

Դաս 06: JS OOP Objects

Նախքան **OOP**-ն ուսումնասիրել, վերհիշենք թե ինչ է իրենից ներկայացնում օբյեկտները **JS**-ում: Օբյեկտը ստրուկտուրա է, որում տվյալները ներկայացված են որպես բանալիների և նրանց արժեքների զույգեր:

Օբյեկտը հայտարարվում է զույգ ձևավոր փակագծերի օգնությամբ: Սկզբում գրվում է բանալի բառը, այնուհետև վերջակետ և բանալի բառի արժեքը: Օբյեկտի մեջ կարելի է գրել ցանկացած բանակության բանալի բառեր իրենց արժեքներով: Մեկից ավելի բանալի բառերը իրենց արժեքներով պետք է միմյանցից առանձնացվեն ստորակետով:

Այժմ ուսումնասիրենք օբյեկտների մեթոդները`

1. Object.create()

create() մեթոդը թույլ է տալիս ստեղծել օբյեկտ օգտագործելով որպես օրինակ մեկ այլ օբյեկտ:

```
const animal = { eats: true };
const dog = Object.create(animal); // ստեղծում ենք dog
// օբյեկտ օգտագործելով որպես նմուշ animal
dog.barks = true;

console.log(dog.eats); // true (ժառանգվել է animal ից)
console.log(dog.barks); // true (սեփական հատկություն)
```

2. Object.assign()

assign() մեթոդը թույլ է տալիս ստեղծել օբյեկտ օգտագործելով որպես օրինակ մեկից ավելի օբյեկտներ: Եթե նմուշներից մի քանիսը ունեն կրկնվող բանալի բառեր, կցուցադրվի վերջին օբյեկտի բանալի բառը իր արժեքով:

```
const target = { name: "Alice" };
const source1 = { age: 25, name: "Alice2" };
const source2 = { profession: "Engineer" };

Object.assign(target, source1, source2);
```

```
console.log(target);  
// { name: "Alice2", age: 25, profession: "Engineer" }
```

3. delete obj.key

Delete obj.key մեթոդը ջնջում է օբյեկտի նշված մեթոդը:

```
const car = { brand: "Toyota", model: "Camry" };  
delete car.model; // ջնջում է "model"-ը  
console.log(car);  
// { brand: "Toyota" }
```

4. Object.defineProperty()

defineProperty() մեթոդը թույլ է տալիս սահմանել օբյեկտի որոշ բանալիների համար առանձնահատուկ հատկություններ՝

```
Object.defineProperty(user, "id", {  
  value: 123,  
  writable: false, // Չի կարելի արժեքը փոխել  
  enumerable: true, // Տեսանելի է լիցենզիոն ցիկլներում  
  configurable: false // Չի կարելի ջնջել  
});  
  
console.log(user.id); // 123  
user.id = 456; // Չի փոփոխվի  
console.log(user.id); // 123
```

5. Object.keys(obj)

Object.keys(obj) մեթոդը ստանում է օբյեկտի բանալիները և վերադարձնում զանգվածում

```
console.log(Object.keys(book));  
// ["title", "author", "year"]
```

6. Object.values(obj)

Object.values(obj) մեթոդը ստանում է օբյեկտի բանալիները և արժեքները: Վերադարձնում է զանգվածում որպես մեկ տողային արտահայտություն, եթե արժեքը տողային տիպի է, և որպես 2 առանձին արտահայտություններ, եթե տողային տիպի տվյալ չէ:

```
console.log(Object.values(book));  
// ["JavaScript: The Good Parts", "Douglas Crockford",  
2008]
```

7. Object.entries(obj)

Object.entries(obj) մեթոդը վերադարձնում է զանգված, որի մեջ զանգվածի տեսքով օբյեկտի բանալիները և արժեքները:

```
console.log(Object.entries(book));  
// [[ "title", "JavaScript: The Good Parts" ],  
[ "author", "Douglas Crockford" ], [ "year", 2008 ]]
```

8. spread-օպերատոր

Այն պատճենում է օբյեկտը:

```
const original = { a: 1, b: 2 };  
const copy = { ...original };  
console.log(copy); // { a: 1, b: 2 }
```

9. key in object

In մեթոդը թույլ է տալիս ստուգել, արդյոք բանալին առկա է օբյեկտում թե ոչ:

```
const user = { name: "John", age: 30 };  
  
console.log("name" in user); // true  
console.log("height" in user); // false
```

10. object.hasOwnProperty(key)

hasOwnProperty մեթոդը նույնպես թույլ է տալիս ստուգել բանալին առկա է օբյեկտում թե ոչ

```
const person = { name: "Alice", age: 25 };  
console.log(person.hasOwnProperty("name")); // true  
console.log(person.hasOwnProperty("gender")); // false
```

11. Object.getPrototypeOf(obj) === prototypeName

getPrototypeOf() մեթոդը թույլ է տալիս ստուգել, արդյոք տվյալ օբյեկտը հանդիսանում է մեկ այլ օբյեկտի ժառանգը թե ոչ:

```
const proto = { greeting: "Hello" };  
const obj = Object.create(proto);
```

```
console.log(Object.getPrototypeOf(obj) === proto); //  
true  
console.log(obj.greeting);
```

12. Object.freeze(obj)

freeze() մեթոդը սառեցնում է օբյեկտին և թույլ չի տալիս, որ որևէ տվյալ փոփոխվի:

```
const config = { server: "localhost", port: 8080 };  
Object.freeze(config);  
  
config.port = 3000; // չի փոփոխվի  
console.log(config);  
// { server: "localhost", port: 8080 }
```