

Call by Reference, Call by Value

< 목차 >

Call by Reference

Call by Reference란?

예제 코드

단점 보완

Call by Value

Call by Value란?

예제 코드

Call by Reference

Call by Reference란?

: 함수 호출 시, 인수로 전달되는 변수의 참조 값을 함수 내부로 전달하는 방식

- 전달된 변수의 값을 변경하면, 호출한 쪽에서도 해당 변수의 값이 저장된다.
 - 인자로 전달되는 값이 변수의 주소이다.
 - 함수 내에서 변수의 값을 변경하면 해당 주소에 저장된 값이 변경된다.
- 장점 : 복사하지 않고 직접 참조를 하기에 속도가 빠르다.
- 단점 : 직접 참조를 하기에 원래 값에 영향을 받는다. (리스크 존재)

예제 코드

```
function addOne(arr) {  
  arr.push(1);  
  return arr;  
}
```

```

}

let nums = [1, 2, 3];
let result = addOne(nums);

console.log(result);      // [1, 2, 3, 1]
console.log(nums);       // [1, 2, 3, 1]

```

⇒ addOne 함수에 nums 변수의 참조 값을 전달하면, arr 변수에도 nums 변수의 참조값이 복사

⇒ 함수 내부에서 arr 값을 변경하면 원본 nums 값도 함께 변경

단점 보완

원래 값이 영향을 받는 단점을 보완하려면 어떻게 해야 할까?

- 깊은 복사(deep copy)를 활용한다.
 - 메모리를 소모하게 되고, 빠른 속도로 작동한다는 장점을 잃을 수 있음
 - 깊은 복사와 Call by Reference를 함께 사용할 때는 상황에 따라 적절히 판단할 것
 - 원본 변경을 해야하는 인수라면 그대로 동작
 - 불변성을 지켜야 한다면 깊은 복사 사용

Call by Value

Call by Value란?

: 함수가 인수로 전달받은 값을 복사하여 처리하는 방식

- 전달된 값이 기본형(원시 타입) 데이터인 경우에만 해당한다.

- 변수가 가진 값을 복사하여 전달하므로 함수 내에서 값을 변경해도 원본 값은 변하지 않는다.
- 값의 불변성을 유지하는 데 용이하다.
- 장점 : 복사하여 처리하기 때문에 원래의 값이 안전하게 보존된다.
- 단점 : 복사하여 처리하기 때문에 메모리 사용량이 늘어난다.

예제 코드

```
function addOne(x) {
  x += 1;
  return x;
}

let num = 5;

console.log(addOne(num)); // 6
console.log(num);        // 5
```

⇒ num 변수가 가진 값을 addOne() 함수에 인수로 전달하면, x라는 새로운 변수에 값이 복사된다.

⇒ addOne() 함수 내에서 x 값을 변경하여도 원본 num 값은 변경되지 않는다.