JavaScript中的对象是有着属性和方法的一种特殊数据类型。

常见的对象有数字Number，字符串String，日期Date，数组Array等。

注意：这里讲的Numbr是对象Number，和基本数据类型中的基本类型Number是不一样的。



1. 创建一个数字对象（new Number()）

可以通过new Number()创建一个数字对象。

与基本类型的数字不同，对象类型的数字，拥有更多的属性和方法。

<script>

var x = new Number(123);

document.write('数字对象x的值:' + x);

document.write("<br>");

document.write('数字对象x的类型:' + typeof x); //通过typeof 获知这是一个object

document.write("<br>");

var y = 123;

document.write('基本类型y的值:' + y);

document.write("<br>");

document.write('基本类型y的类型:' + typeof y); //通过typeof 获知这是一个number

</script>

1. 最小值 最大值（MIN\_VALUE、MAX\_VALUE）

Number对象能够取到的最大和最小边界：

Number对象的最小值:5e-324

Number对象的最大值:1.7976931348623157e+308

<script>

document.write('Number对象的最小值：' + Number.MIN\_VALUE);

document.write('<br />');

document.write('Number对象的最大值：' + Number.MAX\_VALUE);

</script>

1. 表示不是数字（isNaN()）

NaN（Not a Number）表示不是一个数字。

当通过非数字创建Number的时候，就会得到NaN。

注意：应该使用函数isNaN()来判断是否“不是一个数字”。

<script>

function p(s) {

document.write(s);

document.write('<br />');

}

var a = new Number("123abc");

p('通过非数字字符串“123abc”创建出来的Number对象a的值是：' + a);

p('但是，a==Number.NaN会返回：' + (a == Number.NaN));

p('正确判断是否是NaN的方式是调用isNaN函数：' + isNaN(a));

</script>

1. 返回一个数字的小数表达（toField()）

toField返回一个数字的小数表达式。

<script>

function p(s) {

document.write(s);

document.write('<br />');

}

var a = new Number(123);

p('数字对象123通过toFiled(2)保留两位小数：' + a.toFixed(2));

var b = new Number(3.1415926);

p('PI通过toFiled(3)保留三位小数：' + b.toFixed(3));

</script>

1. 返回一个数字的科学计数表达（toExponential()）

返回一个Number对象的科学计数法表达。

<script>

function p(s) {

document.write(s);

document.write('<br />');

}

var a = new Number(123);

p('数字对象123通过toExponential返回计数法表达：' + a.toExponential());

var b = new Number(3.1415926);

p('数字对象3.1415926通过toExponential返回计数法表达：' + b.toExponential());

</script>

1. 返回一个数字对象的基本数字类型（valueOf()）

方法valueOf()返回一个基本类型的数字。

通过typeOf()判断数据类型可以发现，一种是object，一种是number。

<script>

function p(s) {

document.write(s);

document.write('<br />');

}

var a = new Number(123);

var b = a.valueOf();

p('数字对象a的类型是：' + typeof a); //返回object

p('通过valueOf()返回的值的类型是' + typeof b); //返回number

</script>

1. 乘法计算

<style type="text/css">

input {

width: 50px;

}

</style>

<script>

function calc() {

var num1 = document.getElementById("num1").value;

var num2 = document.getElementById("num1").value;

num1 = parseFloat(num1);

num2 = parseFloat(num2);

var result = num1 \* num2;

var num = new Number(result);

document.getElementById("result").value = num.toExponential();

}

</script>

<input type="text" id="num1" />\*

<input type="text" id="num2" /> =

<input type="text" id="result" />

<input type="button" value="运算" onclick="calc()" />