Chapter - 7

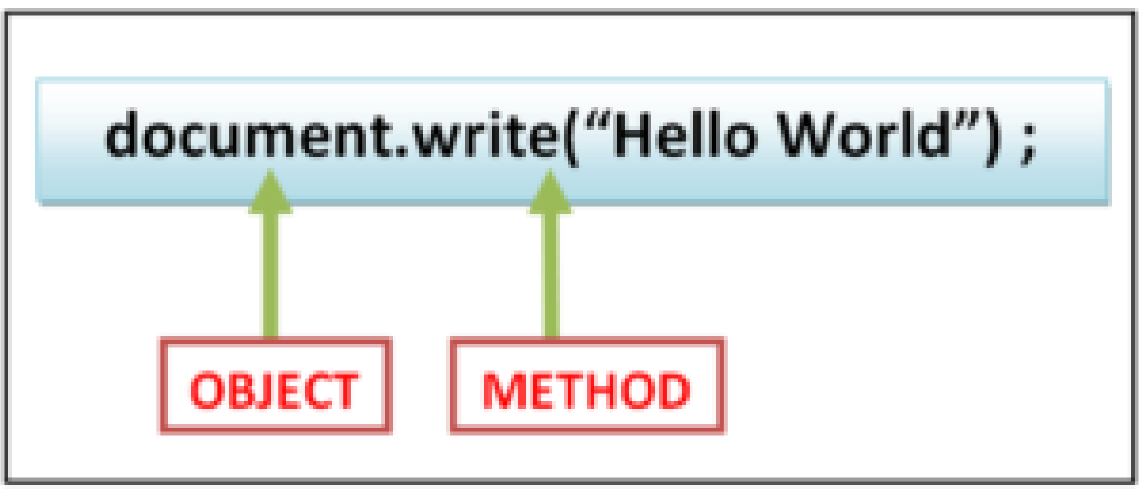
जावास्क्रिप्ट में ऑब्जेक्ट्स क्या है (What is Objects in JavaScript)

जावास्क्रिप्ट में लगभग सभी Elements Objects कहलाते हैं |ऑब्जेक्ट किसी स्पेशल टाइप के Data या Variable होते है जिसकी Property और Method होती है | जावास्क्रिप्ट में हम जो भी इन्सर्ट करते हैं जैसे – Array, Functions, Date, Maths आदि सभी Objects के अंतर्गत आते हैं |हम हमेशा जावास्क्रिप्ट में स्क्रिप्टिंग करते समय OBJECTS का इस्तेमाल करते ही हैं|

साधारण अर्थो में जावास्क्रिप्ट में सब कुछ ऑब्जेक्ट होता है। एक Object कुछ Properties और Methods को एक जगह Bind करने के लिए प्रयोग किया जाता है। ये Properties और Methods किसी Single Entity को Represent करती है। जावास्क्रिप्ट आपको built in objects provide करती है। जैसे की JavaScript में Strings Objects है। आप कोई भी String Variable Create करके उससे Related Properties और Methods प्रयोग कर सकते है।

जैसे – document.write("Hello World") ;

यहाँ document एक object है और write() एक method उपरोक्त उदाहरण के आधार पर हम यह कह सकते हैं कि मेथड को एक्सेस करने के लिए ऑब्जेक्ट का प्रयोग किया जाता है |



Creating Objects in JavaScript

अधिकतर प्रोग्रामिंग लैंग्वेज में ऑब्जेक्ट्स एक ही तरीके से क्रिएट किये जाते है। जावास्क्रिप्ट में आप ऑब्जेक्ट्स तीन तरह से क्रिएट कर सकते है।

- Creating Objects in Single Statement
- Creating Objects with New Keyword
- Creating Objects Using Constructor

First Method

Creating Objects in Single Statement

पहले तरीके में आप Object का नाम लिखते है और Curly Brackets में Properties और उनकी Values को Colon से Separate करते है। आप चाहे तो JavaScript में Objects किसी Variables की तरह एक Single Statement में Create कर सकते है।

Example :
var obj1 = {Name:"yourName",Age:24};

इस उदाहरण में Object किसी Variable की तरह Create किया गया है। Curly brackets में Properties के नाम और Values को Colon से Separate किया गया है। और हर Pair को Comma (,) से Separate किया गया है। किसी भी Property की Value को Access Object के नाम के आगे dot (.) Operator लगाकर और फिर Property का नाम लिखकर Create कर सकते है।

document.write(obj1.Name);

Single Statement में Object Create करना सबसे आसान तरीका है।

Second Method

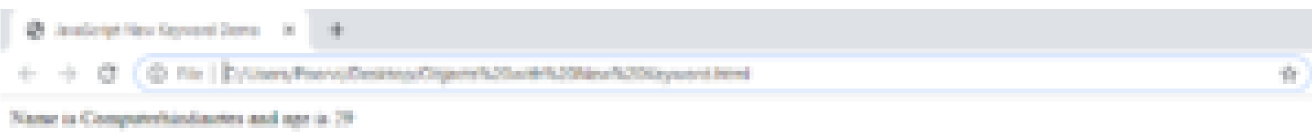
Creating Objects with New Keyword

दूसरे तरीके में New Keyword के द्वारा एक Object instance create किया जा सकता है।इस तरीके में पहले New Keyword के साथ Object Create कर लिया जाता है। बाद में Object के नाम के आगे dot (.) Operator लगाकर Property का नाम और Values Define की जाती है। अधिकतर Programming languages में इसी तरीके से Objects Create किये जाते है। इसका उदाहरण नीचे दिया जा रहा है।

```
<html>
<head>
<title>JavaScript New Keyword Demo</title>
</head>
<body>

<script type="text/javascript">
// Creating object
```

```
var obj1 = new Object();
// Creating object variables

obj1.Name=”computerhindinotes”;
obj1.Age=29;
// Printing object variables
document.write(“Name is “+Name+”and age is “+Age);
</script>

</body>
</html>
```

Output



Third Method

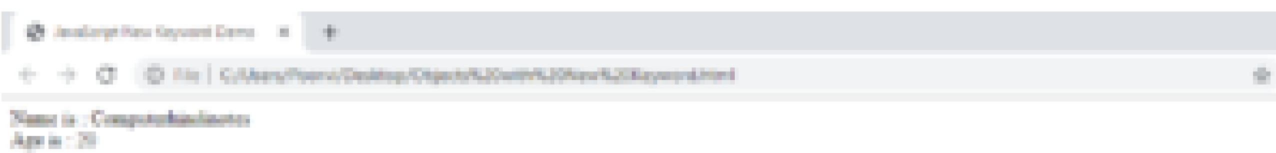
Creating Objects Using Constructor

तीसरे तरीके में object constructor प्रयोग किया जाता है। इन सभी तरीकों के बारे में निचे detail से दिया जा रहा है। JavaScript में Object Create करने का ये तरीका थोड़ा जटिल (Complicated) है। इस तरीके में एक Function Create किया जाता है जो Properties की Values को Argument की तरह लेता है। ये Function C language के किसी Structure या C++ की Classes की तरह काम करता है और Object Create होने के लिए एक Structure Provide करता है।

इस फंक्शन का नाम ऑब्जेक्ट के नाम जैसा ही होता है। जब New Keyword के साथ Object Create किया जाता है तो साथ ही Properties की Values भी Pass की जाती है। ये Values this keyword के द्वारा Original Properties को Apply की जाती है। इसका उदाहरण नीचे दिया जा रहा है।

```
<html>
<head>
<title>JavaScript Objects Demo</title>

<script type=”text/javascript”>
function Employee(name,age)

{
this.Name = name
this.Age = age;
}
</script>

</head>
<body>

<script type=”text/javascript”>
// Creating object using constructor
var obj1 = new Employee(“Computerhindinotes”,20);
document.write(“Name is :”+obj1.Name+”<br>”);
document.write(“Age is :”+obj1.Age);
</script>

</body>
</html>
```

Output



Object Properties

- किसी भी जावास्क्रिप्ट ऑब्जेक्ट का सबसे महत्वपूर्ण हिस्सा Properties हैं। Properties जावास्क्रिप्ट ऑब्जेक्ट से जुड़ीं वैल्यू हैं। जावास्क्रिप्ट ऑब्जेक्ट Unordered Properties का एक संग्रह है।
- Properties आमतौर पर change, add और delete किये जा सकते हैं, लेकिन कुछ केवल Read किये जाते हैं।
- जावास्क्रिप्ट ऑब्जेक्ट्स में Named Value को Properties कहा जाता हैं।

Property	Value
firstName	Rohit
lastName	Sharma
age	50
eyeColor	blue

Add New Properties in JavaScript

- आप किसी मौजूदा Object को केवल एक Value देकर नई Properties जोड़ सकते हैं।
- मान लें कि person object पहले से मौजूद है – आप फिर उसे नई Properties दे सकते हैं:

Example – person.nationality = “English”;

Delete Properties in JavaScript

डिलीट कीवर्ड किसी प्रॉपर्टी को किसी ऑब्जेक्ट से हटा देता है:

Example – var person = {firstName:”Rohit”, lastName:”Sharma”, age:50, eyeColor:”blue”};

delete person.age; // or delete person[“age”];

- डिलीट कीवर्ड प्रॉपर्टी की वैल्यू और प्रॉपर्टी दोनों को ही डिलीट कर देता है।
- हटाए जाने के बाद, प्रॉपर्टी को दोबारा वापस जोड़ने से पहले इसका इस्तेमाल नहीं किया जा सकता है।
- हटाए गए ऑपरेटर को ऑब्जेक्ट प्रॉपर्टी पर उपयोग करने के लिए डिज़ाइन किया गया है। इसका वेरिएबल या फंक्शन पर कोई प्रभाव नहीं पड़ता है।
- हटाए गए ऑपरेटर को पूर्वनिर्धारित जावास्क्रिप्ट ऑब्जेक्ट प्रॉपर्टी पर उपयोग नहीं किया जाना चाहिए। यह आपके एप्लिकेशन को क्रैश कर सकता है।

Property Attributes in JavaScript

- सभी प्रॉपर्टीज का एक नाम है। इसके अलावा उनकी एक वैल्यू भी है।
- वैल्यू, प्रॉपर्टी की attributes में से एक है।
- अन्य attributes हैं: enumerable, configurable, और writable|

- ये attributes परिभाषित करती हैं कि प्रॉपर्टी तक कैसे पहुँचा जा सकता है (क्या यह readable है ?, क्या यह writable है?)
- जावास्क्रिप्ट में, सभी attributes को पढ़ा जा सकता है, लेकिन केवल वैल्यू एट्रिब्यूट को बदला जा सकता है

Prototype Properties in JavaScript

जावास्क्रिप्ट ऑब्जेक्ट्स उनके प्रोटोटाइप की प्रॉपर्टी को प्राप्त करते हैं। डिलीट कीवर्ड inherited properties को नहीं हटाता है, लेकिन यदि आप एक प्रोटोटाइप प्रॉपर्टी हटाते हैं, तो यह प्रोटोटाइप से object inherited को प्रभावित करेगा।

JavaScript Object Methods

Example –

```
var person = {
firstName: “Rohit”,
lastName : “Sharma”,
id : 4488,
fullName : function() {
return this.firstName + ” ” + this.lastName;
}
};
```

फ़ंक्शन की परिभाषा में, this फ़ंक्शन के “owner” को संदर्भित करता है।

उपरोक्त उदाहरण में, this वह पर्सनल ऑब्जेक्ट है जो Fullname फ़ंक्शन का “owns” है।

दूसरे शब्दों में, this.firstName का अर्थ है firstName की प्रॉपर्टी this object है|

JavaScript Methods

- जावास्क्रिप्ट मेथड्स ऐसी क्रियाएं हैं जो ऑब्जेक्ट्स पर की जा सकती हैं।
- जावास्क्रिप्ट मेथड एक प्रॉपर्टी है जिसमें फ़ंक्शन परिभाषा होती है।

Property	**Value**
firstName	Rohit
lastName	Sharma
age	50

eyeColor	blue
fullName	function() {return this.firstName + " " + this.lastName;}

Using Built-In Methods

किसी टेक्स्ट को अपरकेस में कनवर्ट करने के लिए स्ट्रिंग ऑब्जेक्ट के toUpperCase() मेथड का उपयोग करता है:

Var message = "Hello world!";
var x = message.toUpperCase();
उपरोक्त कोड के Execution के बाद x का मान होगा:
HELLO WORLD!
Adding a Method to an Object

ऑब्जेक्ट में एक नई मेथड जोड़ना आसान है:

Example –
person.name = function () {
return this.firstName + " " + this.lastName;
};