WSA | Forward looking IT finishing school

Objects

JavaScript

Table of Content

- Objects

Objects

(JavaScript)

Objects

- JavaScript is an “object-based” language
- In JavaScript, everything is an object (except for language constructs, keywords and operators)
- Objects plays many roles from representing data to manipulation of HTML documents via document object model (DOM), to interface with browser and more

Objects

- An object is an unordered collection of data, including primitive types, functions and other objects
- An object is a collection of properties, each of which has a name and a value
- JavaScript objects are dynamic—properties can be added and deleted

Types of Objects

- User Defined Objects
- Native Objects
- Host Objects
- Document Objects

User Defined Objects

- User defined objects are custom objects created by the programmer
- Objects can be nested within other objects and this allows to create complex (composite) data structures consisting of data and methods

Native Objects

- These include objects associated with data types such as String, Number and Boolean, as well as objects that allow creation of user defined objects and composite types
- Native objects include JavaScript Functions, Date, Math and RegExp manipulation
- Exception handling and error are also native objects
- Native objects are governed by the ECMAScript language standard

Host Objects

- Host objects are those objects that are supported by host environments typically as browser
- Browser based host objects include window, navigator, XMLHTTPRequest, location, History etc

Document Objects

- Document objects are part of the Document Object Model (DOM), as defined by W3C

Creating Objects

- Objects can be created by two ways
  - Object Literal
  - Constructor Function

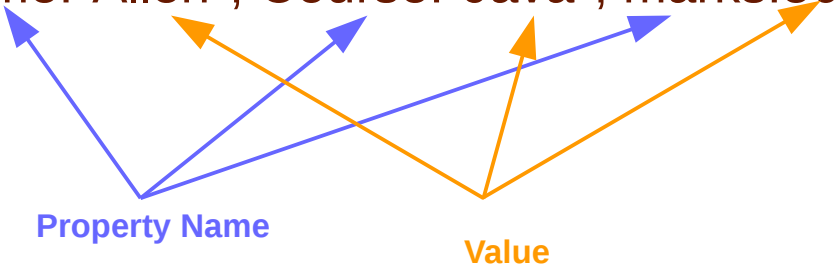
Object Literal

Syntax :

```
var objectName = { }; //Plain object
```

- Creating Object by defining property and values

```
var student = {Name:"Allen", Course:"Java", marks:50}
```



Object Methods

- Methods are actions that can be performed on objects
- A JavaScript method is a property containing a function definition

Accessing Object Methods

Creating Object Method

```
methodName : function()
{
    . . . code . . .
}
```

Accessing Object Method

```
objectName.methodName;
```

Constructor Function

- Constructor function is the another way to create objects
- Rules for construction function
  - Constructor function name will be the object type
  - ‘this’ operator is used in construction function to refer to the object
  - There shall be no “return” statement in constructor function

Constructor Function
(Example)

```
function Student (pname, pcourse, pmarks) {
    this.name = pname;
    this.course = pcourse;
    this.marks = pmarks;
}
/* The type of above object is Student
   name, course, marks are properties */
 student = new Student(“Mac”, ”Java”, 65); // student is an object
```

Accessing Object
(Properties)

You can access object properties in two ways:

Syntax :
objectName.propertyName
or
objectName[propertyName]

Example :
student.name

Objects - Example

```
<script>
var student = {
    firstName : "Smith",
    surName  : "Joe",
    course      : "Java",
    marks       :  50,
    fullName   : function() {
        return this.firstName + " " + this.surName;
    }
};
</script>
```

Objects - Example

```
<p id="ex"></p>
<p id="ex1"></p>
<script>
. . .
document.getElementById("ex").innerHTML =
// Accessing Properties
student.firstName + " " + student.course + " " + student.marks;
// Accessing Methods
document.getElementById("ex1").innerHTML = student.fullName();

</script>
</body>
```

Object Garbage Collection

- JavaScript uses garbage collection to automatically delete objects when they are no longer referenced

```
Example :
var s1 = new Student(“MAC” ,”Java”, 65);
//clean up by setting the object to null
s1 = null;
```

Deleting Object Properties

- The delete operator can be used to completely remove properties from an object
- Setting the property to undefined or null only changes the value of a property and does not result removal of the property

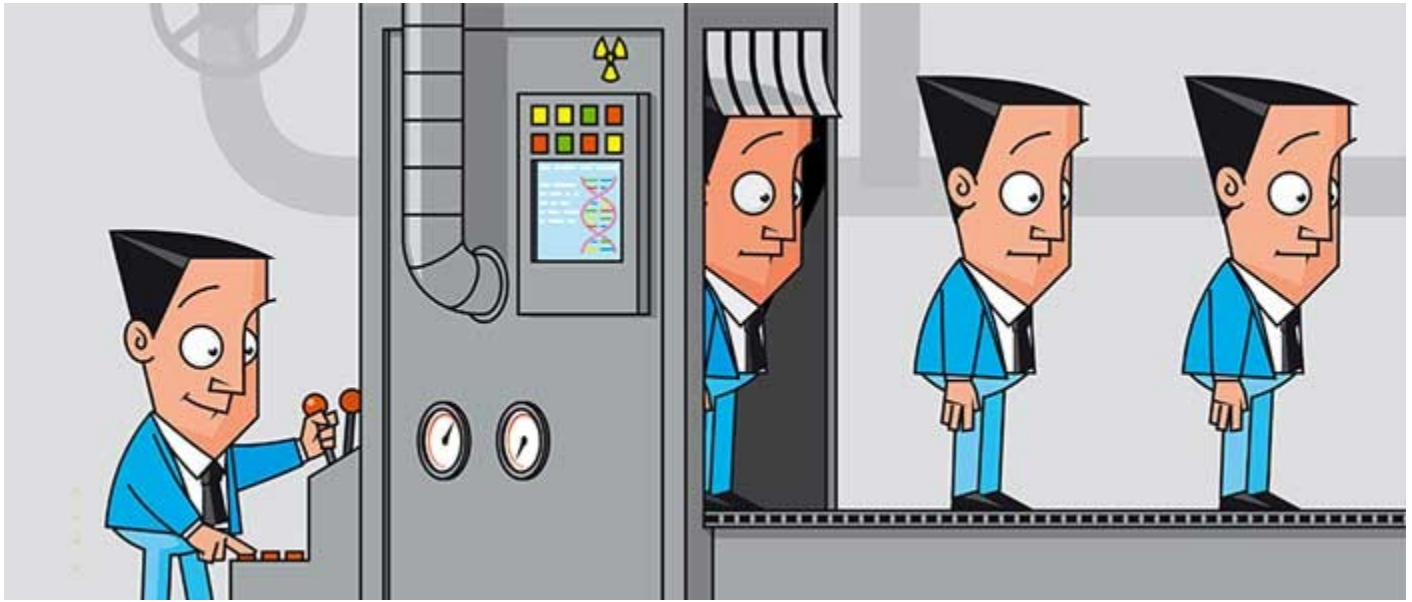
Syntax :

```
delete objectName.propertyname;
```

Exercise

- Write a JavaScript program to create object product, add the property Product Name, Quantity and price. Access all the properties and display them.

Prototype



JavaScript Object Prototype

JavaScript is an object-based language based on prototypes. There are some features :

- Every JavaScript object has a prototype. The Prototype is also an object.
- All JavaScript objects inherit properties from their prototype.
- Define and create a set of objects with constructor functions.
- Can add or remove properties dynamically to individual objects or to the entire set of objects.
- Inherit properties by following prototype chain.

Creating a Prototype Example

```
<body>
<p id="ex"></p>
<script>
function student(name, age , marks) {
    this.name = name;
    this.age =age;
    this.marks=marks;
}
```

```
var javaStud = new student("Smith", 25 , 65 );
var embedStud = new student("Mac", 24, 62);

// Creating two new objects using new keyword
// From same prototype student
document.getElementById("ex").innerHTML =
"Java Student Marks " + javaStud.marks +
"Embedded Student Marks " + embedStud.marks;
</script>
</body>
```

JavaScript Prototypes

- A prototype is an internal object from which other objects inherit properties.
- Its main purpose is to allow multiple instances of an object to share a common property.
- Objects created using an object literal, or with new Object(), inherit from a prototype called Object.prototype.
- Objects created with new String() inherit the String.prototype.
- A prototype can have a prototype, and so on.
- The Object.prototype is on the top of the prototype chain.
- All JavaScript objects (Date, Array, RegExp, Function, ...) inherit from the Object.prototype.

JavaScript Prototypes

The prototype property allows to add properties and method to an object.

The value holding the prototype of an object is sometimes called the internal prototype link. It's also been historically called __proto__.

Declaring prototype property

object.prototype.name=value ;

or

object._proto_.name=value;

Prototype Property Example

```
<script>
function employee(name, dept)
{
  this.name = name;
  this.dept = dept;
}

employee.prototype.salary = 20000;
var xy = new employee("Mac", "CS");
document.write(xy.salary);
</script>
```

Inheritance via Prototype Chain

- In JavaScript, the inheritance is prototype-based. That means that there are no classes. Instead, an object inherits from another object.
- Its main purpose is to allow multiple instances of an object to share a common property.
- Thus object properties which are defined using the prototype object are inherited by all instances which reference it.
- Adding properties and methods to the prototype property will automatically add the method or property to all objects created by the constructor function.

Inheritance via Prototype Chain Example

```
<script>
function employee() //parent
{

}
employee.prototype.organization="ABC";
var em =  new employee();
document.write(em.organization);
function manager() //child
{

}
manager.prototype = employee.prototype;
```

Inheritance via Prototype Chain Example

```
manager.prototype.name = "John ";

var mg =  new manager();
document.write(mg.organization);
document.write(mg.name);

/*parent employee gets name property,
bcoz child type manager has name property */
/* which is to be shared prototype
reference between two objects */

document.write(em.name);
</script>
```

Inheritance Example - 2

```
<script>
var person = function()
  {
   this.name = "Mac";
   }
 person.prototype.city = "Banglore";

var student =function ()
{
 this.course = " Java ";
}

student.prototype = new person();

var  data = new student();
document.write(data.name);
document.write(data.city);
document.write(data.course);
</script>
```

The Object.create()

- The Object.create() creates a new object with the specified prototype object and properties.
- The Object.create() defined by ECMAScript 5.
- The Object.create() method only uses the prototype and not the constructor.

Syntax :

Object.create(proto)

Object.create(proto, descriptors)

The Object.create()

```
<script>
var car = {
     color : "green",
     getInfo   : function ()
     {
          alert(this.color);
     }
};
/* The object car passed to Object.create()
 is used as the prototype for new object.*/
var car1 = Object.create(car);
car1.color = "blue";
alert(car.getInfo());
alert(car1.getInfo());
</script>
```

Exercise

- Write a JavaScript program to create object book

  a) Add the property book name, author name

  b) Add the prototype property price .

  c) Display all the properties.

- Write a JavaScript program to create Parent object employee (Property : Employee Name , Employee Id , Salary) and Child object Manager (Property : Manager Name , Branch). Inherit all the properties of employee and display all the properties.

WSA | Forward looking IT finishing school

WebStack Academy

#83, Farah Towers,
1st Floor, MG Road,
Bangalore – 560001

M: +91-809 555 7332
E: training@webstackacademy.com

WSA in Social Media:

www.webstackacademy.com