 경우
가비지 콜렉터를 내장하고 있는 **매니지드 언어**로서
자동으로 메모리 누수(leak)를 방지한다

2023.04.13

E.O.D

이도하와 김도경