

JS-중급 10

☰ 태그	
📅 날짜	@2023년 6월 2일

클로저 (Closure)

```
let one;
one = 1;

// 전역 Lexical 환경
// one : 초기화 x 사용불가
// addOne : function 사용가능

function addOne(num) {
  console.log(one + num);
  // one과 num을 내부Lexical환경에서 먼저 찾고
  // 없으면 전역Lexical 환경에서 찾음
}

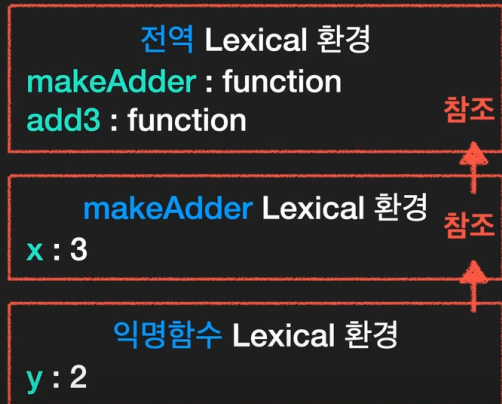
addOne(5);

// 내부 Lexical환경 (전역 Lexical환경을 참조)
// num : 5
```

어휘적 환경(Lexical Environment)

```
function makeAdder(x){  
  return function(y){  
    return x + y;  
  }  
}
```

```
const add3 = makeAdder(3);  
console.log(add3(2));
```



어휘적 환경 (Lexical Environment)

```
function makeAdder(x) {  
  return function(y) {  
    //y를 가지고 있고 상위 함수인 makeAdder의 x에 접근 가능  
    return x + y;  
  }  
}
```

```
const add3 = makeAdder(3);  
console.log(add3(2)); //5  
// add3 함수가 생성된 이후에도 상위함수인 makeAdder의  
// x에 접근 가능
```

```
const add10 = makeAdder(10);  
console.log(add10(5)); // 15  
console.log(add3(1)); // 4
```

add10과 add3은 서로 다른 환경이다.

이런걸 Closure라고 한다

- 함수와 렉시컬 환경의 조합
- 함수가 생성될 당시의 외부 변수를 기억
- 생성 이후에도 계속 접근 가능

```
function makeCounter() {  
  let num = 0;  
  
  return function(){  
    return num++;  
  };  
}  
  
let counter = makeCounter();  
  
console.log(counter());  
console.log(counter());  
console.log(counter());  
  
0  
1  
2
```