

좋은 질문이에요 😊

JavaScript에서 object의 key / value를 사용하는 방법을 기초부터 차근차근 정리해볼게요.

## 1 객체(Object) 기본 구조

javascript

```
const person = {
  name: "윤우",
  age: 2,
  hobby: "블록놀이"
};
```

- name , age , hobby ➡ key (속성 이름)
- "윤우" , 2 , "블록놀이" ➡ value (값)

📄 코드 복사

## 2 value 가져오기 (가장 중요)

### ✅ 1. 점(.) 표기법 (가장 많이 사용)

javascript

```
console.log(person.name); // 윤우
console.log(person.age); // 2
```

✅ key가 고정된 문자열일 때 사용

📄 코드 복사

### ✅ 2. 대괄호([]) 표기법

javascript

```
console.log(person["name"]); // 윤우
```

✅ key가 변수일 때 반드시 사용

📄 코드 복사

javascript

```
const key = "hobby";
console.log(person[key]); // 블록놀이
```

❗ person.key ❌ (이건 "key"라는 이름을 찾음)

📄 코드 복사

## 3 value 수정하기

javascript

```
person.age = 3;
console.log(person.age); // 3
```

또는

📄 코드 복사

javascript

```
person["hobby"] = "책읽기";
```

📄 코드 복사

#### 4 key / value 추가하기

javascript

```
person.gender = "남";
```

또는

📄 코드 복사

javascript

```
person["height"] = 90;
```

📄 코드 복사

#### 5 key / value 삭제하기

javascript

```
delete person.hobby;
```

📄 코드 복사

#### 6 객체에 key가 있는지 확인

javascript

```
console.log("name" in person); // true
console.log("job" in person); // false
```

또는

📄 코드 복사

javascript

```
console.log(person.hasOwnProperty("age")); // true
```

📄 코드 복사

#### 7 객체 전체 순회하기 (★ 자주 사용)

##### ◆ for...in (key 기준)

javascript

```
for (let key in person) {
  console.log(key, person[key]);
}
```

출력:

arduino

```
name 윤우
age 3
gender 남
height 90
```

#### 8 Object 메서드 (실무에서 많이 씬)

📄 코드 복사

##### ✔ key만 가져오기

javascript

```
Object.keys(person);
// ["name", "age", "gender", "height"]
```

##### ✔ value만 가져오기

📄 코드 복사

```
javascript

Object.values(person);
// ["윤우", 3, "남", 90]
```

✅ key + value 쌍

📄 코드 복사

javascript

```
Object.entries(person);
```

javascript

```
[
  ["name", "윤우"],
  ["age", 3],
  ["gender", "남"],
  ["height", 90]
]
```

9 entries + 구조분해 (실무 예제)

javascript

```
Object.entries(person).forEach(([key, value]) => {
  console.log(`${key} : ${value}`);
});
```

10 객체에서 특정 값 찾기

📄 코드 복사

javascript

```
const hasAge3 = Object.values(person).includes(3);
console.log(hasAge3); // true
```

📄 코드 복사

11 자주 헷갈리는 포인트 ⚠️

❌ 잘못된 예

javascript

```
person[key] // key가 변수일 때만 가능
person.key  // "key"라는 이름을 찾을
```

✅ 정리

📄 코드 복사

상황	사용법
고정된 key	obj.name
변수 key	obj[key]

🔥 다음 단계로 배우면 좋은 것

- 객체 깊은 복사 / 얕은 복사
- 중첩 객체 접근
- 객체 + map, filter
- REST API 응답(JSON) 다루기

원하시면 실무 예제 (React / REST 응답) 기준으로도 설명해 드릴게요 🍌

배열에 안에 딕셔너리 형식 사용하는 방법

아주 좋은 질문이에요 🍌

JavaScript에서 “배열 안에 딕셔너리(객체) 형식” 사용하는 방법을 기초 → 실무 순서로 쉽게 설명할게요.

JS에서는 딕셔너리 = **Object(객체)** 라고 생각하면 됩니다.

## 1 기본 구조 (배열 + 객체)

javascript

```
const users = [
  { id: 1, name: "윤우", age: 2 },
  { id: 2, name: "민준", age: 3 },
  { id: 3, name: "서연", age: 4 }
];
```

- users 📁 배열
- { id: 1, name: "윤우", age: 2 } 📁 객체(딕셔너리)

📄 코드 복사

## 2 배열 안 객체 값 접근하기

### ✅ 인덱스로 접근

javascript

```
console.log(users[0].name); // 윤우
console.log(users[1].age); // 3
```

### ✅ 변수 key로 접근

📄 코드 복사

javascript

```
const key = "name";
console.log(users[0][key]); // 윤우
```

📄 코드 복사

## 3 배열 순회하면서 객체 사용하기 (★ 핵심)

### 💎 forEach (가장 많이 씬)

javascript

```
users.forEach(user => {
  console.log(user.name, user.age);
});
```

📄 코드 복사

### 💎 for...of

javascript

```
for (const user of users) {
  console.log(user.id, user.name);
}
```

## ◆ for (기본)

[📄 코드 복사](#)

javascript

```
for (let i = 0; i < users.length; i++) {  
  console.log(users[i].name);  
}
```

## 4 객체 값 수정하기

[📄 코드 복사](#)

javascript

```
users[0].age = 3;
```

## 5 객체 추가 / 삭제

### ✅ 객체 추가

javascript

```
users.push({ id: 4, name: "지우", age: 1 });
```

### ✅ 객체 삭제

[📄 코드 복사](#)

javascript

```
users.splice(1, 1); // index 1부터 1개 삭제
```

[📄 코드 복사](#)

## 6 특정 조건으로 찾기 (★ 실무 필수)

### 🔍 find (하나 찾기)

javascript

```
const user = users.find(u => u.id === 2);  
console.log(user.name); // 민준
```

[📄 코드 복사](#)

### 🔍 filter (여러 개 찾기)

javascript

```
const result = users.filter(u => u.age >= 3);
```

[📄 코드 복사](#)

## 7 map으로 새 배열 만들기

javascript

```
const names = users.map(u => u.name);  
console.log(names); // ["윤우", "민준", "서연"]
```

[📄 코드 복사](#)

## 8 객체 + 구조 분해 (깔끔한 코드)

javascript

```
users.forEach(({ name, age }) => {  
  console.log(name, age);  
});
```

```
});
```

[📄 코드 복사](#)

## 9 배열 안 객체 정렬하기

javascript

```
users.sort((a, b) => a.age - b.age);
```

[📄 코드 복사](#)

## 10 실제 REST API 응답 예시 (실무 느낌)

javascript

```
const response = {  
  status: 200,  
  data: [  
    { id: 1, product: "우유", price: 1500 },  
    { id: 2, product: "과자", price: 1200 }  
  ]  
};
```

[📄 코드 복사](#)

```
response.data.forEach(item => {  
  console.log(item.product, item.price);  
});
```

## 1 자주 하는 실수 ⚠

### ✖ 잘못된 접근

javascript

```
users.name // ✖ 배열에는 name이 없음
```

[📄 코드 복사](#)

### ✔ 올바른 접근

javascript

```
users[0].name
```

[📄 코드 복사](#)

## 🔥 정리 한 줄

배열은 순서(index), 객체는 key → 배열[index].key

## 다음으로 배우면 좋은 것 🚀

- 배열 안 객체 중첩 구조
- reduce 로 합계 계산
- React에서 리스트 렌더링
- 백엔드(JSON) 데이터 처리

원하시면 **Spring REST API 응답** → **React에서 처리하는 예제**로도 설명해 드릴게요 😊

📄 🌟 🔖 ↶ ↷ ...

