

**|Object.keys,
values, entries |**

이은주

2023.05.05

Object.keys()

```
const student = {  
  name: 'eunjoo',  
  age: 22,  
  city: 'suwon'  
};
```

```
const keys = Object.keys(student);  
console.log(keys);
```

```
▶ (3) ['name', 'age', 'city']  VM10:8
```

객체의 키만 담은 배열을 반환

Object.keys()

```
const parent = { a: 1, b: 2 };  
const child = Object.create(parent);  
child.c = 3;
```

```
const keys = Object.keys(child);  
console.log(keys);
```

▶ ['c']

VM28:6

객체의 상속 속성을 제외하고 직접적으로 속성으로 가지고 있는
속성들만을 반환

Object.values()

```
const student = {  
  name: 'eunjoo',  
  age: 22,  
  city: 'suwon'  
};  
  
const values = Object.values(student);  
console.log(values);
```

▶ (3) ['eunjoo', 22, 'suwon'] VM36:8

객체의 속성 값들을 배열 형태로 반환하는 메소드

Object.values()

```
const parent = { a: 1, b: 2 };  
const child = Object.create(parent);  
child.c = 3;  
  
const values= Object.values(child);  
console.log(values);
```

▶ [3]

VM208:6

객체의 상속 속성을 제외하고 직접적으로 속성으로 가지고 있는
속성들만을 반환

Object.entries

```
const student = {  
  name: 'eunjoo',  
  age: 22,  
  city: 'suwon'  
};  
  
const entries =  
Object.entries(student);  
console.log(entries);
```

VM40:8

```
▼ (3) [Array(2), Array(2), Array(2)]  
  i  
  ▶ 0: (2) ['name', 'eunjoo']  
  ▶ 1: (2) ['age', 22]  
  ▶ 2: (2) ['city', 'suwon']  
    length: 3  
  ▶ [[Prototype]]: Array(0)
```

Object.keys() + Object.values() = Object.entries

Object.values()

```
const parent = { a: 1, b: 2 };  
const child = Object.create(parent);  
child.c = 3;
```

```
const entries= Object.entries(child);  
console.log(entries);
```

```
▼ [Array(2)] ⓘ VM285:6  
  ► 0: (2) ['c', 3]  
    length: 1  
  ► [[Prototype]]: Array(0)
```

객체의 상속 속성을 제외하고 직접적으로 속성으로 가지고 있는
속성들만을 반환

Object.fromEntries

```
const entries = [['name', 'eunjoo'],  
['age', 22], ['city', 'suwon']];
```

```
const student =  
Object.fromEntries(entries);  
console.log(student);
```

VM155:4

```
▶ {name: 'eunjoo', age: 22, city: 'suwon'}
```

Object.entries() 메소드와 반대의 역할을 수행하는 메소드

Object.fromEntries

```
const entries = [[1, 'eunjoo'], [2,  
22], [3, 'suwon']];  
const student =  
Object.fromEntries(entries);  
console.log(student);
```

VM182:3

```
▶ {1: 'eunjoo', 2: 22, 3: 'suwon'}
```

ES2022에서 더이상 첫 번째 요소가 문자열 또는 심볼이어야 한다는 제약조건을 가지지 않음