



# Chap36 디스트럭처링 할당

## ▼ 디스트럭처링 할당

구조화된 배열 과 같은 이터러블 또는 객체를 destructuring(비구조화, 구조 파괴)하여 1개 이상의 변수에 개별적으로 할당하는 것

즉 구조를 분해하여 할당하는 것을 말한다. 배열, 객체 리터럴에서 필요한 값만 추출하여 변수에 할당할 때 유용

## ▼ 배열 디스트럭처링 할당

### 1. ES5

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

var one=arr[0];
var two=arr[1];
var three=arr[2];

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

### 2. ES6

- **우변**은 배열 디스트럭처링 할당의 대상 즉 **이터러블**이어야한다. 할당 기준은 인덱스, 즉 **순서대로 할당**된다.
- **좌변**에는 할당받을 변수를 선언.( **배열 리터럴 형태로 선언해야한다.**)
- 선언과 할당을 분리할 수 있지만 const 키워드로 변수를 선언할 수 없으므로 권장하지 않는다.
- 변수의 개수와 이터러블의 요소 개수가 반드시 일치할 필요가 없다.
- 변수의 기본 값을 설정할 수 있다.

```
const arr=[1,2,3];
const [one,two]=arr;
const [three]={}; //TypeError:{} is not iterable
```

```
const [four]; //SyntaxError~

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

- Rest요소 사용 가능

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

## ▼ 객체 디스트럭처링 할당

### 1. ES5

- 디스트럭처링 할당을 하기 위해서는 **프로퍼티 키**를 사용했다.

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

var firstName=user.firstName;
var lastName=user.lastName;

console.log(firstName, lastName); //Ungmo Lee
```

### 2. ES6

```
const user={firstName:'Ungmo', lastName:'Lee'};
//프로퍼티 키를 기준으로 디스트럭처링 할당이 이루어지기때문에 순서에 의미
const {lastName, firstName}=user;

console.log(firstName, lastName); // Ungmo Lee
```

- 우변은 객체여야하며 할당 기준은 프로퍼티 키다. 즉, **순서에 의미가 없으며 선언된 변수 이름과 프로퍼티 키가 일치하면 할당된다.**
- 좌변에는 프로퍼티 값을 할당받을 변수를 선언해야한다. (단, 객체 리터럴 형태로, 축약 표현 가능)
- 우변에 객체또는 객체로 평가될 수 있는 표현식을 할당하지 않으면 에러 발생
- 객체의 프로퍼티 키와 다른 변수 이름으로 할당 받기 위해서는 아래와 같이 변수 선언하기

```
const user={firstName:'Ungmo', lastName:'Lee'};
const {lastName:ln, firstName: fn} =user;
console.log(fn,ln); //Ungmo Lee
```

- 변수에 기본값 설정 가능
- 객체 디스트럭처링 할당은 객체를 인수로 전달받는 함수의 매개변수에도 사용 가능
- 배열의 요소가 객체인 경우 배열 디스트럭처링 할당과 혼용하여 사용할 수 있다.

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

const [, {id}]=todos;
console.log(id); //2
```

- Rest프로퍼티 사용할 수 있다. 단 반드시 마지막에 위치하여 사용해야 한다.