

13장 스코프

스코프란?

- 스코프(유효범위) = 식별자가 유효한 범위
- 식별자 = 변수, 함수, 클래스 등의 이름을 의미
- 즉, 어떤 함수, 변수 혹은 클래스의 이름을 참조할 수 있는 범위를 의미한다

식별자 결정



```
var x = 'global';

function foo() {
  var x = 'local';
  console.log(x);
}

foo(); // 1

console.log(x); // 2
```

- 자바스크립트 엔진은 스코프를 통해 어떤 변수를 참조해야 할 것인지를 결정한다.
- 식별자를 검색할 때 사용하는 규칙을 의미한다.
- 전역에 선언된 x와 foo에 선언된 x는 이름이 동일한 식별자이지만 스코프가 다른 별개의 변수다.

스코프의 종류

- 전역 스코프
 - 코드 가장 바깥 영역을 말한다.
 - 전역에 선언된 식별자는 어디서든지 참조할 수 있다
- 지역 스코프
 - 지역이란 함수 몸체 내부를 의미한다
 - 지역 변수는 자신의 지역 스코프와 하위 지역 스코프에서 유효하다

스코프 체인

```
var x = 'outer';
var y = 'outer';

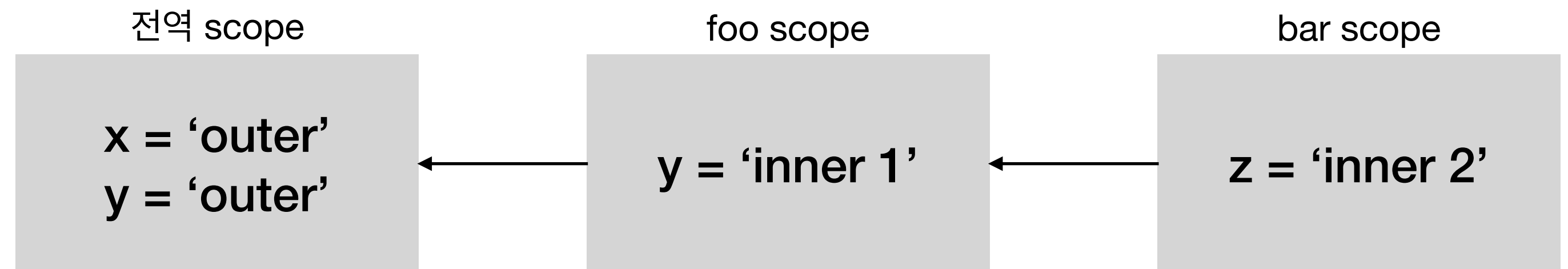
function foo() {
  var y = 'inner 1';

  function bar() {
    var z = 'inner 2';

    console.log(x); // 1
    console.log(y); // 2
    console.log(z); // 3
  }
}

foo();
```

- 자바스크립트는 함수 안에 함수를 정의할 수 있다(중첩 함수).
- 이처럼 함수의 지역 스코프도 중첩될 수 있다. => 스코프가 계층적 구조를 갖는다.
- 스코프가 계층적으로 연결된 것을 스코프 체인이라고 하며, 변수를 검색할 때(식별자 결정)할때, 가장 아래에 있는 스코프에서 시작하여 선언된 변수를 찾는다.



함수 레벨 스코프



```
var i = 10;

for(var i=0; i< 5; i++) {
    console.log(i);
}

console.log(i); // ?
```

- C나 자바 등을 비롯한 대부분의 프로그래밍 언어는 함수 몸체만이 아니라 모든 코드 블록 (if, for, while, try/catch)이 지역 스코프를 만든다. => 블록 레벨 스코프라 한다
- 자바스크립트에서 var 키워드로 선언된 변수는 오로지 함수의 코드블록만을 지역 스코프로 인정한다. => 이를 함수 레벨 스코프라 한다.
- ES6에서 도입된 let, const는 블록 레벨 스코프를 지원한다

렉시컬 스코프



```
var x = 1;
```

```
function foo() {  
  var x = 10;  
  bar();  
}
```

```
function bar() {  
  console.log(x);  
}
```

```
foo(); // ?  
bar(); // ?
```

- 렉시컬 스코프(정적 스코프)는 정의가 평가되는 시점에 상위 스코프가 정적으로 결정된다.
- 자바스크립트를 비롯한 대부분의 프로그래밍 언어는 렉시컬 스코프를 따른다.
- 즉, 자바스크립트는 함수를 어디서 호출했는지가 아니라 함수를 어디서 정의했는지에 따라 상위 스코프를 결정한다.
- 이와 반대되는 개념으로는, 동적 스코프가 있다. 이는 함수가 호출되는 시점에 상위 스코프가 결정된다.