

Object and it's Method in

JavaScript

How to Create objects in JavaScript?

```
const person = {  
    name: 'Deepa'  
};
```

This is simple and most popular way

* you can also use new keyword

```
const person = new Object();  
person.name = 'Deepa';
```

* you can also use 'new' with user defined constructor function

```
Eg → function Person(name) {  
    this.name = name;  
}
```

Now anytime you want Person object

```
const personOne = new Person('deepa');
```

* Using Object.create() to create new Objects

The Object.create() methods creates a new object, using an existing object as prototype of the newly created object

It contains 2 parameter:

- First parameter is mandatory that serves prototype of new object to be created
- Second is optional, it contains properties to be added to new object

Eg → const orgObject = { company: 'ABC' };

```
const employee = Object.create(orgObject,
  { name: { value: 'EmpOne' } } );
```

```
console.log(employee); // { company: 'ABC' }
console.log(employee.name); // 'EmpOne'
```

Date: / /

* Using Object.assign() to create new obj.

The Object.assign() method is used to copy all enumerable own properties value from one or more source objects to target object.

It will return target object.

Eg → const orgObject = { company: 'ABC' }
const carObject = { carName: 'Ford' }

const employee = Object.assign({}, orgObject, carObject);

Now you can get employee object that has company and carName as its property

```
console.log(employee);
// { carName: 'Ford', company: 'ABC' }
```

* Using Object.defineProperties()

This method defines new or modify existing property on object

Date: / /

```
const object1 = {};
```

```
Object.defineProperties(object1, {  
    property1: {  
        value: 42,  
        get: () =>  
            value + 1  
    }  
});
```

```
console.log(object1.property1); // 42
```

Similarly we also have `Object.defineProperty()`

* Using `Object.entries()`

It returns an array of object's key value pairs.

The order of array is same as provided by a `for...in` loop

```
const object1 = {  
    a: 'something',  
    b: 'nothing'  
};
```

for (const [key, value] of Object.entries(object))

```
{ console.log(` ${key}: ${value}`); }
```

```
// "a": something
  "b": nothing
```

Object.freeze()

It freezes an object
No longer can be changed

Eg - const obj = {
 prop: 42
};

```
Object.freeze(obj);  
console.log(obj)
```

```
obj.prop = 43;  
console.log(obj.prop);
```

// output 42
It can no longer be change due to freeze

Date: / /

Object.fromEntries()

It transforms a list of key-value pairs into an object

Eg →

```
const entries = new Map([  
  ['foo', 'bar'],  
  ['baz', 42]  
]);
```

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

```
// Object { foo: "bar", baz: 42 }
```