Object의 메소드

Object.assign 메소드

Object.assign(to, from)

from 객체의 모든 멤버 변수 값을, to 객체에 복사한다.

from 객체의 멤버 변수와 같은 이름의 멤버 변수가 to 객체 있다면,
to 객체의 멤버 변수의 값이 from 객체의 멤버 변수 값으로 바뀐다.

from 객체의 멤버 변수와 같은 이름의 멤버 변수가 to 객체에 없다면,
그 멤버 변수가 to 객체에 새로 만들어지고 값이 대입된다.

assign 메소드는 to 객체를 리턴한다.

```
let person = { name: "홍길동", age: 16 };
let info = { age: 20, department: "소프", year: 2 };

Object.assign(person, info);
console.log(person);
```

```
// [object Object]
{
  "name": "홍길동",
  "age": 20,
  "department": "소프",
  "year": 2
}
```

객체 복제

```
let person1 = { name: "홍길동", age: 16 };

let person2 = Object.assign({ }, person1);
console.log(person2); //{ name: '홍길동', age: 16 }
```

let person2 = Object.assign({ }, person1);
person1 객체의 모든 멤버 변수값을 { } 빈 객체에 복사한다.
그렇게 값이 채워진 객체가 리턴되어, person2 변수에 대입된다.

즉 person1 객체를 복제한 새 객체가 만들어져서, person2 변수에 대입되었다.

객체 복제2

```
let person1 = { name: "홍길동", age: 16 };

let person2 = Object.assign({ }, person1);
console.log(person2);

let person3 = { ...person1 };
console.log(person3);
```

`let person3 = { ...person1 };`
person1 객체의 모든 속성이 대입된 새 객체가 생성된다.
즉 person1 객체가 복제된다.

Object.entries

Object.entries(객체)

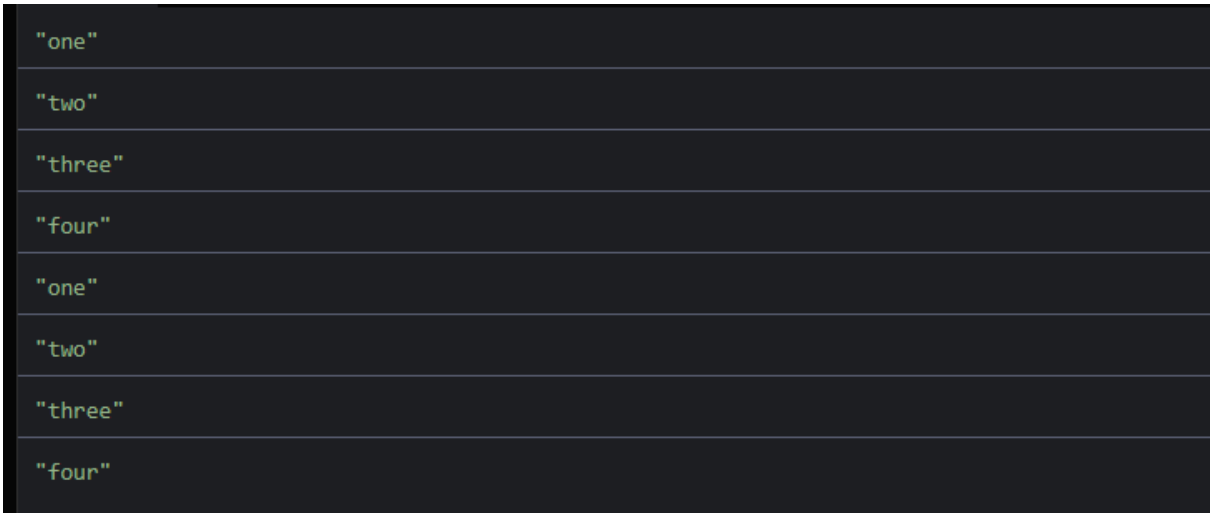
객체의 모든 멤버 변수 값이 들어있는 2차원 배열을 리턴한다.
리턴되는 배열은 다음과 같은 형태이다.

[[멤버변수이름1, 값], [멤버변수이름2, 값], [멤버변수이름3, 값], ...]

```
let person = { name: "홍길동", age: 16 };

console.log(Object.entries(person)); //[ [ 'name', '홍길동' ], [ 'age', 16 ] ]
```

for of

```
let a = ["one", "two", "three", "four"];

for (let i = 0; i < a.length; ++i)
    console.log(a[i]);

for (let s of a)
    console.log(s);
```




```
let person = { name: "홍길동", age: 16 };
```

```
for (let a of Object.entries(person))
  console.log(a);
```

```
// [object Array] (2)
["name","홍길동"]
// [object Array] (2)
["age",16]
```

```
let person = { name: "홍길동", age: 16 };

for (let [key, value] of Object.entries(person))
  console.log(key, value);
```

```
"name" "홍길동"
"age" 16
```

Object.entries(person) 메소드가 리턴하는 배열의 원소 각각에 대해서,
그 원소를 [key, value]에 대입하여 (destructuring assignment)
for 문의 본문을 반복 실행한다.

Object.entries(person) 메소드가 리턴하는 배열의 원소는
[멤버변수이름, 값] 형태의 배열이다.

Object.keys, Object.values

Object.keys(객체)

객체의 멤버 변수 이름 목록을 배열로 리턴하다.

```
let person = { name: "홍길동", age: 16, department: "소프" };

console.log(Object.keys(person)); //[ 'name', 'age', 'depar
tment' ]
```

Object.values(객체)

객체의 멤버 변수 이름 목록을 배열로 리턴하다.

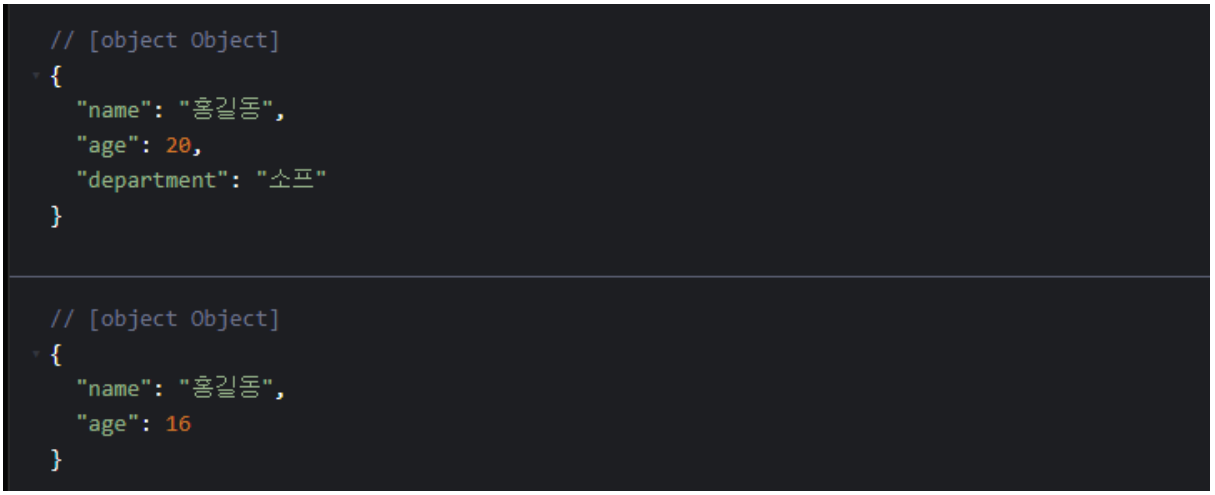
```
let person = { name: "홍길동", age: 16, department: "소프" };

console.log(Object.values(person)); //[ '홍길동', 16, '소프'
]
```

Object.freeze 메소드

Object.freeze(객체)

객체를 수정할 수 없는 상태로 변경한다.
객체 내부 값을 변경할 수 없게 된다.

```
let person1 = { name: "홍길동", age: 16 };
person1.age = 20;
person1.department = "소프";
console.log(person1);

let person2 = { name: "홍길동", age: 16 };
Object.freeze(person2);
person2.age = 20;
person2.department = "소프";
console.log(person2);
```



`Object.freeze(person2);`
person2 객체를 수정할 수 없게 된다.
멤버 변수 값을 변경할수 없고, 새 멤버 변수를 추가할 수도 없고, 멤버 변수를 제거할 수도 없다.

Object.isFrozen(객체)

객체가 freeze 되었는지 여부를 리턴한다.

```
let person = { name: "홍길동", age: 16 };
console.log(Object.isFrozen(person)); //fasle

Object.freeze(person);
console.log(Object.isFrozen(person)); //true
```