

10분 세미나

실행 컨텍스트 를 알아보자 🤔

Mash-up 10기 노드팀 임지영

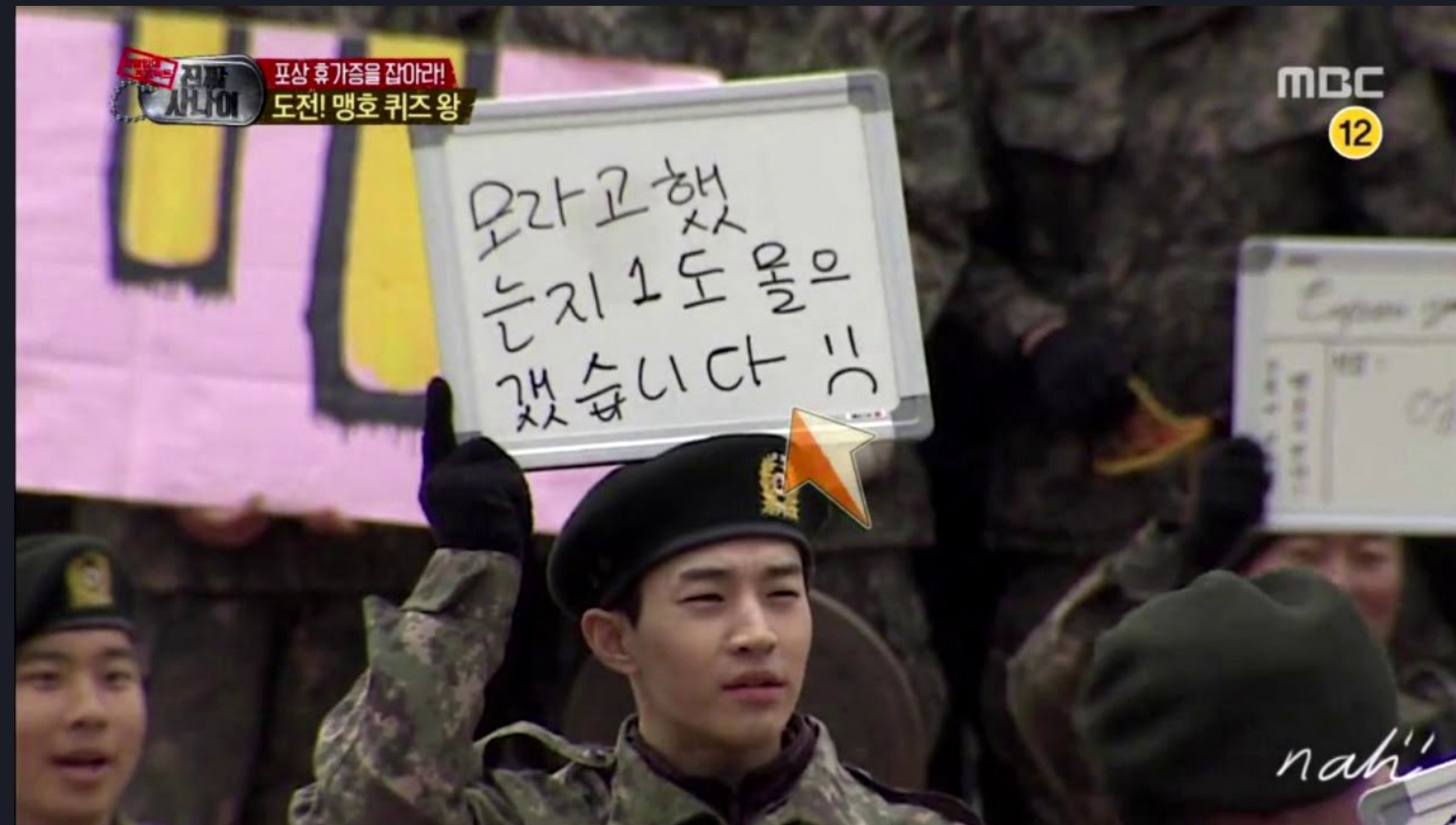
# 실행 컨텍스트란?

영어로는 **Execution Context**.

자바스크립트 엔진에서 프로그램을 동작하려 할 때,  
실행에 필요한 정보들을 정해진 '맥락'에 맞게 구분한 것

실행 가능한 코드가 되기 위해 필요한 환경

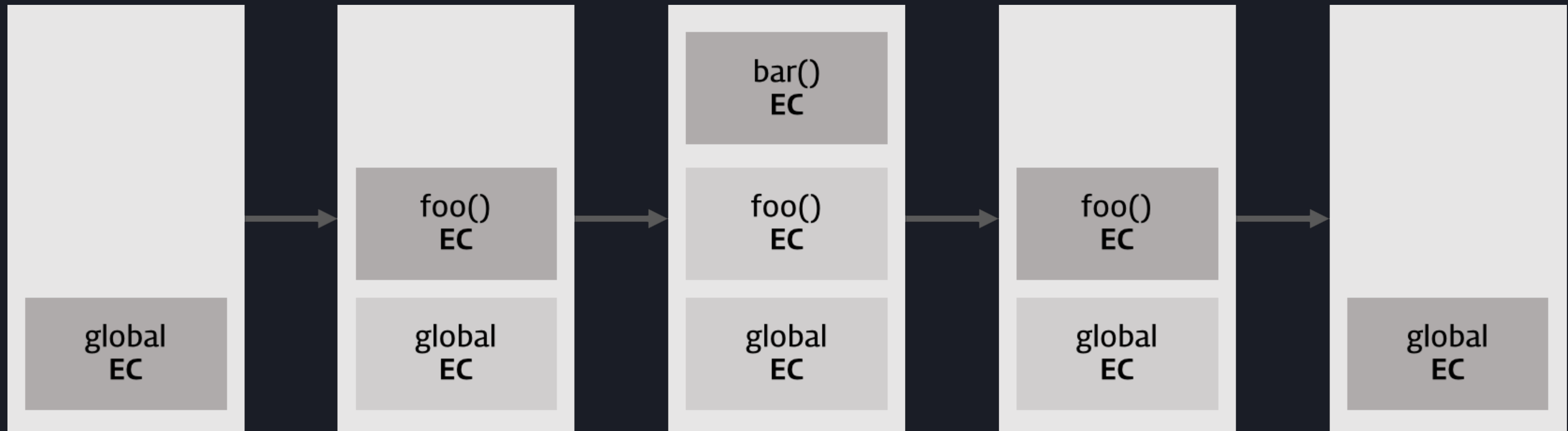
# 구분 ? 환경?



So, 자바스크립트 엔진은 실행 컨텍스트를 물리적 객체의 형태로 관리한다.

# 실행 컨텍스트 스택

프로그램이 시작하고 끝나는 동안 생성되고 소멸되는 실행 컨텍스트들 관리하는 스택  
LIFO (Last Input First Out) 으로 관리한다.



# 실행 컨텍스트 종류

Global Context

전역 컨텍스트



코드 전역을 아우르는 컨텍스트

프로그램의 모든 전역 변수, 전역 함수 등을 포함하는 환경  
프로그램이 종료될 때까지 유지

Functional Context

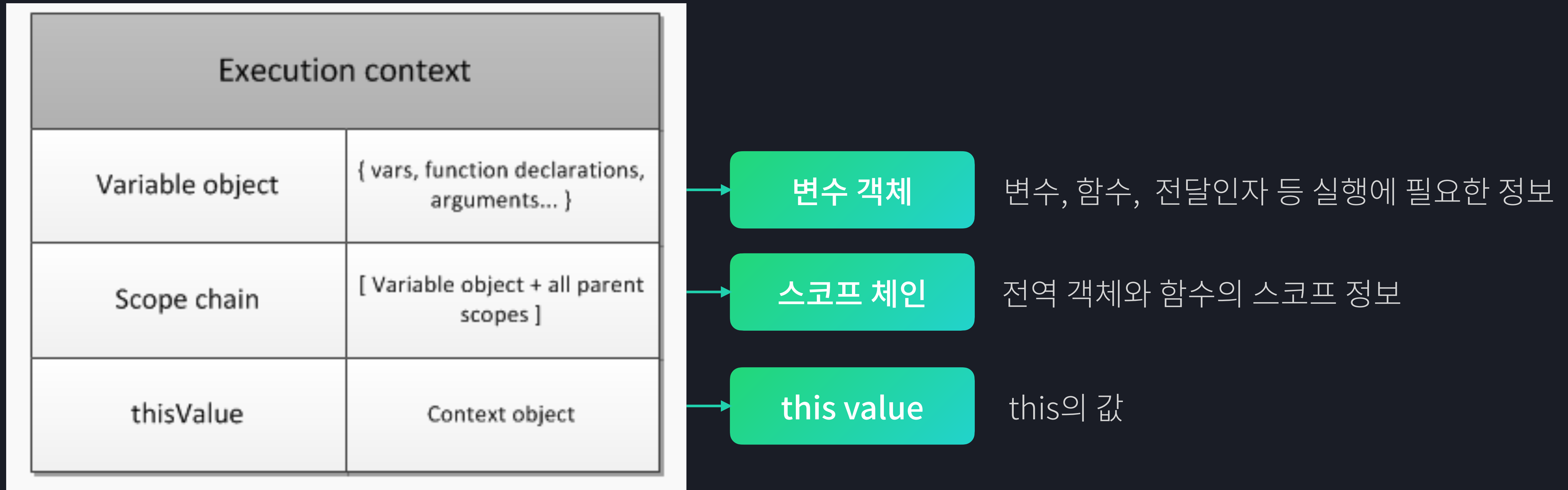
함수 컨텍스트



함수를 호출할 때 생기는 컨텍스트

특정 함수 내에서 사용하는 변수, 함수들의 정보 포함  
함수의 모든 동작이 종료되면 소멸 (클로저는 예외)

# 실행 컨텍스트 구성요소



# 이 코드로 한 번 보겠습니다.

```
1  var x = 'xxx';
2
3  function foo () {
4      var y = 'yyy';
5
6      function bar () {
7          var z = 'zzz';
8          console.log(x + y + z);
9      }
10     bar();
11 }
12
13 foo();
```

## 1. 프로그램 실행 시 전역 컨텍스트 생성

- 1.1 스코프 체인 생성
- 1.2 변수 객체 실행
- 1.2 this value 결정

## 2. 함수 호출 시 함수 컨텍스트 생성

- 2.1 스코프 체인 생성
- 2.2 변수 객체 실행
- 2.2 this value 결정

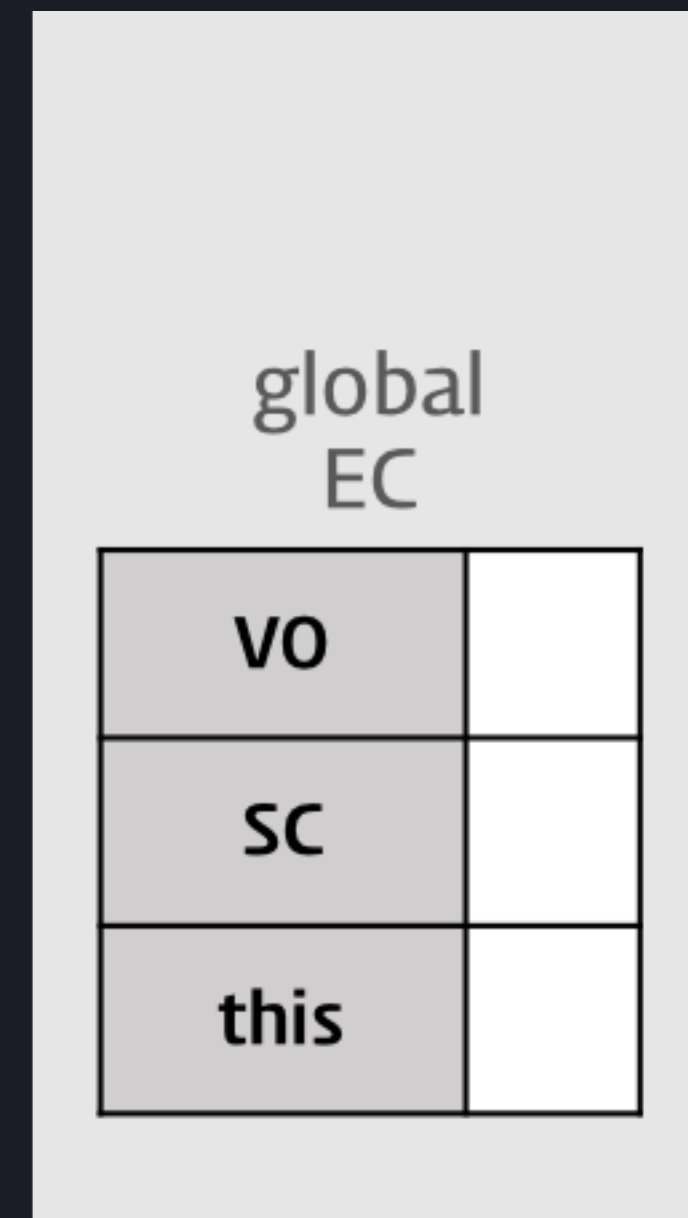
## 3. 함수 종료 시 컨텍스트 제거

## 4. 다른 함수 컨텍스트 생성/제거 반복....

## 5. 프로그램 종료 시 전역 컨텍스트 제거

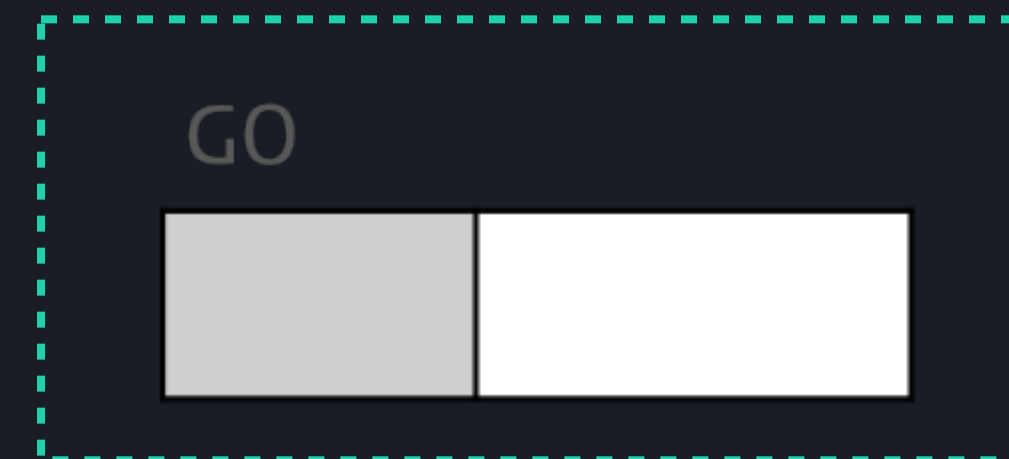
# 1. 전역 컨텍스트 생성

전역 객체가 생긴 뒤 전역 실행 컨텍스트가 생기고 스택에 쌓인다.



실행컨텍스트 스택

전역 객체 (Global Object)

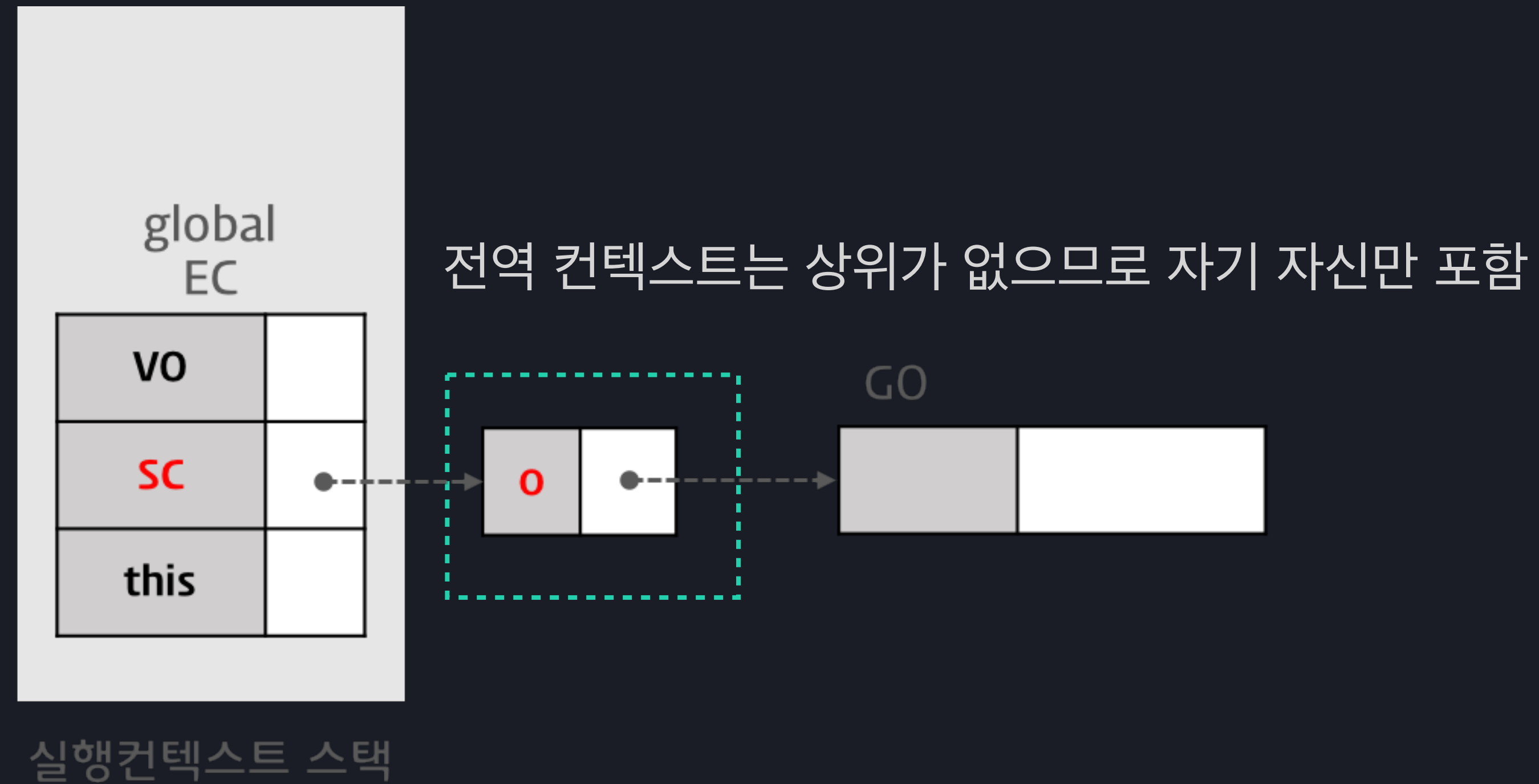


DOM, BOM, Built-in object 등이 설정되어 있다.



# 1-1. 스코프 체인 생성 및 초기화

자기 자신과 상위 스코프들의 변수 객체를 배열 형태로 추가한다.



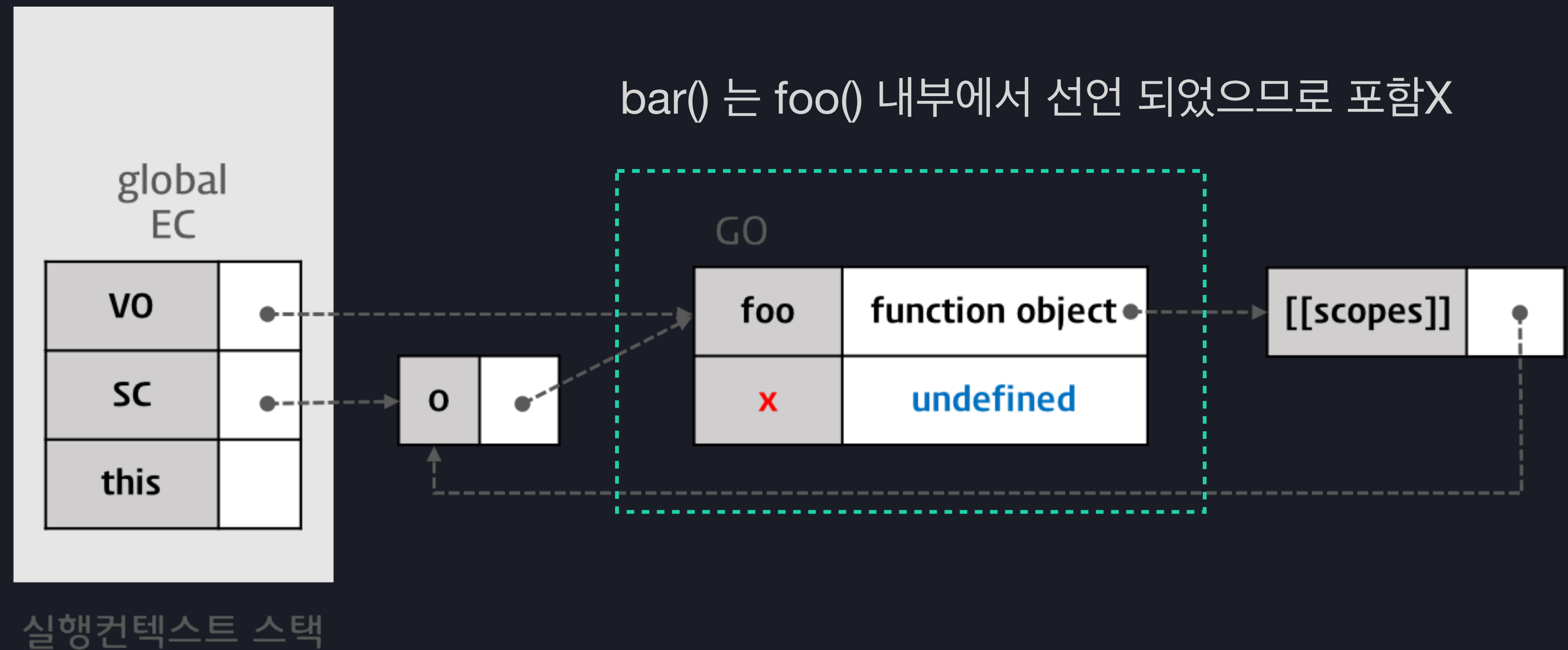
## 1-2. 변수 객체화 실행

현재 전역 범위에서 선언한 변수 또는 함수들의 정보를 추가한다.

```

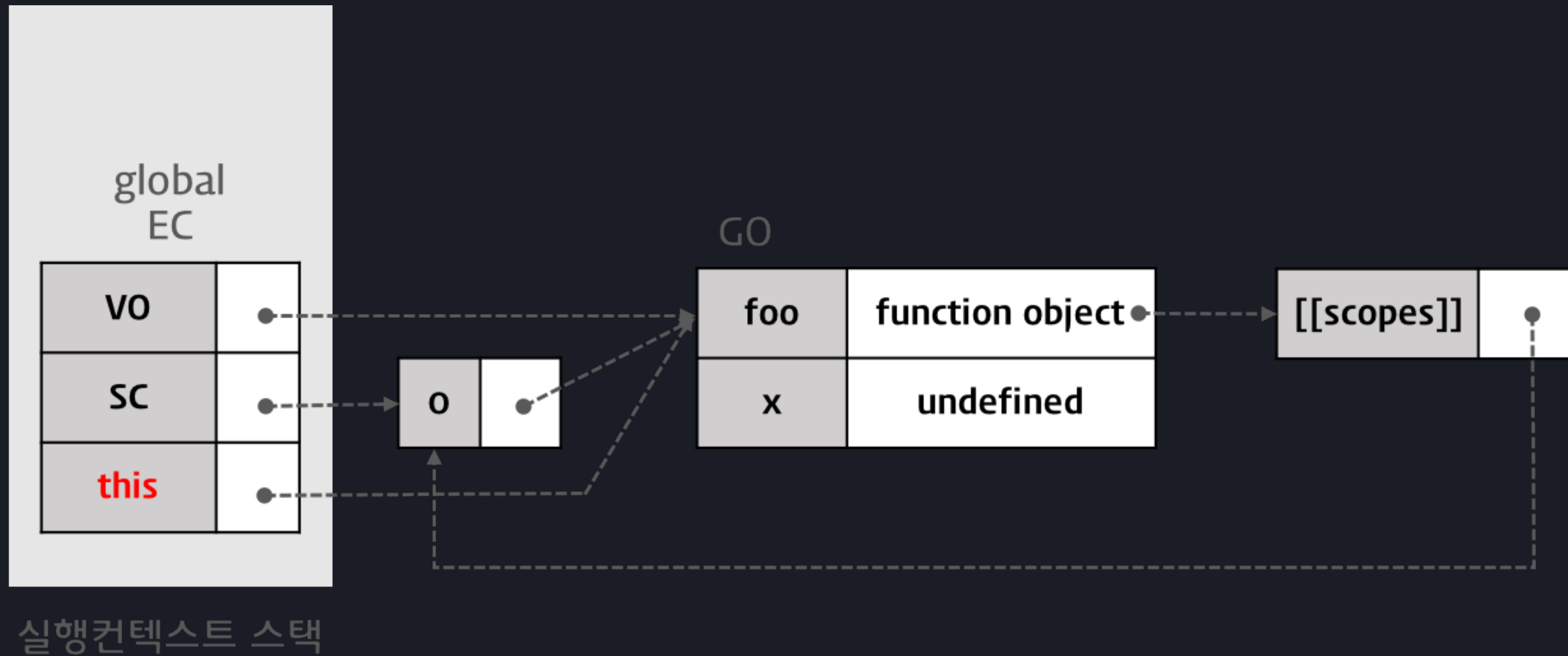
1  var x = 'xxx';
2
3  function foo () {
4      var y = 'yyy';
5
6      function bar () {
7          var z = 'zzz';
8          console.log(x + y + z);
9      }
10     bar();
11 }
12
13 foo();

```



## 1-3. this 값 결정

따로 변경하지 않는 한 window로 나온다.

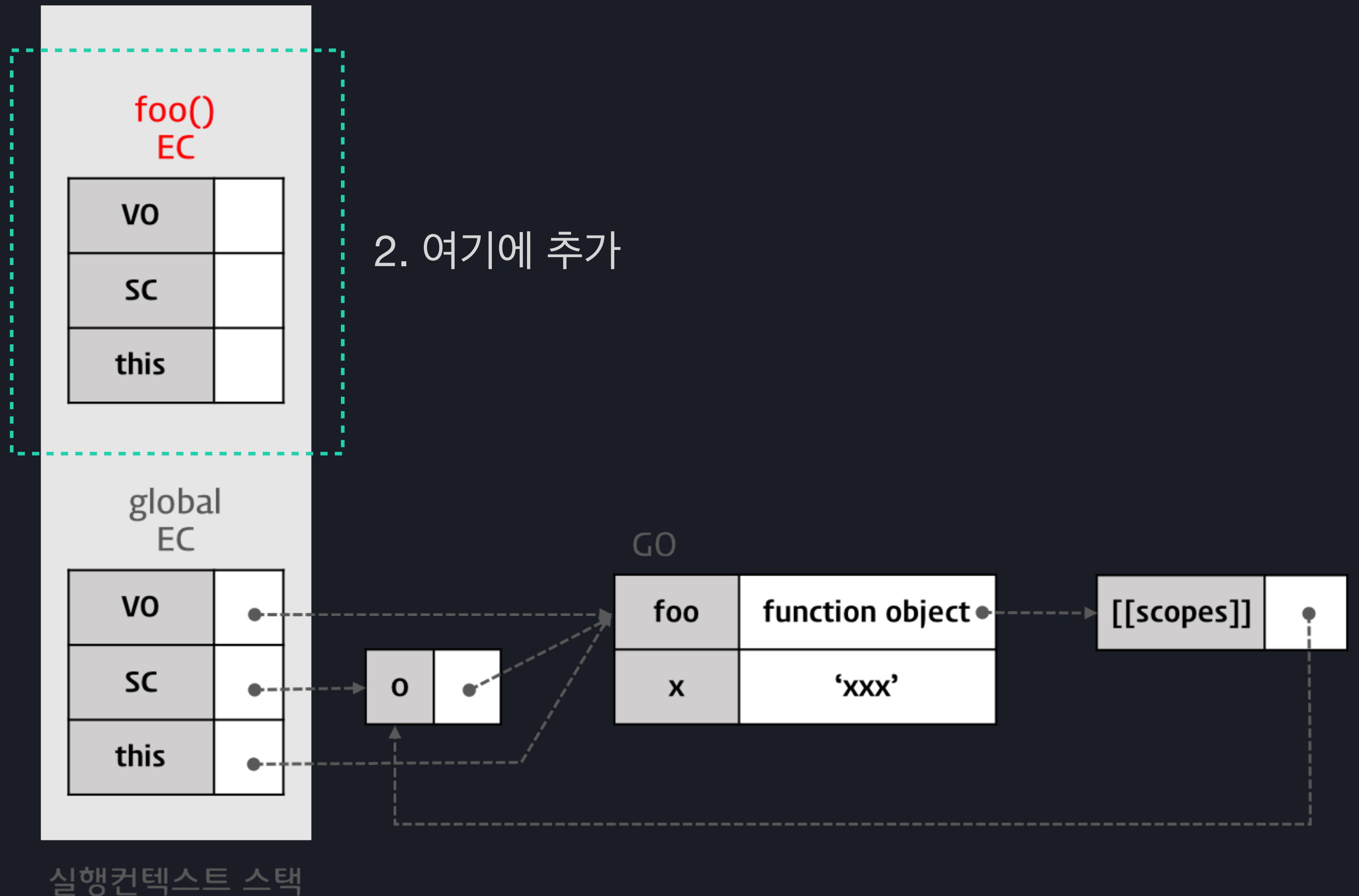


## 2. 함수 컨텍스트 생성

함수 호출 시 함수 컨텍스트가 생성되고 스택에 쌓인다.

```
Title
1  var x = 'xxx';
2
3  function foo () {
4    var y = 'yyy';
5
6    function bar () {
7      var z = 'zzz';
8      console.log(x + y + z);
9    }
10   bar();
11 }
12
13 foo();
```

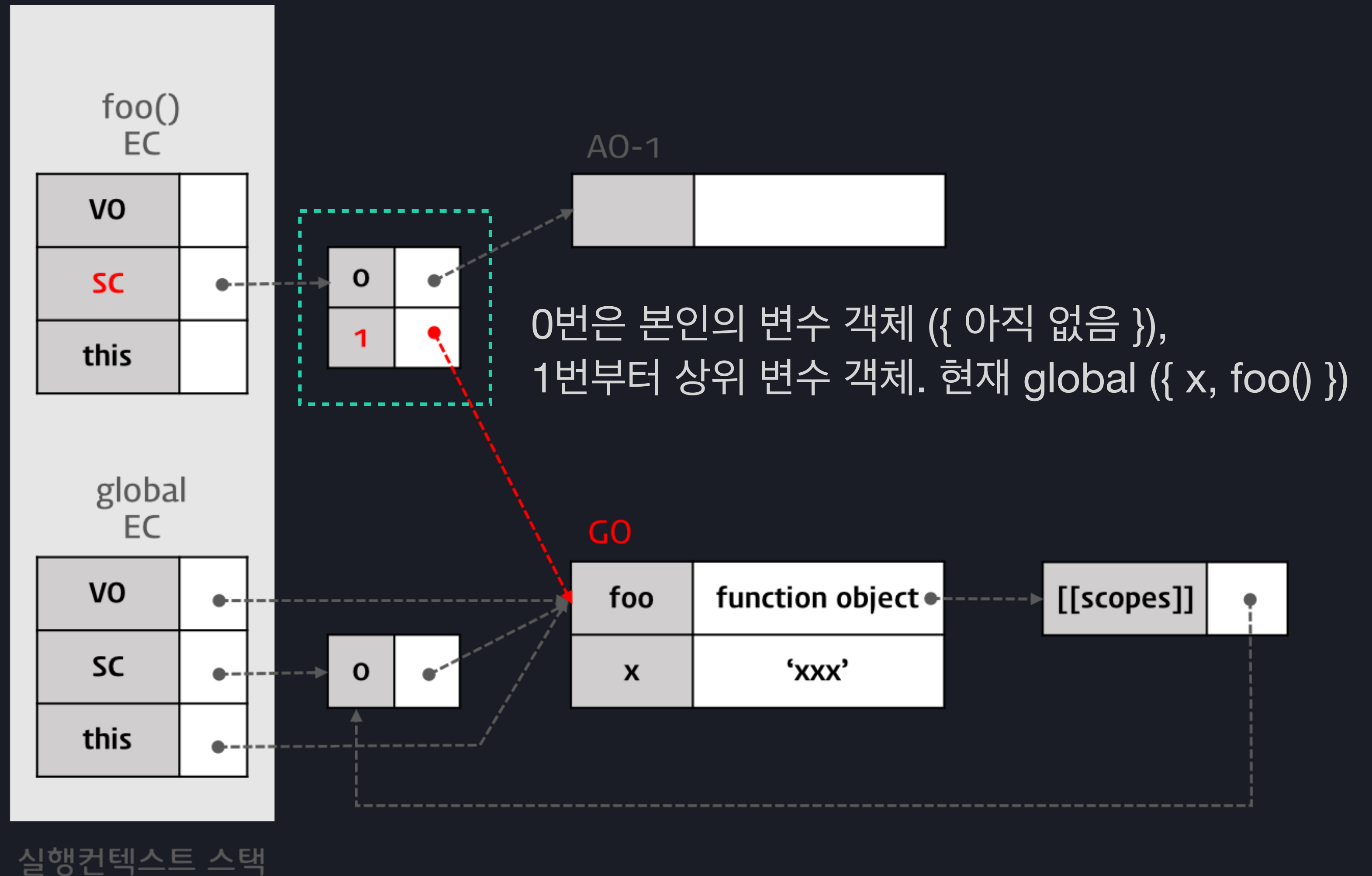
1. 이 시점에서 생성되고



## 2-1. 스코프 체인 생성 및 초기화

전역 컨텍스트와 방식 동일

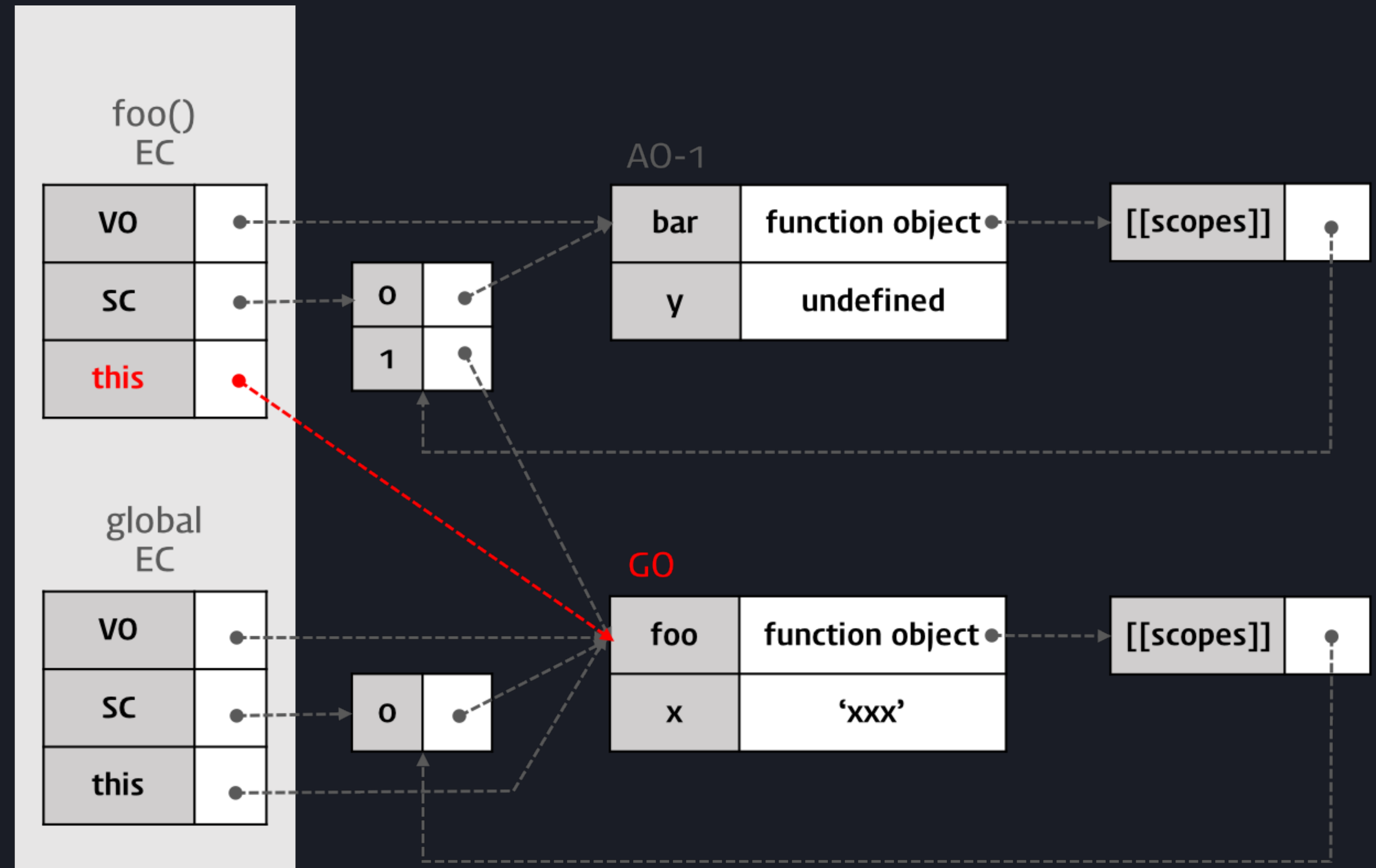
```
1  var x = 'xxx';
2
3  function foo () {
4    var y = 'yyy';
5
6    function bar () {
7      var z = 'zzz';
8      console.log(x + y + z);
9    }
10   bar();
11 }
12
13 foo();
```



## 2-2, 2.3 변수 객체화 실행, this 값 결정

전역 컨텍스트와 방식 동일

```
1  var x = 'xxx';
2
3  function foo () {
4    var y = 'yyy';
5
6    function bar () {
7      var z = 'zzz';
8      console.log(x + y + z);
9    }
10   bar();
11 }
12
13 foo();
```

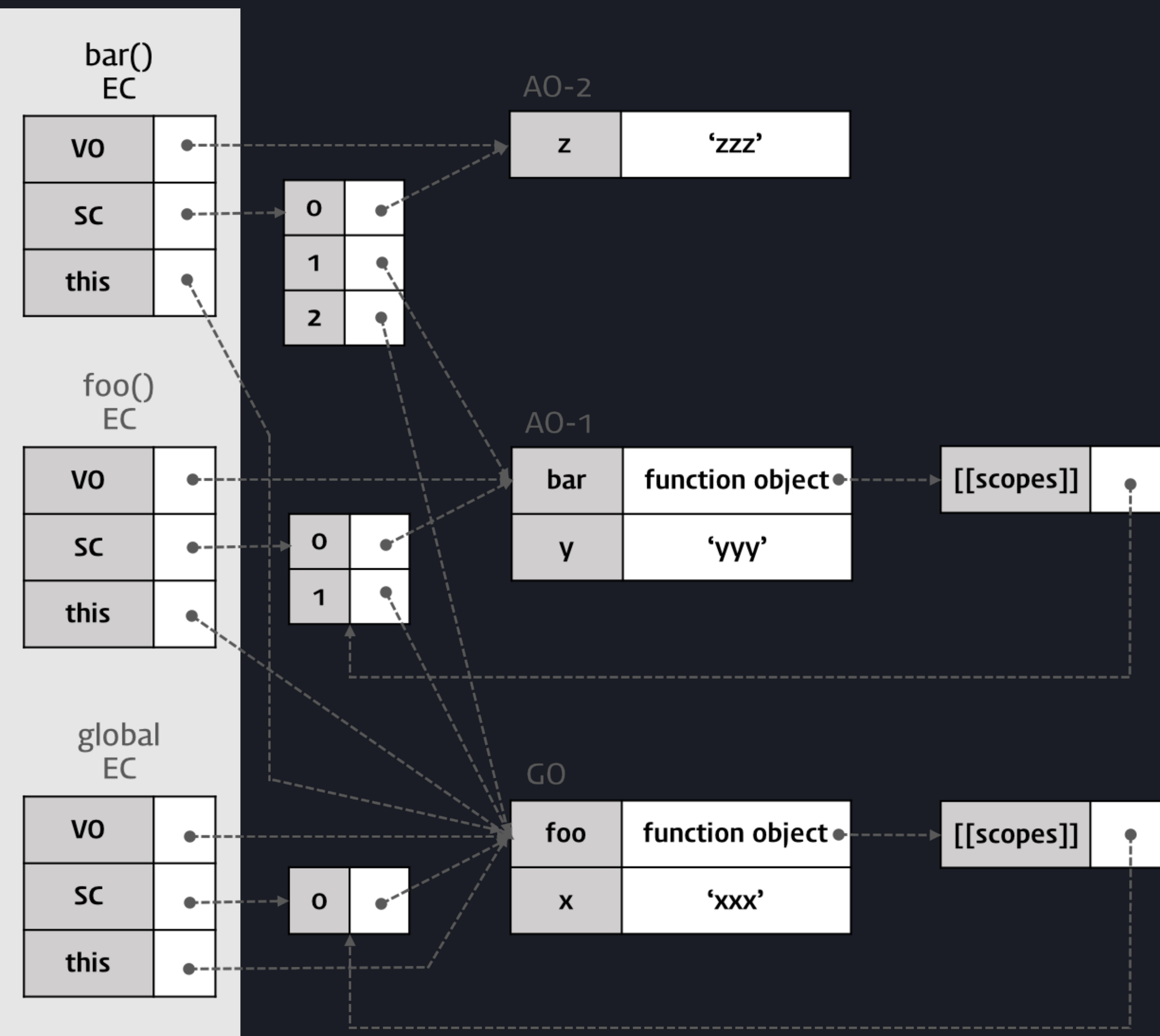


실행컨텍스트 스택

# 최종\_진짜 최종.ppt

```
1  var x = 'xxx';
2
3  function foo () {
4    var y = 'yyy';
5
6    function bar () {
7      var z = 'zzz';
8      console.log(x + y + z);
9    }
10   bar();
11 }
12
13 foo();
```

console.log = xxxyyyzzz



실행컨텍스트 스택

# tl:dr;

실행 컨텍스트란 자바스크립트 프로그램을 실행하기 위한 환경

실행 컨텍스트 스택에서 관리하며, 전역 컨텍스트와 함수 컨텍스트로 나뉜다.

각 컨텍스트마다 변수 객체, 스코프 체인, this값 으로 구성되어 있다.



감사합니다



# Reference

[Execution Context | PoiemaWeb]

<https://poiemaweb.com/js-execution-context>

[(JavaScript) 실행 컨텍스트 - 클로저와 호이스팅 - ZeroCho Blog]

<https://www.zerocho.com/category/JavaScript/post/5741d96d094da4986bc950a0>