

Ch36 디스트럭처링 할당

구조화된 배열과 같은 이터러블 또는 객체를 destructuring하여 1개 이상의 변수에 개별적으로 할당하는 것을 **디스트럭처링 할당**이라고 한다.

36.1 배열 디스트럭처링 할당

ES6의 **배열 디스트럭처링 할당**은 배열의 각 요소를 배열로부터 추출하여 1개 이상의 변수에 할당한다. 이때 **배열 디스트럭처링 할당의 대상(할당문의 우변)**은 **이터러블** 이어야 하며, **할당 기준은 배열의 인덱스** 다. (순서대로 할당됨)

```
const arr = [1, 2, 3];

// ES6 배열 디스트럭처링 할당
// 변수 one, two, three를 선언하고 배열 arr을 디스트럭처링하여 할당한다.
// 이때 할당 기준은 배열의 인덱스다.
const [one, two, three] = arr;

console.log(one, two, three); // 1 2 3
```

배열 디스트럭처링 할당을 위한 변수에 Rest 파라미터와 유사하게 **Rest 요소 ...** 을 사용할 수 있고 **Rest 요소**는 Rest 파라미터와 마찬가지로 **반드시 마지막에 위치**해야 한다.

```
//Rest 요소
const [x, ...y] = [1, 2, 3];
console.log(x, y); // 1 [ 2, 3 ]
```

36.2 객체 디스트럭처링 할당

ES6의 **객체 디스트럭처링 할당**은 객체의 각 프로퍼티를 객체로부터 추출하여 1개 이상의 변수에 할당한다. 이때 객체 디스트럭처링 할당의 대상은 객체이어야 하며, **할당 기준은 프로퍼티 키** 다. 즉, 순서는 의미가 없으며 선언된 변수 이름과 프로퍼티 키가 일치하면 할당된다.

```
const user = { firstName: 'Ungmo', lastName: 'Lee' };

// 프로퍼티 키를 기준으로 디스트럭처링 할당이 이루어진다.
// 프로퍼티 키가 lastName인 프로퍼티 값을 ln에 할당하고,
// 프로퍼티 키가 firstName인 프로퍼티 값을 fn에 할당한다.
const { lastName: ln, firstName: fn } = user;

console.log(fn, ln); // Ungmo Lee
```

배열의 요소가 객체인 경우 배열 디스트럭처링 할당과 객체 디스트럭처링 할당을 **혼용**할 수 있다.

```
const todos = [
  { id: 1, content: 'HTML', completed: true },
  { id: 2, content: 'CSS', completed: false },
  { id: 3, content: 'JS', completed: false }
];

//todos 배열의 두 번째 요소인 객체로부터 id 프로퍼티만 추출한다.
const [, { id }] = todos;
console.log(id); //2
```

객체 디스트럭처링 할당을 위한 변수에 Rest 파라미터나 Rest 요소와 유사하게 **Rest 프로퍼티 ...** 을 사용할 수 있고 **Rest 프로퍼티** 역시 **반드시 마지막에 위치**해야 한다.