জাভাস্ক্রিপ্টে শুধু এক টাইপের নম্বর আছে।

নম্বর দশমিক অথবা দশমিক ছাড়া লিখা যায়।

**জাভাস্ক্রিপ্ট নম্বর**

জাভাস্ক্রিপ্ট নম্বর দশমিক অথবা দশমিক ছাড়া লিখা যায়ঃ

var a = 105.24; // দশমিকসহ নম্বর

var b = 105; // দশমিক ছাড়া নম্বর

Copy

অতিরিক্ত বড় অথবা ছোট নম্বরকে বৈজ্ঞানিক গানিতিক সূচক আকারে লিখা যায়ঃ

var a = 324e4; // 3240000

var b = 324e-4; // 0.0324

Copy

**জাভাস্ক্রিপ্ট নম্বরগুলো সবসময় ৬৪-বিট ফ্লোটিং পয়েন্ট**

অন্যান্য প্রোগ্রামিং ল্যাংগুয়েজের মত, জাভাস্ক্রিপ্ট বিভিন্ন ধরনের নম্বর ডিফাইন করে না, যেমন integers, short, long, float ইত্যাদি।

জাভাস্ক্রিপ্ট নম্বরগুলো সব সময় ডাবল প্রিসিশন দশমিক বিন্দু নম্বর হিসেবে সংরক্ষিত থাকে, এই ক্ষেত্রে এটি international IEEE 754 স্ট্যান্ডার্ড অনুসরন করে।   
  
এই ফরম্যাট ৬৪ বিট নম্বর হিসেবে সংরক্ষিত থাকে, যখন নম্বরগুলো (ভগ্নাংশ) ০ থেকে ৫১ বিটে সংরক্ষিত থাকে,সূচক ৫২ থেকে ৬২ বিটে থাকে, এবং প্রতীক ৬৩ বিটে থাকেঃ

|  |  |  |
| --- | --- | --- |
| **ভ্যালু (aka Fraction/Mantissa)** | **সূচক** | **চিহ্ন** |
| ৫২ বিট (০ - ৫১) | ১১ বিট (৫২ - ৬২) | ১ বিট (৬৩) |

**নির্ভুলতা**

ইন্টিজারগুলো (নম্বরগুলো একটি সীমা ছাড়া অথবা গানিতিক সূচক) ১৫ সংখ্যা পর্যন্ত নির্ভুল বিবেচনা করে।

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>ইন্টিজারগুলো ১৫ সংখ্যা পর্যন্ত

নির্ভুল বিবেচনা করে। </h4>

<button onclick="myFunc()">নিজে চেষ্টা করি </button>

<p id="test"></p>

<script>

function myFunc() {

var a = 999999999999999;

var b = 9999999999999999;

document.getElementById("test").innerHTML = a + "<br>" + b;

}

</script>

</body>

</html>

Copy

**ফলাফল**

#### ইন্টিজারগুলো ১৫ সংখ্যা পর্যন্ত নির্ভুল বিবেচনা করে।

999999999999999  
10000000000000000

দশমিক সংখ্যার সর্বোচ্চ নম্বর হলো ১৭, কিন্তু দশমিক সংখ্যার বীজগনিত সব সময় ১০০% সঠিক হয় নাঃ

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>দশমিক সংখ্যার বীজগনিত সব সময় ১০০% সঠিক হয় না।

</h4>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

function myFunc() {

var a = 0.2 + 0.1;

document.getElementById("test").innerHTML = "0.2 + 0.1 = " + a;

}

</script>

</body>

</html>

Copy

**ফলাফল**

#### দশমিক সংখ্যার বীজগনিত সব সময় ১০০% সঠিক হয় না।

0.2 + 0.1 = 0.30000000000000004

উপরের সমস্যা সমাধানের জন্য, ভাগ এবং গুন এক্ষেত্রে সহযোগিতা করে:

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>দশমিক সংখ্যার বীজগনিত সব সময় ১০০% সঠিক হয় না।</h4>

<h5>কিন্তু এটি আপনাকে গুণ এবং ভাগ করতে সাহায্য করে।

</h5>

<button onclick="myFunc()"> নিজে চেষ্টা করি </button>

<p id="test"></p>

<script>

function myFunc() {

var a = (0.3\*10 + 0.1\*10) / 10;

document.getElementById("test").innerHTML = "0.3 + 0.1 = " + a;

}

</script>

</body>

</html>

Copy

**ফলাফল**

#### দশমিক সংখ্যার বীজগনিত সব সময় ১০০% সঠিক হয় না।

##### কিন্তু এটি আপনাকে গুণ এবং ভাগ করতে সাহায্য করে।

0.3 + 0.1 = 0.4

**হেক্সাডেসিমেল**

জাভাস্ক্রিপ্ট সংখ্যা ধ্রুবকগুলোকে হেক্সাডেসিমেল হিসেবে প্রকাশ করে যদি তাদের পুর্বে 0x থাকে।

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>জাভাস্ক্রিপ্ট সংখ্যা ধ্রুবকগুলোকে হেক্সাডেসিমেল হিসেবে প্রকাশ করে যদি তাদের পুর্বে 0x থাকে।</h4>

<button onclick="myFunc()">নিজে চেষ্টা করি </button>

<p id="test"></p>

<script>

function myFunc() {

document.getElementById("test").innerHTML = "0xFF = " + 0xFF;

}

</script>

</body>

</html>

Copy

**ফলাফল**

#### জাভাস্ক্রিপ্ট সংখ্যা ধ্রুবকগুলোকে হেক্সাডেসিমেল হিসেবে প্রকাশ করে যদি তাদের পুর্বে 0x থাকে।

0xFF = 255

কখনো নম্বরের শুরুতে ০ দিবে না (যেমন ০৭)।  
যদি সামনে ০ থাকে তাহলে কিছু জাভাস্ক্রিপ্ট ভার্সন সংখ্যাগুলোকে অক্টাল হিসেবে ব্যাখ্যা করে।

জাভাস্ক্রিপ্ট নম্বরগুলোকে ১০ ভিত্তিক ডেসিমেল নম্বর হিসেবে প্রদর্শন করে।

কিন্তু আপনি toString() মেথড ব্যবহার করে নম্বরগুলোকে ১৬ ভিত্তিক (হেক্স), ৮ ভিত্তিক(অক্টাল), অথবা ২ ভিত্তিক(বাইনারি) হিসেবে আউটপুট নিতে পারেন।

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>আপনি toString() মেথড ব্যবহার করে নম্বরগুলোকে ১৬ ভিত্তিক (হেক্স), ৮ ভিত্তিক(অক্টাল), অথবা ২ ভিত্তিক(বাইনারি) হিসেবে আউটপুট নিতে পারেন।</h4>

<p id="test"></p>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<script>

function myFunc() {

var scores = 214;

document.getElementById("test").innerHTML = "214 = " +

scores + " Decimal, " +

scores.toString(16) + " Hexadecimal, " +

scores.toString(8) + " Octal, " +

scores.toString(2) + " Binary."

}

</script>

</body>

</html>

Copy

**ফলাফল**

#### আপনি toString() মেথড ব্যবহার করে নম্বরগুলোকে ১৬ ভিত্তিক (হেক্স), ৮ ভিত্তিক(অক্টাল), অথবা ২ ভিত্তিক(বাইনারি) হিসেবে আউটপুট নিতে পারেন।

214 = 214 Decimal, d6 Hexadecimal, 326 Octal, 11010110 Binary.

**অসীম**

অসীম (অথবা -অসীম) এমন একটি মান আপনি যদি সম্ভাব্য বড় নম্বরের বাইরে হিসেব করেন তাহলে জাভাস্ক্রিপ্ট একটি মান রিটার্ন করবে।

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> ইনফিনিটি সম্ভবপর সর্বোচ্চ সংখ্যা পর্যন্ত হিসাব করে একটি সংখ্যা রিটার্ন করে। </h4>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

function myFunc() {

var scores = 4;

var text = "";

while (scores != Infinity) {

scores = scores \* scores;

text = text + scores + "<br>";

}

document.getElementById("test").innerHTML = text;

}

</script>

</body>

</html>

Copy

**ফলাফল**

#### ইনফিনিটি সম্ভবপর সর্বোচ্চ সংখ্যা পর্যন্ত হিসাব করে একটি সংখ্যা রিটার্ন করে।

16  
256  
65536  
4294967296  
18446744073709552000  
3.402823669209385e+38  
1.157920892373162e+77  
1.3407807929942597e+154  
Infinity

০(শূণ্য) দ্বারা ভাগ করলে ইনফিনিটি মান প্রদর্শন করবে :

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>শূন্য দ্বারা ভাগও ইনফিনিটি উৎপন্ন করে।</h4>

<button onclick="myFunc()">নিজে চেষ্টা করি</button>

<p id="test"></p>

<script>

function myFunc() {

var a = 5/0;

var b = -5/0;

document.getElementById("test").innerHTML = a + "<br>" + b;

}

</script>

</body>

</html>

Copy

**ফলাফল**

#### শূন্য দ্বারা ভাগও ইনফিনিটি উৎপন্ন করে।

Infinity  
-Infinity

typeOf ইনফিনিটি একটি নম্বর রিটার্ণ করে।

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>ইনফিনিটি হচ্ছে একটি সংখ্যা।</h4>

<button onclick="myFunc()">নিজে চেষ্টা করি </button>

<p id="test"></p>

<script>

function myFunc() {

document.getElementById("test").innerHTML = typeof Infinity;

}

</script>

</body>

</html>

Copy

**ফলাফল**

#### ইনফিনিটি হচ্ছে একটি সংখ্যা।

number

**NaN - Not a Number**

NaN একটি জাভাস্ক্রিপ্টের রিজার্ভ শব্দ যা "একটি নম্বর নয়" এটি নির্দেশ করে।

সংখ্যা ছাড়া বীজগনিত করার চেষ্টা করলে ফলাফল NaN আসবে।(Not a Number)

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> একটি সংখ্যা অ সাংখ্যিক স্ট্রিং দ্বারা বিভক্ত (NaN-একটি নম্বর না) হয়ঃ </h4>

<p id="test"></p>

<script>

document.getElementById("test").innerHTML = 123 / "Lion";

</script>

</body>

</html>

Copy

**ফলাফল**

#### একটি সংখ্যা অ সাংখ্যিক স্ট্রিং দ্বারা বিভক্ত (NaN-একটি নম্বর না) হয়ঃ

NaN

যাই হোক, যদি স্ট্রিং সাংখ্যিক মান ধারন করে তাহলে ফলাফল নম্বর প্রদর্শন করেঃ

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>একটি সংখ্যা সাংখ্যিক স্ট্রিং দ্বারা বিভক্ত হয়ে একটি সংখ্যা হয়ঃ

</h4>

<p id="test"></p>

<script>

document.getElementById("test").innerHTML = 123 / "7";

</script>

</body>

</html>

Copy

**ফলাফল**

#### একটি সংখ্যা সাংখ্যিক স্ট্রিং দ্বারা বিভক্ত হয়ে একটি সংখ্যা হয়ঃ

17.571428571428573

যদি মান একটি নম্বর হয় তাহলে আপনি গ্লোবাল জাভাস্ক্রিপ্ট ফাংশন isNaN() ব্যবহার করবেন।

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var a = 123 / "Lion";

document.getElementById("test").innerHTML = isNaN(a);

</script>

</body>

</html>

Copy

**ফলাফল**

true

NaN এর জন্য। আপনি যদি NaN কে একটি গাণিতিক অপারেশণ হিসেবে লিখেন,তাহলেও ফলাফল NaN আসবে:

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> আপনি একটি গাণিতিক অপারেশন মধ্যে NaN ব্যবহার করা হলে, ফল এছাড়াও NaN হবেঃ</h4>

<p id="test"></p>

<script>

var a = NaN;

var b = 8;

document.getElementById("test").innerHTML = a + b;

</script>

</body>

</html>

Copy

**ফলাফল**

#### আপনি একটি গাণিতিক অপারেশন মধ্যে NaN ব্যবহার করা হলে, ফল এছাড়াও NaN হবেঃ

NaN

 অথবা ফলাফল যুক্ত হতে পারে:

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> আপনি একটি গাণিতিক অপারেশন মধ্যে nan ব্যবহার করার ফলে এটি কনকাটিনেশন হতে পারেঃ</h4>

<p id="test"></p>

<script>

var a = NaN;

var b = "8";

document.getElementById("test").innerHTML = a + b;

</script>

</body>

</html>

Copy

**ফলাফল**

#### আপনি একটি গাণিতিক অপারেশন মধ্যে nan ব্যবহার করার ফলে এটি কনকাটিনেশন হতে পারেঃ

NaN8

NaN একটি নম্বর এবং typeof NaN নম্বর রিটার্ণ করেঃ

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>typeof NaN হয়ঃ</h4>

<p id="test"></p>

<script>

document.getElementById("test").innerHTML = typeof NaN;

</script>

</body>

</html>

Copy

**ফলাফল**

#### typeof NaN হয়ঃ

number

**নম্বর অবজেক্ট হতে পারে**

সাধারণত জাভাস্ক্রিপ্ট নম্বরগুলো মৌলিক মান হয়ঃ **var a = 324**

যখন keyword new থাকবে তখন নম্বরগুলো অবজেক্ট হিসেবে নির্ধারন করেঃ **var y = new Number(324)**

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<p id="test"></p>

<script>

var a = 324;

var b = new Number(324);

document.getElementById("test").innerHTML = typeof a + "<br>" + typeof b;

</script>

</body>

</html>

Copy

**ফলাফল**

number  
object

নম্বর অবজেক্ট তৈরি করবেন না। এটি এক্সিকিউশন স্পিড ধীরগতির করে দেয়।  
**new** কি-ওয়ার্ড কোডকে জটিল করে তোলে। এটি অনেক সময় অপ্রত্যাশিত ফলাফল তৈরি করেঃ

যখন == (equality) অপারেটর ব্যবহার করা হয়, তখন নম্বরগুলো সমান দেখায়:

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4> কখনো অবজেক্ট নাম্বার হিসেবে তৈরি হয় না। </h4>

<h4> নাম্বার এবং অবজেক্ট নিরাপদে তুলনা করা যাবে না।</h4>

<p id="test"></p>

<script>

var a = 324; // a is a number

var b = new Number(324); // b is an object

document.getElementById("test").innerHTML = (a==b);

</script>

</body>

</html>

Copy

**ফলাফল**

#### কখনো অবজেক্ট নাম্বার হিসেবে তৈরি হয় না।

#### নাম্বার এবং অবজেক্ট নিরাপদে তুলনা করা যাবে না।

true

যখন === equality অপারেটর ব্যবহার করা হয়,তখন equal নম্বরগুলো সমান নয়, কারন === অপারেটর ধরন এবং মান দুইটাই সমান প্রকাশ করে।

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>কখনো অবজেক্ট নাম্বার হিসেবে তৈরি হয় না।</h4>

<h4>নাম্বার এবং অবজেক্ট নিরাপদে তুলনা করা যাবে না।</h4>

<p id="test"></p>

<script>

var a = 324; // a is a number

var b = new Number(324); // b is an object

document.getElementById("test").innerHTML = (a===b);

</script>

</body>

</html>

Copy

**ফলাফল**

#### কখনো অবজেক্ট নাম্বার হিসেবে তৈরি হয় না।

#### নাম্বার এবং অবজেক্ট নিরাপদে তুলনা করা যাবে না।

false

এর একটি মন্দ দিক হচ্ছে অবজেক্ট তুলনা করতে পারে নাঃ

**উদাহরণ**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>জাভাস্ক্রিপ্ট উদাহরণ</title>

</head>

<body>

<h4>কখনো অবজেক্ট নাম্বার হিসেবে তৈরি হয় না। </h4>

<h4>জাভাস্ক্রিপ্ট অবজেক্ট তুলনাযোগ্য নয়। </h4>

<p id="test"></p>

<script>

var a = new Number(324); // a is a object

var b = new Number(324); // b is an object

document.getElementById("test").innerHTML = (a==b);

</script>

</body>

</html>

Copy

**ফলাফল**

#### কখনো অবজেক্ট নাম্বার হিসেবে তৈরি হয় না।

#### জাভাস্ক্রিপ্ট অবজেক্ট তুলনাযোগ্য নয়।

false