@linkinsunil

JAVASCRIPT OBJECT METHODS

Object.keys()

A simple way to iterate over an object and return all the object's keys

```
const employee = {
  name: "Bruce",
  age: 35,
  occupation: "Batman",
  level: 10
}

console.log(Object.keys(employee))
```

Output :

```
▶ (4) ["name", "age", "occupation", "level"]
```

@linkinsunil @officialskv

Object.values()

Iterates over the object and return all the object's values

```
const employee = {
  name: "Bruce",
  age: 35,
  occupation: "Batman",
  level: 10
}

console.log(Object.values(employee))
```

Output :

```
▶ (4) ["Bruce", 35, "Batman", 10]
```

Object.entries()

Takes an object and returns its own enumerable string - keyed property [key : value] pairs of the object

```
const employee = {
  name: "Bruce",
  age: 35
};

for (const [name, age] of
Object.entries(employee)) {
  console.log(`${name}: ${age}`);
}
```

Output :

```
name: Bruce
age: 35
```

Object.create()

Creates a new object using an existing object as the prototype of the newly created object

```
let Superhero = {
    name: "Bruce",
    display() {
      console.log("Name :", this.name);
    }
};

let superhero1 = Object.create(Superhero);
superhero1.name = "Batman";
superhero1.display();
```

Output :

```
Name :  Batman
```

Object.assign()

Copies all enumerable and own properties from the source object to the target object. It returns the target object. It is also called shallow copy.

```
const target = { a: 1, b: 2 };
 const source = { b: 4, c: 5 };

const newTarget  = Object.assign(target, source);
console.log(target);
console.log(newTarget);
```

Output :

```
▶ {a: 1, b: 4, c: 5}
```

Object.seal()

Prevents new properties from being added to the object and prevents existing properties from removing

```
const pen = {
  price: 1000
};

Object.seal(pen);
pen.price = 500;
console.log(pen); // {price: 500}

delete pen.price;
pen.brand = "cello";
console.log(pen); // {price: 500}
```

Object.freeze()

- New properties can not be added
- Existing properties can not be removed
- Can not change the enumerability, configurability or writability of existing properties
- Can not change values of existing object properties and prototype.

```
const pen = {
  price: 1000
};

Object.freeze(pen);
pen.price = 500; //shows error in strict mode
console.log(pen.price); // 1000
```

Useful ? Let me know in the comments

Sunil Vishwakarma

Frontend Developer

and **Follow** for more