|  |  |
| --- | --- |
| **undefined** | 声明了但未赋值 |
| Boolean | 布尔 |
| Number | 数字 |
| String | 字符串 |
| **var** | 动态类型 |
| Typeof | 变量类型判断 |
| null | 空对象/对象不存在 |

1. **声明了但未赋值**

当一个变量被声明了，却没有赋值的时候，叫做undefined。

<script>

var x; //声明了变量x,但是没有赋值

document.write('声明了，但是没有赋值的变量 x: ' + x);

</script>

1. **布尔**

<script>

var x = true;

var y = false;

document.write("布尔值:" + x);

document.write("<br>");

document.write("布尔值:" + y);

</script>

1. **数字**

JavaScript中的Number可以表示十进制，八进制，十六进制整数，浮点数，科学计数法。

<script>

var a = 10; //十进制

var b = 012; //第一位是0，表示八进制

var c = 0xA; //0x开头表示十六进制

var d = 3.14; //有小数点表示浮点数

var e = 3.14e2; //使用e的幂表示科学计数法

document.write("十进制 10 的值: " + a);

document.write("<br>");

document.write("八进制 012 的值: " + b);

document.write("<br>");

document.write("十六进制 0xA 的值: " + c);

document.write("<br>");

document.write("浮点数 3.14 的值: " + d);

document.write("<br>");

document.write("科学记数法 3.14e2 的值: " + e);

document.write("<br>");

</script>

1. **字符串**

与Java不同，**JavaScript中没有字符的概念，只有字符串**，所以单引号和双引号，都用来表示字符串。

<script>

var x = 'hello '; //单引号

var y = "javascript"; //双引号

document.write("单引号的字符串:" + x);

document.write("<br>");

document.write("双引号的字符串:" + y);

</script>

1. **动态类型**

变量的类型是动态的，当值是整数的时候，就是Number类型，当值是字符串的时候，就是String类型。

<script>

var x = 10; //Number类型

document.write("此时 x的值是 " + x + " 类型是数字");

document.write("<br>");

x = "hello javascript"; //String类型

document.write("此时 x的值是 " + x + " 类型是字符串");

</script>

1. **变量类型判断**

**使用typeof来进行判断数据类型**

正是因为变量是动态类型的，所以无法确定当前到底是哪种类型，这个时候，就可以使用typeof来进行判断

<script>

var x;

document.write('声明了但是未赋值的时候，类型是： ' + typeof x);

document.write("<br>");

x = 5;

document.write('赋值为5之后，类型是： ' + typeof x);

document.write("<br>");

x = 5.1;

document.write('赋值为5.1之后，类型是： ' + typeof x);

document.write("<br>");

x = true;

document.write('赋值为true之后，类型是： ' + typeof x);

document.write("<br>");

x = "hello";

document.write('赋值为hello之后，类型是： ' + typeof x);

</script>

1. **空对象/对象不存在**

Null表示一个对象不存在，因为此处都是基本类型，而null是和对象相关的，所以会放在JavaScript中的对象中进行详解。