js字符串转换成数字与数字转换成字符串的实现方法_javascript技巧_脚本之家

js字符串转换成数字

将字符串转换成数字，得用到**parseInt**函数。

parseInt(string) ： 函数从string的开始解析，返回一个整数。

举例：

parseInt('123') : 返回 123（int）；

parseInt('1234xxx') : 返回 1234（int）；

如果解析不到数字，则将返回一个NaN的值，可以用isNaN()函数来检测；

举例 ：

```
var i = parseInt('abc');
if (isNaN(i))
{
alert('NaN value');
}
```

同样的parseFloat函数是将字符串转换成浮点数。

举例：parseFloat('31.24abc') : 返回 31.24;

js数字转换成字符串

将字符串转换成数字，得用到**String**类的**toString**方法

举例：

```
var i = 10;
var s = i.toString();
alert(typeof s); //将输出 String
```

js数字与字符串的区别

js的数字的加法与字符串的连接都是 + 符号， 所以究竟是加还是字符串的连接就取决与变量的类型。

举例：

```
var a = 'abc' + 'xyz'; //a的值为：abcxyz,字符串与字符串是连接
var a = 10 + 5; //a的值为：15，数字是加
var a = 'abc' + 10; //a的值为：abc10，字符串与数字，自动将10转换成字符串了
var a = 'abc' + 10 + 20 + 'cd'; //a的值为:abc1020cd
var a = 10 + 20 + 'abc' + 'cd'; //a的值为:30abccd，可以数字加的先数字加，然后再连接
```

补充：

js字符串转换数字。方法主要有三种

转换函数、强制类型转换、利用js变量弱类型转换。

1. 转换函数：

js提供了parseInt()和parseFloat()两个转换函数。前者把值转换成整数，后者把值转换成浮点数。只有对String类型调用这些方法，这两个函数才能正确运行；对其他类型返回的都是NaN(Not a Number)。

一些示例如下：

```
parseInt("1234blue"); //returns 1234
parseInt("0xA"); //returns 10
parseInt("22.5"); //returns 22
parseInt("blue"); //returns NaN
```

parseInt()方法还有基模式，可以把二进制、八进制、十六进制或其他任何进制的字符串转换成整数。基是由

parseInt()方法的第二个参数指定的，示例如下：

```
parseInt("AF", 16); //returns 175
parseInt("10", 2); //returns 2
parseInt("10", 8); //returns 8
parseInt("10", 10); //returns 10
```

如果十进制数包含前导0，那么最好采用基数10，这样才不会意外地得到八进制的值。例如：

```
parseInt("010"); //returns 8
parseInt("010", 8); //returns 8
parseInt("010", 10); //returns 10
```

parseFloat()方法与parseInt()方法的处理方式相似。

使用parseFloat()方法的另一不同之处在于，字符串必须以十进制形式表示浮点数，parseFloat()没有基模式。

下面是使用parseFloat()方法的示例：

```
parseFloat("1234blue"); //returns 1234.0
parseFloat("0xA"); //returns NaN
parseFloat("22.5"); //returns 22.5
parseFloat("22.34.5"); //returns 22.34
parseFloat("0908"); //returns 908
parseFloat("blue"); //returns NaN
```

2. 强制类型转换

还可使用强制类型转换（type casting）处理转换值的类型。使用强制类型转换可以访问特定的值，即使它是另一种类型的。

ECMAScript中可用的3种强制类型转换如下：

Boolean(value)——把给定的值转换成Boolean型；

Number(value)——把给定的值转换成数字（可以是整数或浮点数）；

String(value)——把给定的值转换成字符串。

用这三个函数之一转换值，将创建一个新值，存放由原始值直接转换成的值。这会造成意想不到的后果。

当要转换的值是至少有一个字符的字符串、非0数字或对象时，Boolean()函数将返回true。如果该值是空字符串、数字0、undefined或null，它将返回false。

可以用下面的代码段测试Boolean型的强制类型转换。

```
Boolean(""); //false - empty string
Boolean("hi"); //true - non-empty string
Boolean(100); //true - non-zero number
Boolean(null); //false - null
Boolean(0); //false - zero
Boolean(new Object()); //true - object
```

Number()的强制类型转换与parseInt()和parseFloat()方法的处理方式相似，只是它转换的是整个值，而不是部分值。

示例如下：

用　法 结　果

```
Number(false) 0
Number(true) 1
Number(undefined) NaN
Number(null) 0
Number( "5.5 ") 5.5
Number( "56 ") 56
Number( "5.6.7 ") NaN
Number(new Object()) NaN
Number(100) 100
```

最后一种强制类型转换方法String()是最简单的，示例如下：

```
var s1 = String(null); //"null"
```

```
var oNull = null;
var s2 = oNull.toString(); //won't work, causes an error
```

3. 利用js变量弱类型转换

举个小例子，一看，就会明白了。

上例利用了js的弱类型的特点，只进行了算术运算，实现了字符串到数字的类型转换，不过这个方法还是不推荐的