

A BLOG ABOUT OBJECTS AND ITS INTERNAL REPRESENTATION IN JAVASCRIPT

- In JavaScript, almost "everything" is an object.
- Nearly all objects in JavaScript are instances of Object; a typical object inherits properties (including methods) from Object.prototype, although these properties may be shadowed
- The Object class represents one of JavaScript's data types. It is used to store various keyed collections and more complex entities.

All JavaScript values, **except primitives**, are objects.

To begin with let us understand the difference between **Primitive Types And Objects**

JavaScript defines 5 types of primitive data types:

- String
 - Number
 - Boolean
 - Null
 - Undefined
1. **Booleans** can be objects (if defined with the new keyword)
 2. **Numbers** can be objects (if defined with the new keyword)
 3. **Strings** can be objects (if defined with the new keyword)
 4. **Dates** are always objects
 5. **Maths** are always objects
 6. **Regular expressions** are always objects
 7. **Arrays** are always objects
 8. **Functions** are always objects
 9. **Objects** are always objects

JavaScript variables can contain **single values**

Primitive values are immutable (they are hardcoded and therefore cannot be changed).

if $x = 3.14$, you can change the value of x . But you cannot change the value of 3.14.

Objects are variables too. But objects can contain many values.

In Objects , The values are written as name : **VALUE PAIRS**

Example

```
var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};
```

A JavaScript object is a **collection of named values** and the **named values**, in JavaScript objects, are called **properties**.

Creating a JavaScript Object

With JavaScript, you can define and create your own objects.

There are different ways to create new objects:

Define and create a single object, using an object literal.

Ex: `var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};`

Define and create a single object, with the keyword **new**.

```
Ex: var person = new Object();
person.firstName = "John";
person.lastName = "Doe";
person.age = 50;
person.eyeColor = "blue";
```

Define an **object constructor**, and then create objects of the constructed type.

```
function obj(value1, value2, ...) {
  this.key1 = value1;
  this.key2 = value2;
  ...
}
```

In ECMAScript 5, an object can also be created with the function **Object.create()**.

Object.create() can be used for **single inheritance** and **Object.assign()** for **multiple inheritance**.
Object.fromEntries() — method transforms a list of key-value pairs into an object.

JavaScript Objects are Mutable

Objects are mutable: They are addressed by reference, not by value.

If person is an object, the following statement will not create a copy of person:

Ex: `var x = person;`

- This will not create a copy of person.
- The object x is not a copy of person. It is person. Both x and person are the same object.
- Any changes to x will also change person, because x and person are the same object.