

Working With the Object Type

In this lesson, we'll cover the object type in JavaScript.
Let's begin!

WE'LL COVER THE FOLLOWING



- Instantiating and accessing objects
- Examples



The Object Type



You have already learned that objects are the cornerstones of the JavaScript language. Objects in JavaScript, just like in most programming languages with object-oriented features, keep data and functionality together.

JavaScript objects are very different in their implementation and behavior from traditional objects in Java, C++, or C#.

In this lesson, you will learn the basics of the object type. You'll get acquainted with nitty-gritty details in [Chapter 9](#).

Instantiating and accessing objects

All objects in JavaScript set out as nonspecific groups of data and related functionality. You create instances by utilizing a new operator that is followed by the name of the object type (constructor function).

The easiest way to instantiate objects is the `Object()` constructor that accepts an optional value as its argument. This value is stored as the wrapped value of the object, and it can be accessed through the `valueOf()` function.

If you do not use the constructor argument or specify either null or undefined, an empty object will be used as the wrapped object value:

Examples

```
var obj1 = new Object();
var obj2 = new Object(null);
var obj3 = new Object(undefined);
var obj4 = new Object(12.3);
var obj5 = new Object(true);
var obj6 = new Object("Hi");
console.log(obj1.valueOf()); // Object {}
console.log(obj2.valueOf()); // Object {}
console.log(obj3.valueOf()); // Object {}
console.log(obj4.valueOf()); // 12.3
console.log(obj5.valueOf()); // true
console.log(obj6.valueOf()); // Hi
```



If you do not pass any argument to the `Object` constructor, you may omit parentheses:

```
var obj = new Object;
```



Nonetheless, it is a practice to *avoid*.

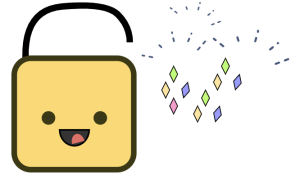
In many object-oriented languages, `Object` is a root class of a hierarchy, and all other types inherit its properties, methods, and other members.

In JavaScript, object properties and methods are present on all object

In JavaScript, every object has properties and methods and is present on an object instances including other, more specific objects.

Achievement unlocked!

Congratulations! You've learned the basics of JavaScript objects.



Great work! Give yourself a round of applause! :)

In the *next lesson*, we will get acquainted with function prototypes.