

# JS 里的数据类型转换

任意类型的值转换为字符串

## toString()

toString()函数返回一个表示该对象的字符串

类型	返回值	例子
数值	相应的字符串	<pre>var x = 1; x.toString(); // 返回 "1"</pre>
字符串	还是原来的值	<pre>var x = "qwer"; x.toString(); // 返回 "qwer"</pre>
布尔值	<pre>true</pre> 转换为字符串 <pre>"true"</pre> <pre>false</pre> 转换为字符串 <pre>"false"</pre>	<pre>var x = true; x.toString(); // 返回 "true"</pre>
对象(非数组)	返回一个类型字符串: "[object, Object]"	<pre>var x = {a: 1}; x.toString(); // 返回 "[object, Object]"</pre>
对象(数组)	返回该数组的字符串形式	<pre>var x = [1, 2, 3]; x.toString(); // 返回 "1, 2, 3"</pre>
null	报错	<pre>Uncaught TypeError: Cannot read property 'toString' of null</pre>
undefined	报错	<pre>Uncaught TypeError: Cannot read property 'toString' of undefined</pre>

## String()

你也可以使用 String 函数将任意类型的值转换为字符串

类型	返回值	例子
数值	相应的字符串	<code>var x = 1; String(x); // 返回 "1"</code>
字符串	还是原来的值	<code>var x = "qwer"; String(x); // 返回 "qwer"</code>
布尔值	<code>true</code> 转换为字符串 <code>"true"</code> <code>false</code> 转换为字符串 <code>"false"</code>	<code>var x = true; String(x); // 返回 "true"</code>
对象(非数组)	返回一个类型字符串: <code>"[object, Object]"</code>	<code>var x = {a: 1}; String(x); // 返回 "[object, Object]"</code>
对象(数组)	返回该数组的字符串形式	<code>var x = [1, 2, 3]; String(x); // 返回 "1,2,3"</code>
null	返回字符串"null"	<code>var x = null; String(x); // 返回 "null"</code>
undefined	返回字符串"undefined"	<code>var x = undefined; String(x); // 返回 "undefined"</code>

- 从上述内容可以发现 `toString()` 方法和 `String()` 方法不同之处在于 `null` 和 `undefined` 的返回值不同，这是因为 `String()` 方法的底层用的就是 `toString()` 方法，但是 `String()` 针对 `null`、`undefined`、`Symbol` 会有特殊处理，所以使用 `String` 方法将其它对象转化为字符串可以被认为是一种更加安全的做法。
- `toString()` 方法可以将一个数字转为 16 进制的数字

## 加法运算符

- 加法运算符 `+` 是最常见的运算符，用来求两个数值的和。
- JavaScript 允许非数值的相加。
- 比较特殊的是，如果是两个字符串相加，这时加法运算符会变成连接运算符，返回一个新的字符串，将两个原字符串连接在一起。

- 如果一个运算符是字符串，另一个运算符是非字符串，这时非字符串会转成字符串，再连接在一起。

例如: `1 + "a" // "1a"` 所以我们利用 JS 的这种特性可以更快捷的将任意类型的值转换为字符串，

也就是用一个任意类型的值加上一个空字符串 和使用 `String()` 方法得到的结果是一样的。

## 任意类型的值转换为布尔值

### `boolean()`

`boolean` 函数可以将任意类型的值转换为布尔值

#### 转换规则

除了以下五个值的转换结果为 `false`, 其余的值全为 `true`

- `undefined`
- `null`
- `-0` 或 `+0`
- `NaN`
- `''` (空字符串)

当然 `false` 本身转换为布尔值肯定也是 `false` 啦。

注意，所有对象（包括空对象）的转换结果都是 `true`

## 取反运算符(!)

- 取反运算符是一个感叹号，用于将布尔值变为相反值，即 `true` 变成 `false`，`false` 变成 `true`。
- 对于非布尔值，取反运算符会将其转为布尔值。

- 如果对一个值连续做两次取反运算，等于将其转为对应的布尔值，与 `Boolean` 函数的作用相同。这是一种常用的类型转换的写法。

从图中可以看到：`!!`得到的结果和 `Boolean()` 方法得到的结果是一样的，所以我们可以用这种更便捷的方法转换为布尔值

## 任意类型的值转换为数值

### `Number()`

使用 `Number` 函数，可以将任意类型的值转化成数值。

### `parseInt()`

### `parseFloat()`

### 减法运算符

### 数值运算符