

이터레이터 패턴

2012951014 권영근

이터레이터 패턴이란?

집합체(데이터 군)들의 내부표현 구조들을 노출시키지 않고 집합체의 원소들을 순회하며 접근하는 패턴이다.

Provide a way to access the elements of an aggregate object sequentially without exposing its underlying representation

-Gang of Four-

집합체(컬렉션)의 모든 항목에 일일이 접근하는 작업을 집합체(컬렉션 객체)가 아닌 반복자(iterator) 객체에서 맡게 된다.

자바스크립트의 `for..of` 구문을 예로 들어

```
/* Arr ----- */
const arr = [1,2,3];
for(const a of arr) { console.log(a); } // 1, 2, 3

/* Set ----- */
const set = new Set([1, 2, 3];
for(const a of set) { console.log(a); } // 1, 2, 3

/* Map ----- */
const map = new Map([[ 'a', 1], [ 'b', 2], [ 'c', 3]]);
for(const a of map) { console.log(a) } // ["a", 1], ["b", 2]...
```

눈에 보이지 않지만 내부적으로 iterator를 사용하고 있다.

배열의 이터레이터 객체 구현

```
function createArrayIterator(items) {  
  var i = 0;  
  return {  
    next: function() {  
      var done = (i >= items.length);  
      var value = !done ? items[i++] : undefined;  
      return {  
        done: done,  
        value: value  
      };  
    }  
  };  
}  
  
var iterator = createArrayIterator([1, 2, 3]);  
console.log(iterator.next()); // "{ value: 1, done: false }"  
console.log(iterator.next()); // "{ value: 2, done: false }"  
console.log(iterator.next()); // "{ value: 3, done: false }"  
console.log(iterator.next()); // "{ value: undefined, done: true }"  
console.log(iterator.next()); // "{ value: undefined, done: true }"
```

출처

위키피디아:

https://en.wikipedia.org/wiki/Iterator_pattern#JavaScript

함수형 자바스크립트, 유인동