You learned about [primitive and structured data types in JavaScript](https://www.tutorialsteacher.com/javascript/javascript-data-types). An object is a non-primitive, structured data type in JavaScript. Objects are same as variables in JavaScript, the only difference is that an object holds multiple values in terms of properties and methods.

In JavaScript, an object can be created in two ways:

1) using Object Literal/Initializer Syntax

2) using the Object() Constructor function with the [new keyword](https://www.tutorialsteacher.com/javascript/new-keyword-in-javascript).

var p1 = { name:"Steve" }; // object literal syntax

var p2 = new Object(); // Object() constructor function

p2.name = "Steve"; // property

Objects can be declared same as [variables](https://www.tutorialsteacher.com/javascript/javascript-variable) using var or let keywords.

The p1 object is created using the object literal syntax (a short form of creating objects) with a property named name.

The p2 object is created by calling the Object() constructor function with the new keyword. The p2.name = "Steve"; attach a property name to p2 object with a [string](https://www.tutorialsteacher.com/javascript/javascript-string) value "Steve".

Create Object using Object Literal Syntax

The object literal is a short form of creating an object. Define an object in the { } brackets with key:value pairs separated by a comma. The key would be the name of the property and the value will be a literal value or a function.

Syntax:

var <object-name> = { key1: value1, key2: value2,...};

Example: Object Literal Syntax

var emptyObject = {}; // object with no properties or methods

var person = { firstName: "John" }; // object with single property

// object with single method

var message = {

showMessage: function (val) {

alert(val);

}

};

// object with properties & method

var person = {

firstName: "James",

lastName: "Bond",

age: 15,

getFullName: function () {

return this.firstName + ' ' + this.lastName

}

};

Note that the whole key-value pair must be declared. Declaring only a key without a value is invalid, as shown below.

Example: Wrong Syntax

var person = { firstName };

var person = { getFullName: };

Create Objects using Objects() Constructor

Another way of creating objects is using the Object() constructor function using the [new](https://www.tutorialsteacher.com/javascript/new-keyword-in-javascript) keyword. Properties and methods can be declared using the dot notation .property-name or using the square brackets ["property-name"], as shown below.

Example: Create Object using Object() Constructor

var person = new Object();

// Attach properties and methods to person object

person.firstName = "James";

person["lastName"] = "Bond";

person.age = 25;

person.getFullName = function () {

return this.firstName + ' ' + this.lastName;

};

An object can have variables as properties or can have computed properties, as shown below.

Example: Variables as Object Properties

var firstName = "James";

var lastName = "Bond";

var person = { firstName, lastName }

Access JavaScript Object Properties & Methods

An object's properties can be accessed using the dot notation obj.property-name or the square brackets obj["property-name"]. However, method can be invoked only using the dot notation with the parenthesis, obj.method-name(), as shown below.

Example: Access JS Object

var person = {

firstName: "James",

lastName: "Bond",

age: 25,

getFullName: function () {

return this.firstName + ' ' + this.lastName

}

};

person.firstName; // returns James

person.lastName; // returns Bond

person["firstName"];// returns James

person["lastName"];// returns Bond

person.getFullName(); // calling getFullName function

In the above example, the person.firstName access the firstName property of a person object. The person["firstName"] is another way of accessing a property. An object's methods can be called using () operator e.g. person.getFullName(). JavaScript engine will return the function definition if accessed method without the parenthesis.

Accessing undeclared properties of an object will return [undefined](https://www.tutorialsteacher.com/javascript/javascript-null-and-undefined).

If you are not sure whether an object has a particular property or not, then use the hasOwnProperty() method before accessing them, as shown below.

Example: hasOwnProperty()

var person = new Object();

person.firstName; // returns undefined

if(person.hasOwnProperty("firstName")){

person.firstName;

}

The properties and methods will be available only to an object where they are declared.

Example: Object Constructor

var p1 = new Object();

p1.firstName = “James”;

p1.lastName = “Bond”;

var p2 = new Object();

p2.firstName; // undefined

p2.lastName; // undefined

p3 = p1; // assigns object

p3.firstName; // James

p3.lastName; // Bond

p3.firstName = “Sachin”; // assigns new value

p3.lastName = “Tendulkar”; // assigns new value

Enumerate Object’s Properties

Use the for in loop to enumerate an object, as shown below.

Example: Access Object Keys

var person = new Object();

person.firstName = “James”;

person.lastName = “Bond”;

for(var prop in person){

alert(prop); // access property name

alert(person[prop]); // access property value

};

Pass by Reference

Object in JavaScript passes by reference from one function to another.

Example: JS Object Passes by Reference

function changeFirstName(per)

{

per.firstName = “Steve”;

}

var person = { firstName : “Bill” };

changeFirstName(person)

person.firstName; // returns Steve

Nested Objects

An object can be a property of another object. It is called a nested object.

Example: Nested JS Objects

var person = {

firstName: “James”,

lastName: “Bond”,

age: 25,

address: {

id: 1,

country:”UK”

}

};

person.address.country; // returns “UK”

 Points to Remember :

1. JavaScript object is a standalone entity that holds multiple values in terms of properties and methods.
2. Object property stores a literal value and method represents function.
3. An object can be created using object literal or object constructor syntax.
4. Object literal:

var person = {

firstName: “James”,

lastName: “Bond”,

age: 25,

getFullName: function () {

return this.firstName + ‘ ‘ + this.lastName

}

};

Object constructor:

var person = new Object();

person.firstName = "James";

person["lastName"] = "Bond";

person.age = 25;

person.getFullName = function () {

return this.firstName + ' ' + this.lastName;

};

Object properties and methods can be accessed using dot notation or [ ] bracket.

An object is passed by reference from one function to another.

An object can include another object as a property.