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

This code assigns **many values** (Fiat, 500, white) to a **variable** named car:

```
var car = {type:"Fiat", model:"500", color:"white"};
```

The values are written as **name:value** pairs (name and value separated by a colon).

JavaScript objects are containers for **named values** called properties or methods.

Object Definition

You define (and create) a JavaScript object with an object literal:

Example

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

Spaces and line breaks are not important. An object definition can span multiple lines:

Example

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

Object Properties

The **name:values** pairs in JavaScript objects are called **properties**:

Property	Property Value
firstName	John
lastName	Doe
Age	50
eyeColor	blue

Accessing Object Properties

You can access object properties in two ways:

objectName.propertyName

or

objectName["propertyName"]

Example1

```
person.lastName;
```

Example2

```
person["lastName"];
```

Object Methods

Objects can also have **methods**.

Methods are **actions** that can be performed on objects.

Methods are stored in properties as **function definitions**.

Property	Property Value
firstName	John
lastName	Doe
Age	50
eyeColor	blue
fullName	function() {return this.firstName + " " + this.lastName;}

A method is a function stored as a property.

Example

```
var person = {
  firstName: "John",
  lastName : "Doe",
  id       : 5566,
  fullName : function() {
    return this.firstName + " " + this.lastName;
  }
};
```

The **this** Keyword

In a function definition, `this` refers to the "owner" of the function.

In the example above, `this` is the **person object** that "owns" the `fullName` function.

In other words, `this.firstName` means the `firstName` property of **this object**.

Read more about the `this` keyword at JS this Keyword.

Accessing Object Methods

You access an object method with the following syntax:

objectName.methodName()

Example

```
name = person.fullName();
```

If you access a method **without** the () parentheses, it will return the **function definition**:

Example

```
name = person.fullName;
```

Do Not Declare Strings, Numbers, and Booleans as Objects!

When a JavaScript variable is declared with the keyword "`new`", the variable is created as an object:

```
var x = new String();        // Declares x as a String object
var y = new Number();        // Declares y as a Number object
var z = new Boolean();       // Declares z as a Boolean object
```

Avoid `String`, `Number`, and `Boolean` objects. They complicate your code and slow down execution speed.

You will learn more about objects later in this tutorial.

Array Methods

1. **length property** --> If you want to know the number of elements in an array, you can use the length property.
2. **prototype property** --> If you want to add new properties and methods, you can use the prototype property.
3. **reverse method** --> You can reverse the order of items in an array using a reverse method.
4. **sort method -->** You can sort the items in an array using sort method.
5. **pop method** --> You can remove the last item of an array using a pop method.
6. **shift method** --> You can remove the first item of an array using shift method.
7. **push method** --> You can add a value as the last item of the array.

```
<html>
<head>
	<title>Arrays!!!</title>
	<script type="text/javascript">
		var students = new Array("John", "Ann", "Aaron", "Edwin", "Elizabeth");
		Array.prototype.displayItems=function(){
			for (i=0;i<this.length;i++){
				document.write(this[i] + "<br />");
			}
		}
		document.write("students array<br />");
		students.displayItems();
		document.write("<br />The number of items in students array is " + students.length
+ "<br />");
```

```
                document.write("<br />The SORTED students array<br />");
                students.sort();
                students.displayItems();
                document.write("<br />The REVERSED students array<br />");
                students.reverse();
                students.displayItems();
                document.write("<br />THE students array after REMOVING the LAST item<br />");
                students.pop();
                students.displayItems();
    document.write("<br />THE students array after PUSH<br />");
    students.push("New Stuff");
                students.displayItems();
        </script>
</head>
<body>
</body>
</html>
```