
23장

실행 컨텍스트

23.1 소스코드의 타입

소스코드 타입 4가지

1. 전역 코드.
2. 함수 코드.
3. Eval 코드.
4. 모듈 코드.



소스코드의 타입에 따라

실행 컨텍스트 생성, 관리가 달라진다.

23.2 소스코드의 평가와 실행

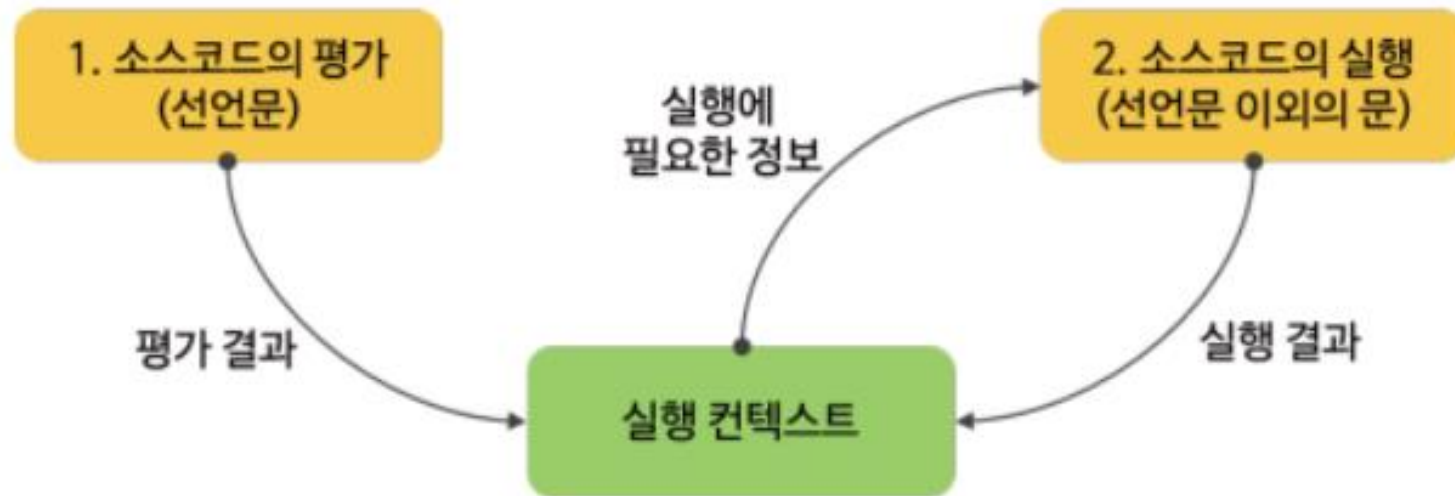


그림 23-2 소스코드의 평가와 실행

23.3 실행 컨텍스트의 역할

```
// 전역 변수 선언
const x = 1;
const y = 2;

// 함수 정의
function foo(a) {
  // 지역 변수 선언
  const x = 10;
  const y = 20;

  // 메서드 호출
  console.log(a + x + y); // 130
}

// 함수 호출
foo(100);

// 메서드 호출
console.log(x + y); // 3
```

23.3 실행 컨텍스트의 역할

1

전역 코드 평가

2

전역 코드 실행

3

함수 코드 평가

4

함수 코드 실행

- 코드가 실행되기 위해 식별자, 스코프, 코드실행순서 등 관리 필요.
- 식별자와 스코프는 실행 컨텍스트의 **렉시컬 환경**으로 관리.
- 코드실행 순서는 **실행 컨텍스트 스택**으로 관리.

23.4 실행 컨텍스트 스택

```
const x = 1;

function foo () {
  const y = 2;

  function bar () {
    const z = 3;
    console.log(x + y + z);
  }
  bar();
}

foo(); // 6
```

23.4 실행 컨텍스트 스택



- 실행 컨텍스트 최상위 스택은 **현재 실행중인 코드**의 실행 콘텍스트.

23.5 렉시컬 환경

- 렉시컬 환경은 식별자와 값, 상위 스코프에 대한 참조를 기록.
- 환경 레코드는 스코프에 포함된 **식별자와 그 값**을 관리.
- 외부 렉시컬 환경에 대한 참조는 **상위 스코프**를 가르킴.

23.5 렉시컬 환경

실행 컨텍스트

렉시컬 환경

실행 컨텍스트 스택

환경 레코드

외부 렉시컬 환경에 대한 참조

23.6 실행 컨텍스트의 생성과 식별자 검색 과정

```
var x = 1;
const y = 2;

function foo (a) {
  var x = 3;
  const y = 4;

  function bar (b) {
    const z = 5;
    console.log(a + b + x + y + z);
  }
  bar(10);
}

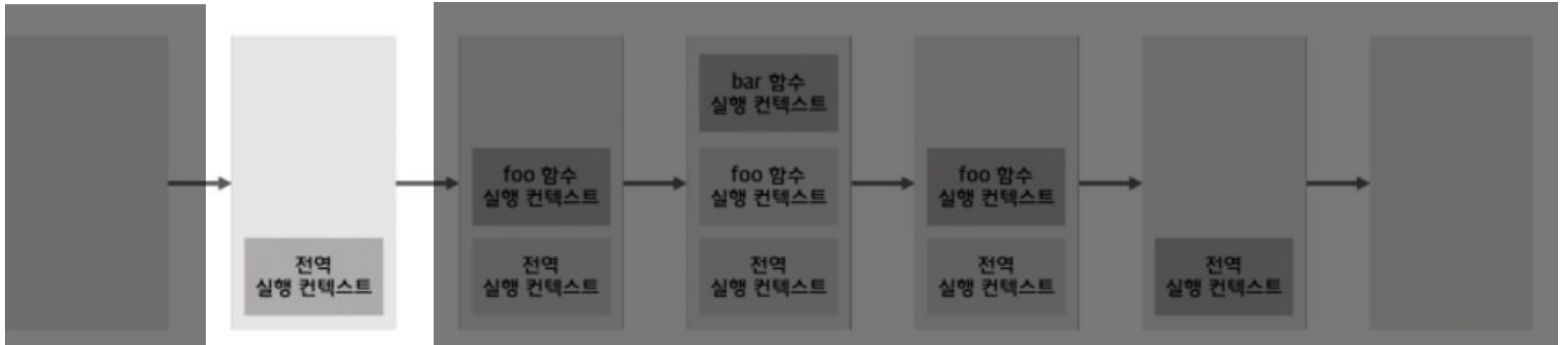
foo(20); // 42
```

23.6 실행 컨텍스트의 생성과 식별자 검색 과정

1. 전역 객체 생성

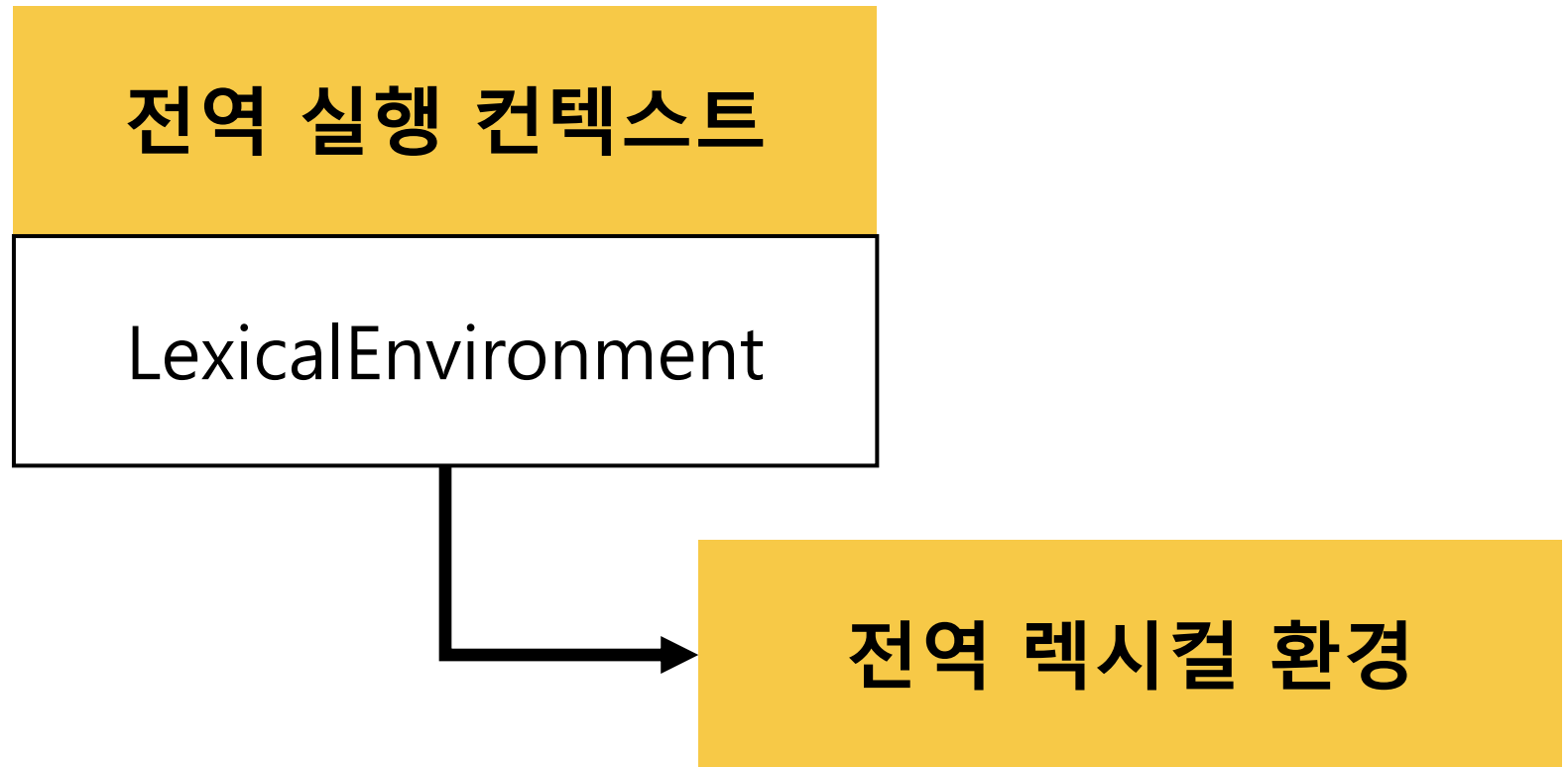
2. 전역 코드 평가

전역 실행 컨텍스트



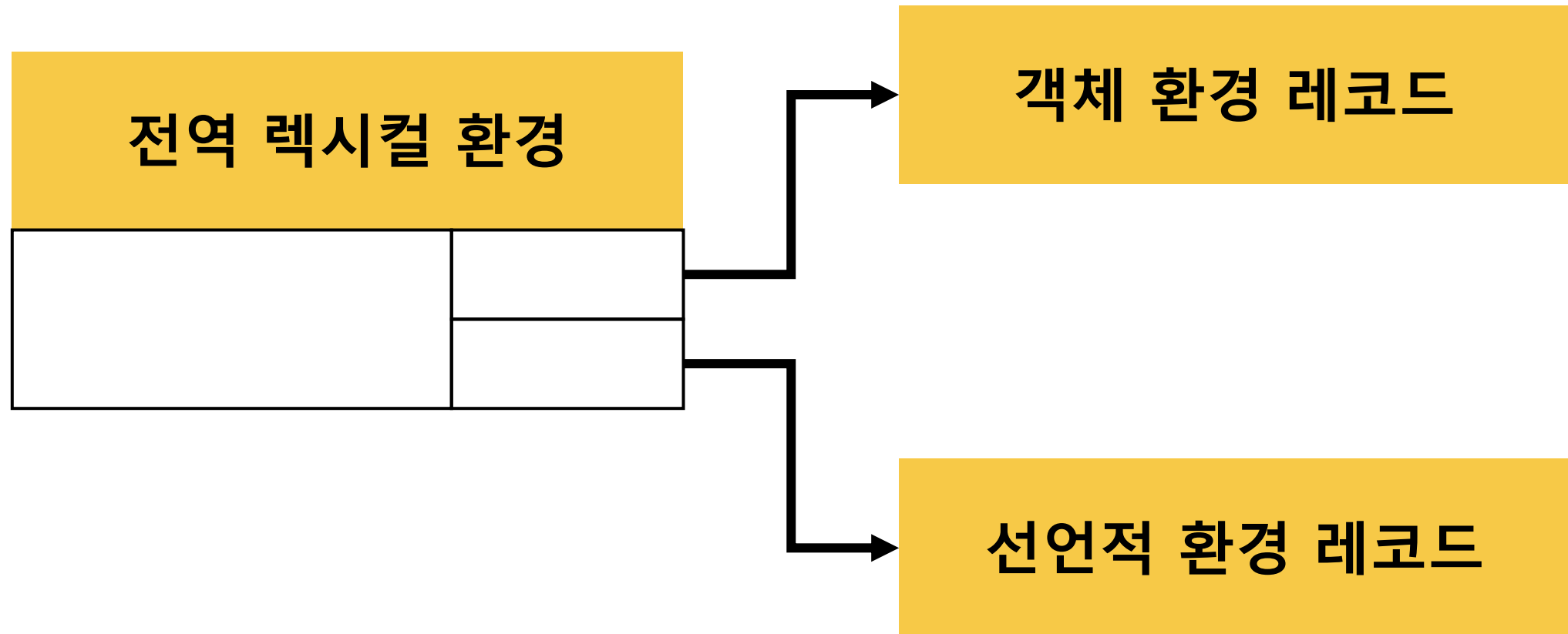
23.6 실행 컨텍스트의 생성과 식별자 검색 과정

2. 전역 코드 평가



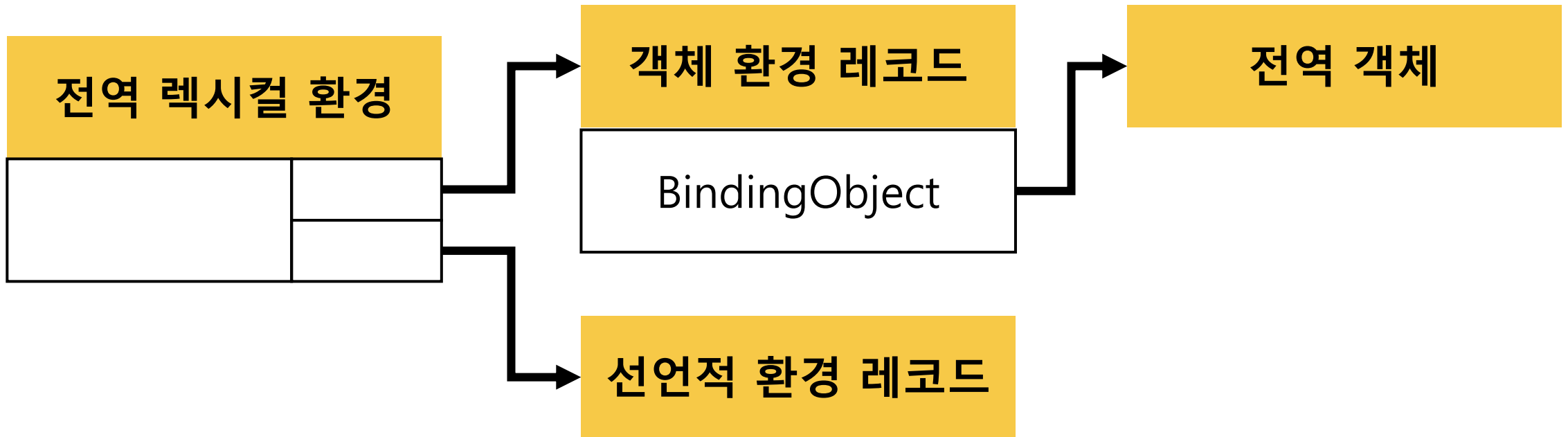
23.6 실행 컨텍스트의 생성과 식별자 검색 과정

2. 전역 코드 평가



23.6 실행 컨텍스트의 생성과 식별자 검색 과정

2. 전역 코드 평가



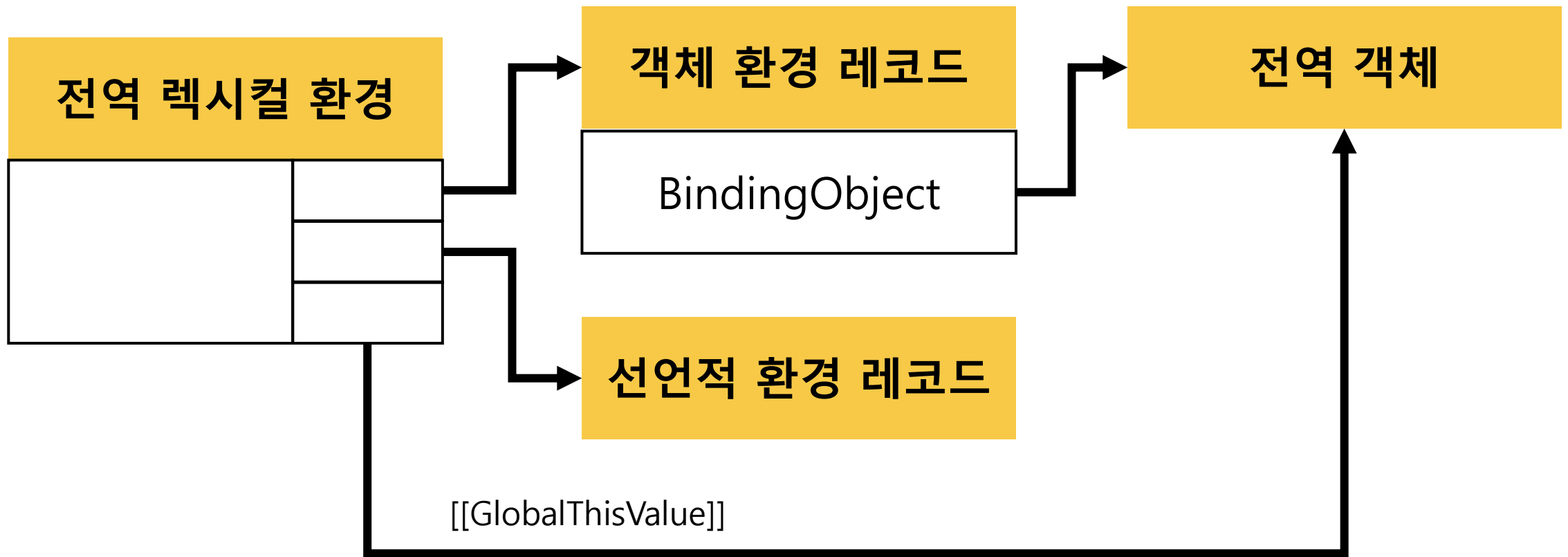
23.6 실행 컨텍스트의 생성과 식별자 검색 과정

2. 전역 코드 평가

- var는 전역객체에 변수 식별자를 등록하는 동시에 undefined를 바인딩.
- 함수는 함수식별자에 함수객체를 즉시 할당.
- let , const 변수는 선언 단계와 초기화 단계가 구분되어 진행.

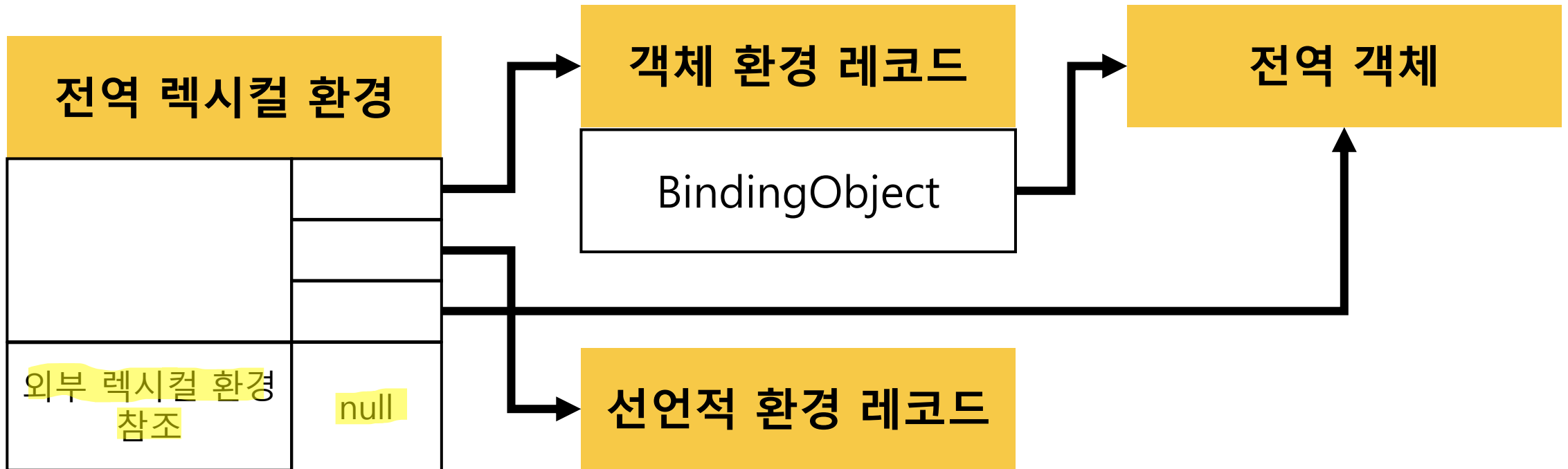
23.6 실행 컨텍스트의 생성과 식별자 검색 과정

2. 전역 코드 평가



23.6 실행 컨텍스트의 생성과 식별자 검색 과정

2. 전역 코드 평가



23.6 실행 컨텍스트의 생성과 식별자 검색 과정

2. 전역 코드 평가

1

실행 컨텍스트
생성

2

렉시컬 환경
생성

3

환경 레코드
생성

4

This 바인딩

5

외부 렉시컬
환경에 대한
참조 결정

23.6 실행 컨텍스트의 생성과 식별자 검색 과정

3. 전역 코드 실행

- **현재 실행 중인 컨텍스트**에서 부터 식별자 검색 시작.
- 실행 중인 컨텍스트에 식별자 없으면 **외부 렉시컬 환경 참조**가 가르키는 상위 스코프로 이동하여 검색.

23.6 실행 컨텍스트의 생성과 식별자 검색 과정

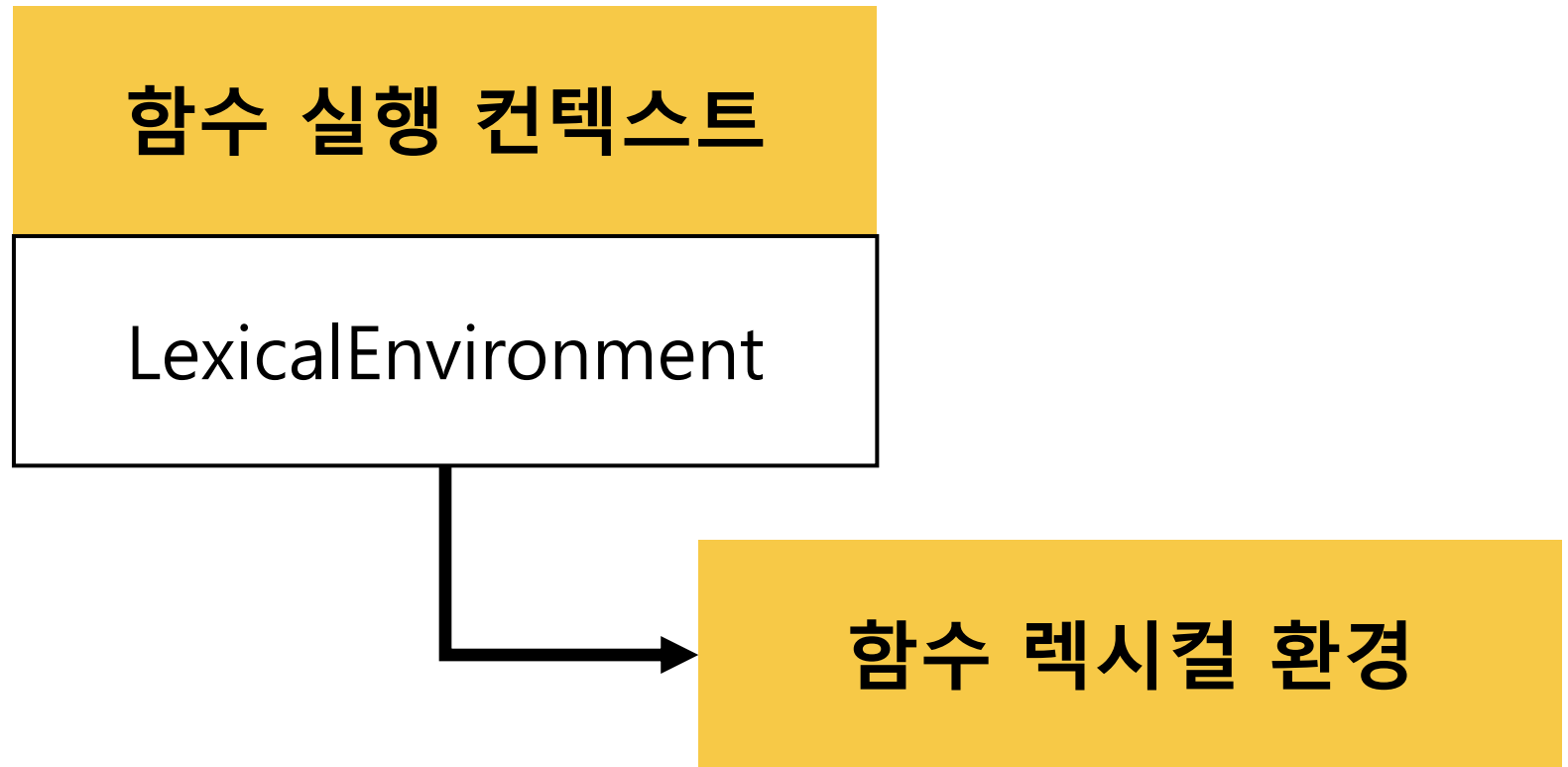
4. foo 함수 코드 평가

함수 실행 컨텍스트



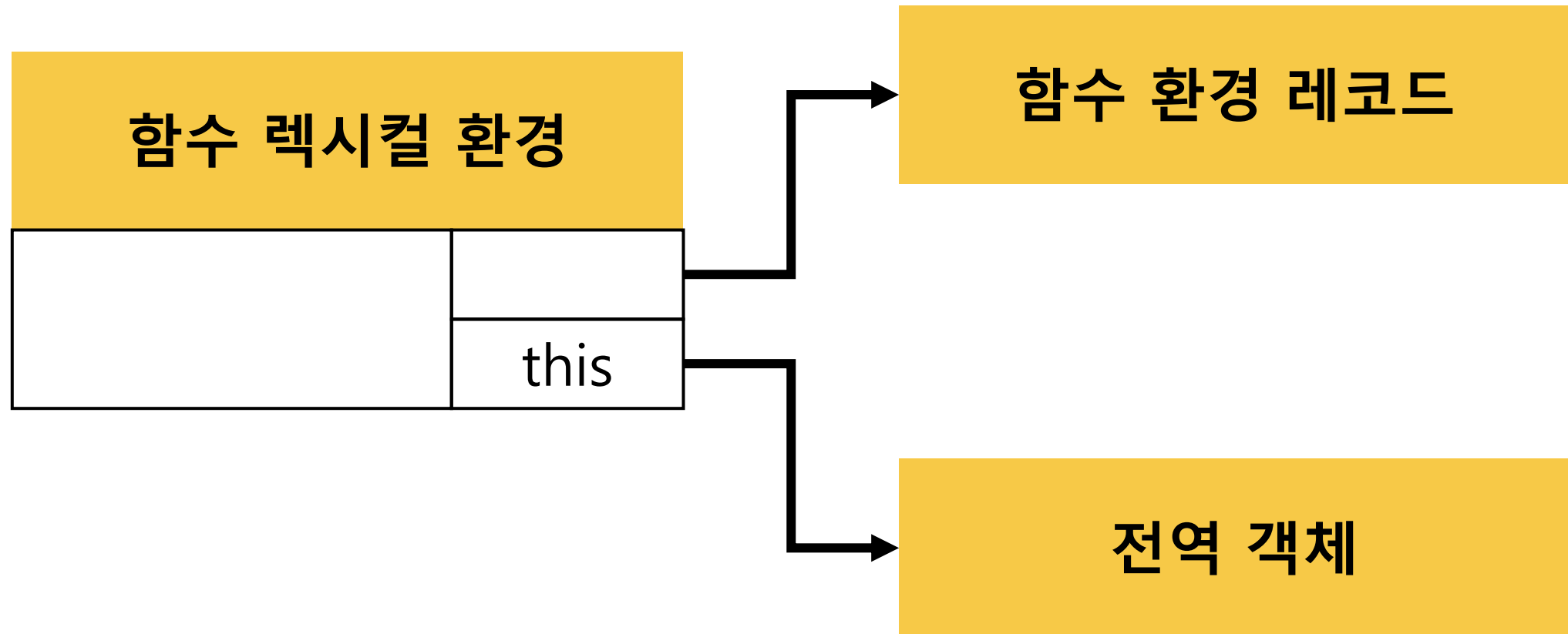
23.6 실행 컨텍스트의 생성과 식별자 검색 과정

4. foo 함수 코드 평가



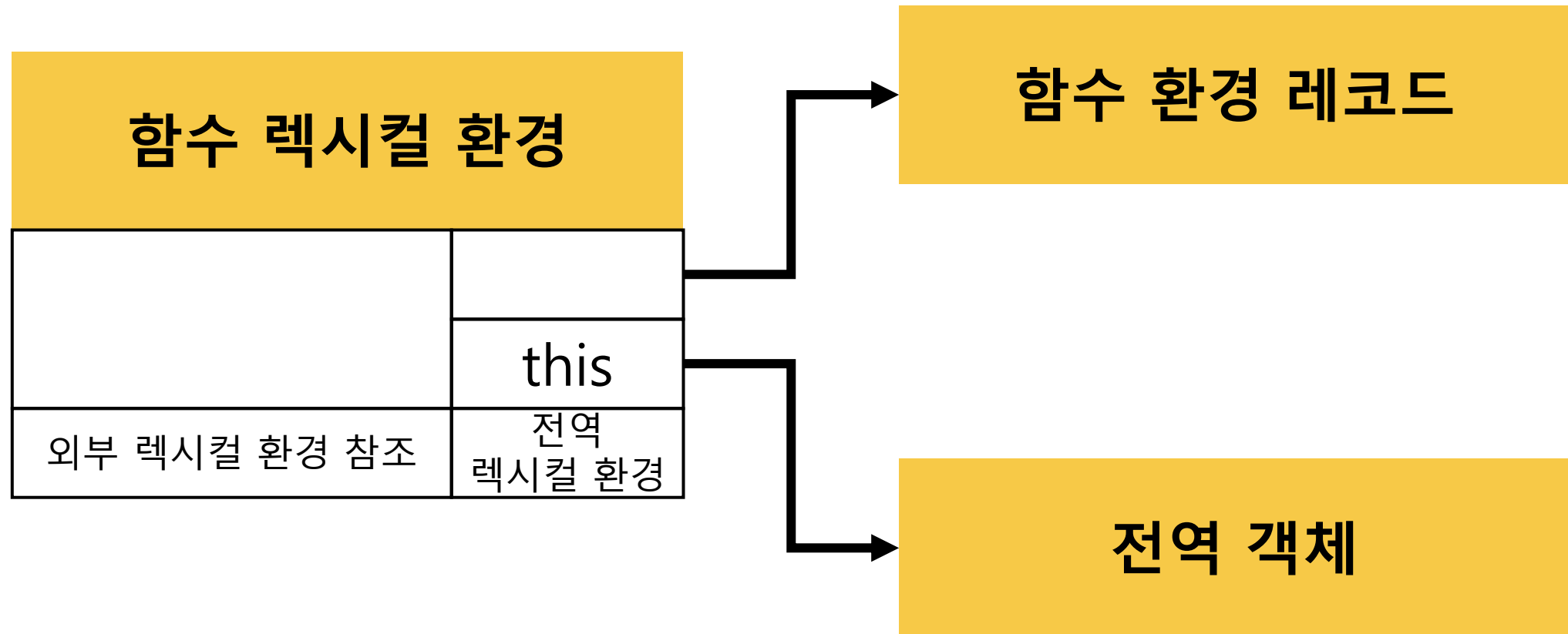
23.6 실행 컨텍스트의 생성과 식별자 검색 과정

4. foo 함수 코드 평가



23.6 실행 컨텍스트의 생성과 식별자 검색 과정

4. foo 함수 코드 평가



23.6 실행 컨텍스트의 생성과 식별자 검색 과정

5. foo 함수 코드 실행

- **현재 실행 중인 컨텍스트**에서 부터 식별자 검색 시작.
- 실행 중인 컨텍스트에 식별자 없으면 **외부 렉시컬 환경 참조**가 가르키는 상위 스코프로 이동하여 검색.

23.6 실행 컨텍스트의 생성과 식별자 검색 과정

6. bar 함수 코드 평가

7. bar 함수 코드 실행

- **현재 실행 중인 컨텍스트**에서 부터 식별자 검색 시작.
- 실행 중인 컨텍스트에 식별자 없으면 **외부 렉시컬 환경 참조**가 가르키는 상위 스코프로 이동하여 검색.

23.6 실행 컨텍스트의 생성과 식별자 검색 과정

8. bar 함수 코드 실행 종료

9. foo 함수 코드 실행 종료

10. 전역 코드 실행 종료

- 실행 컨텍스트 스택에서 하나씩 제거.
- 렉시컬 환경이 참조되고 있지 않다면 메모리 공간 확보 해제.