**学习目标**

1、掌握Number

2、掌握isNaN()

3、掌握数值转换

（1）、Number()

（2）、parseInt()

（3）、parseFloat()

**Number**

表示整数和浮点数

整数：1，2，3，-1，-5

浮点数：1.1，1.5，0.5

操作：用typeof来打印出看数据类型

**NaN**：number中的特殊类型，它也是数值类型，即非数值（Not a Number）是一个特殊的数值

例子：

var age = 20;

console.log(age - 3); // 17

console.log(age - 'abc'); //数字减字符串，是个什么鬼？

解说：

NaN它本来是要给你返回一个数值的值，但是却没有办法返回一个数值，所以返回了一个NaN（哭泣的表情，哥们，我真的没有办法给你返回一个数值啊！）

解说：

NaN的类型：特殊的number类型，用typeof打印了看

说明： 一定要记住

1、任何涉及NaN的操作（例如NaN/10 NaN+1 NaN-1）都会返回NaN。

2、NaN与任何值都不相等，包括NaN本身。

**isNaN()**

语法：isNaN(n)

功能：n是不是,不是一个数字(非数字)，不是数字（包括NaN）返回true，是数字返回false。

返回值：boolean (真或假)

参数：参数n可以是任何类型

例子：

var obj = 12-'a'; //NaN

console.log(isNaN(obj));

var num = 'pzh1000@163.com'; //字符串

console.log(isNaN(num));

var num = 12; //数字

console.log(isNaN(num));

var num = '50'; //字符串

console.log(isNaN(num));

字符串类型的数字也返回false，说明isNaN得到参数后，它会尝试把这个值转换成数值，如果能转换成功，就返回false，如果转换不成功，就返回true。（实际isNaN内部调用了Number()方法）

**数值转换**

**Number()**

1、可以用于任何数据类型

2、只有能转成功与不能转成功

3、不能转成功就是NaN

4、但不论成功与否，用typeof查看，结果都是数值类型。

如果是 Boolean 值， true 和 false 将分别被转换为 1 和 0。

如果是数字值，只是简单的传入和返回。

如果是 null 值，返回 0。

如果是 undefined ，返回 NaN 。

Number()对字符串的转换规则：

1、如果字符串中只包含数字（包括前面带正号或负号的情况），则将其转换为十进制数值，即 "1"会变成 1， "123" 会变成 123，而 "011" 会变成 11（注意：前导的零被忽略了）；

2、如果字符串中包含有效的浮点格式，如 "1.1" ，则将其转换为对应的浮点数值（同样，也会忽略前导零）；

3、如果字符串中包含有效的十六进制格式，例如 "0xf" ，则将其转换为相同大小的十进制整数值；

4、如果字符串是空的（不包含任何字符），则将其转换为 0；

5、如果字符串中包含除上述格式之外的字符，则将其转换为 NaN 。

演示：

var id = '12';

id = Number(id); // id强制转换为数值

console.log(id); //12，转换成功

console.log(typeof id); // number

var id = 'abc';

id = Number(id); // id强制转换为数值

console.log(id); // NaN，转换不成功，不是个数字

console.log(typeof id); // number

**parseInt()**，专门用于把字符串转换成数值，只能转换成整数。

1、如果第一个字符不是数字字符或者负号， parseInt()就会返回 NaN ；

2、会忽略字符串前面的空格，直至找到第一个非空格字符

3、转换空字符串返回NaN。

4、该函数提供第二个参数：转换时使用的基数（即多少进制），提醒一定要加上第二个参数，参数为10

演示：

var aa = '28px';

console.log(parseInt(aa, 10) + 10);

var c = 'abc58';

console.log(parseInt(c, 10)); // NaN

说明：操作的参数必须是以数字开头，从左向右一个一个看，直到非数字为止。后面的都不看了。

console.log(parseInt('0123', 10));

var c = '00011abc58';

console.log(parseInt(c, 10)); // 11

说明：会忽略前面的0。

**parseFloat()**，专门用于把字符串转换成数值，它没有第二个参数。它可以返回整数，也可以返回浮点数。

从第一个字符开始解析每个字符，直至遇见一个无效的浮点数字符为止

说明：

除了第一个小数点有效外，parseFloat()与parseInt()的第二个区别在于

它始终都会忽略前导的零。

举个粟子：

var c = '000.11abc58';

console.log(parseFloat(c)); // 0.11

console.log(parseFloat('0123'));

console.log(parseFloat('0.123'));

console.log(parseFloat('12.34元')); //12.34

console.log(parseFloat('12.34.56元')); //12.34

**总结**：当我们要把一个字符串中的数字提取出来，一般不用Number()，而用parseInt、parseFloat，但是前提条件这些值必须是以数字开头的。如果想提取整数，用parseInt，如果想提取浮点数，用parseFloat。