

INDEX 20

Javascript objects

Javascript is an object oriented programming language.OOP provides four basic capabilities; Abstraction, Inheritance, Encapsulation and Polymorphism.

IN OOP, an object is defined as people, person,thing etc that exists in the real world.

Every object has 2 fundamental characteristics;

- a) The collection of properties
- b) Set of behaviors

For example: car is an object;

Which has properties like make,model, year,color,plate_number etc.

Set of behaviors are start(), run(), stop(), break() , turn() reverse() etc.

A class is defined as the collection of similar types of objects .Therefore also known as blueprint or template whereas an object is the instance of a class.

In javascript, users can define their own set of objects also known as user-defined objects.

How to write javascript program using object Based programming

There are 3 ways to create an object in javascript.

1. [By object literal](#)

2. [By creating instance of an object \(using new keyword\)](#)
3. [By using an object constructor](#)

1. By object literal

An object literal is a comma separated list of paired values enclosed in curly braces.

The syntax for creating object literal is

```
Var object={  
    property1:value1,  
    property2:value2.....propertyN:valueN  
};
```

Example Program:

```
<html>  
<head>  
    <title>js object literal</title>  
    <h1>Compiled by Er.Gaurab Mishra</h1>  
</head>  
<body>  
<script type="text/javascript">  
    var car={  
        make:'Toyota',  
        model:'RAV4',
```

```
        year:2022,  
        number:"NEPAL-2345"  
    };  
    document.write(car.make + " " + car.model + " " + car.year + " " + car.number)  
</script>  
</body>  
</html>
```

Accessing javascript object properties and methods

```
<html>  
<head>  
    <title>Access js object properties and methods</title>  
    <h1>Compiled by Er.Gaurab Mishra</h1>  
</head>  
<body>  
<script type="text/javascript">  
var car={  
    make:'Toyota',  
    model:'RAV4',  
    year:2022,  
    number:"NEPAL-2345",  
    display:function(){  
        return this.make + " " + this.model + " " + this.year + " " + this.number;  
    }  
};
```

```
        }  
    };  
    document.write(car.display());  
</script>  
</body>  
</html>
```

2. By creating an instance of an object (using new keyword)

Instance refers to the copy of the object.

The syntax for creating an object directly is given below:

Var objectname=new Object();

Here a new keyword is used to create a new object and assigns that property to the created object's name.

Example Program:

```
<html>  
<head>  
    <title>By creating instance of an object</title>  
    <h1>Compiled by Er.Gaurab Mishra</h1>  
</head>
```

```
<body>
<script type="text/javascript">
    var car=new Object();
    car.make='Toyota';
    car.model='RAV4';
    car.year=2022;
    car.number="NEPAL-2345";
    document.write(car.make + " " + car.model
        + " " + car.year + " " + car.number)
</script>
</body>
</html>
```

3. By using an object constructor

It creates an object wrapper for given values.

The function name is the same name as we have created for the object using 'new'. It means use the object as a function.

The this keyword refers to the current object.

Syntax:

Var object1=new functionname(value1,value2....);

Here object1 is an object created with a new keyword and function name is the function used by the programmer.

We will be using the same name for the function as created with the new keyword. That will be called as object constructor function.

Example Program:

```
<html>
<head>
  <title>By using an object constructor</title>
  <h1>Compiled by Er.Gaurab Mishra</h1>
</head>
<body>
<script type="text/javascript">
function emp();//emp is an object constructor function
{
  this.id=12;
  this.name='sameer';
  this.age=27;
}
var e=new emp();//object created with new
  document.write(e.id + " " + e.name
    + " " + e.age)

</script>
</body>
```

</html>

Function() constructor

IN oop, a constructor is a special type of function which helps to create an object. It creates a new object with some initial values for respective member variables.

The keywords new Function() are used to define a new constructor

The normal user-defined function is defined by using the keyword function whereas the constructor is defined using new Function()

Syntax:

```
var variable_name=new Function(arg1,arg2,.....,"Function Body");
```

Example program

<html>

<head>

<title>js function constructor</title>

<h1>Compiled by Er.Gaurab Mishra</h1>

```
</head>
<body>
<script type="text/javascript">
  var area=new Function("l","b","return l*b");
    function display()
    {
      var result;
      result=area(10,20);
      document.write(result);
    }
</script>
<button type="button" onclick="display()">Click me</button>
</body>
</html>
```

Array objects

An array can also be defined as an object in js. An array is the collection of similar types of data treated as a single unit.

Each of the items of the array can be accessed using array index which starts from 0 and ends with arraysize-1, where size defines the length of an array.

An array is also defined by using the keyword new Array().

Syntax:

```
var array_name=new Array();
```

Example program:

```
<html>
<head>
    <title>js array objects</title>
    <h1>Compiled by Er.Gaurab Mishra</h1>
</head>
<body>
<script type="text/javascript">
    var cars=new Array(5);
    cars=['maruti','toyota','bmw','szk','abc'];
    document.write(cars);//displays all
    document.write("<br>");
    document.write(cars.pop() + "<br>");
    document.write(cars + "<br>");
    document.write(cars.push('xyz') + "<br>");
    document.write(cars.reverse() + "<br>");
    document.write(cars.sort() + "<br>");
</script>
</body>
</html>
```

String objects

A string is defined as an array of characters .String automatically inserts a null character '\0' at the end of the string.

Syntax:

```
var string_name=new String(string);
```

Example program

```
<html>
<head>
    <title>js string objects</title>
    <h1>Compiled by Er.Gaurab Mishra</h1>
</head>
<body>
<script type="text/javascript">
    var str=new String("Hello");
    document.write(str.length + "<br>");
    document.write(str.toLowerCase() + "<br>");
    document.write(str.toUpperCase() + "<br>");
    document.write(str.charAt(2) + "<br>");
</script>
</body>
```

</html>

Math objects

The math object allows you to perform mathematical tasks. The Math object includes several mathematical constants and methods.

Example Program

```
<html>
<head>
  <title>js Math objects</title>
  <h1>Compiled by Er.Gaurab Mishra</h1>
</head>
<body>
<script type="text/javascript">
  var pi_value=Math.PI;
  var sqrt=Math.sqrt(16);
  var round_value=Math.round(2.6);
  var random_value=Math.random();
  var floor_value=Math.floor(4.7);
  var ceil_value=Math.ceil(2.3);
  var max_value=Math.max(2,3,6,7,9);
```

```
var min_value=Math.min(56,89,4,7,8,2);
document.write(pi_value+"<br>");
document.write(sqrt+"<br>");
document.write(round_value+"<br>");
document.write(random_value+"<br>");
document.write(floor_value+"<br>");
document.write(ceil_value+"<br>");
document.write(max_value+"<br>");
document.write(min_value+"<br>");
</script>
</body>
</html>
```

Date Objects

The date object is used to work with date and times.

The Date object is used to work with dates and times.

You create an instance of the Date object with the "new" keyword.

To store the current date in a variable called "my_date":

```
var my_date=new Date()
```

Example Program

```
<html>
<head>
    <title>js Date objects</title>
    <h1>Compiled by Er.Gaurab Mishra</h1>
</head>
<body>
<script type="text/javascript">
var my_date=new Date();
document.write(my_date + "<br>");
//returns the day of the month(1-31)
document.write(my_date.getDate()+"<br>");
//returns the day of the week(0-6)
document.write(my_date.getDay()+"<br>");
//returns the year(4 digits)
document.write(my_date.getFullYear()+"<br>");
//returns the month ( 0-11)
document.write(my_date.getMonth()+"<br>");
//Returns the hour (0-23)
document.write(my_date.getHours()+"<br>");
//returns milliseconds(0-999)
```

```
document.write(my_date.getMilliseconds()+"<br>");  
//returns minutes ( 0-59)  
document.write(my_date.getMinutes()+"<br>");  
//returns seconds (0-59)  
document.write(my_date.getSeconds()+"<br>");  
</script>  
</body>  
</html>
```