运算符结合性问题

1. Java中所有运算符都是右结合性的
   1. 比如a+b+c = a+(b+c)
2. JavaScript中的运算符要看具体符号，各符号结合性不同

[**js运算符(运算符的结合性)**](http://www.cnblogs.com/stephenykk/p/3143706.html)

 1、javascript具有下列种类的运算符:算术运算符;逻辑运算符;比较运算符;  
  
2、目的分类:字符串运算符;逻辑运算符;逐位运算符;赋值运算符;  
  
3、特殊运算符:条件运算符;typeof运算符;创建对象运算符new;delete运算符;void运算符号;逗号运算符;  
  
算术运算符:+ 、 - 、\* 、 / 、 % 、 -(一元取反) 、 ++ 、 --  
  
等同运算符与全同运算符:==(等于) 、 ===(恒等于)、 !=(不等于) 、 !==(不恒等于)  
  
比较运算符:< 、 > 、 <= 、 >=  
  
字符串运算符: +  
  
逻辑运算符:&& 、 ||、 !、  
  
赋值运算符:=、 +=、 \*=、 -=、 /=  
  
规则:  
  
一、一定要注意传递给运算符的数据类型和返回的数据类型!不同的运算符都期望它的运算数表达式计算出的结果符合某种数据类型。  
  
例如:对字符串是不能进行乘法运算的,"a" \* "b" 是不合法的,但是,在可能的情况下,javascript会把表达式转换成正确的类型,因此,表达式 "3" \* "5" 是合法的,javascript将字符串转换成数字执行运算,结果返回的是数字15,而不是字符串"15"。  
  
二、+ 根据运算数的不同,具有不同的表现: (~~~ +运算符偏爱字符串，只要有操作数是字符串(包括数值字符串)，都会进行字符串连接操作)  
  
字符串 + 字符串 = 字符串(被连接);"a" + "b" = "ab" "5" + "6" = "56"  
  
字符串 + 数字 = (字符串转换成数字)字符串(被连接);"a" + 5 = "a5" 5被转换成字符串 "1" + 0 = "10"  
  
数字 + 数字 = 数字(相加)5 + 5 = 10。  
  
三、注意运算符的结合性,有些运算符从左到右的结合性;有些从右到左结合性。  
  
例如:w = a + b + c 等同于 w = (a + b) + c;  
  
w = ---b 等同于 w = - ( - ( -b ) ); w = a = b = c 等同于 w= ( a = ( b = c ))  
  
一元运算符、赋值运算符、三元运算符的结合性是从右到左的;  
  
四、算术运算符规则:  
  
“+”:可以进行“加法”与“连接”运算;如果2个运算符中的一个是字符串,javascript就将另一个转换成字符串,然后将2个运算数连接起来。  
  
“+”:如果一个运算数是对象,javascript就会将对象转换成数字做加法运算或者字符串做连接运算;  
  
“-”“\*”“/” 如果2个运算数中的一个是是非数字,则将它转化成数字执行数学运算。  
  
“/” 在javascript中,由于所有的数都是浮点数,除法的结果都是浮点数,5 / 2 = 2.5 ;除数为0的结果为正负无穷;0/0为NaN;  
  
“%”模运算符:计算第一个运算数对第二个运算数的模,就是第一个运算数被第二个运算数除时,返回余数。如果运算数是非数字的,则转换成数字。  
  
“-”一元取反:将运算数取反,如果运算数不是数字,则转化成数字。  
  
“++”“--”增量运算符/减量运算符号:运算数必须是一个变量、树组的一个元素或对象的一个属性。如果运算数不是数字,则转换成数字。  
  
注意:如果“++”位于运算数之前,先对运算数进行增量,然后计算运算数增长后的值。  
  
如果“--”位于运算数之后,先计算运算数未增长之前的值,然后对运算数增量。  
  
例如:i = 1; //赋值给i值1  
  
j = ++i; //先将i增量为2,然后将2赋值给j,i的值是2,j的值也是2。  
  
i = 1; //赋值给i值1  
  
j = i++; //先将i赋值给j,然后对将i值增量到2,i的值是2,j的值是1。  
  
减量运算符“--”与“++”相同。  
  
五、等同运算符与全同运算符:  
  
(一)“==”“!==”等同运算符与非等同运算符:  
  
比较2个运算数,返回一个不布尔值。  
  
比较数值、字符串、布尔值使用的都是量值。当且仅当两个变量存放的值相等的时候,他们才相等。  
  
比较对象、数组、函数的时候,使用的是引用,只有2个变量引用的是同一个对象时,他们才是相等的  
  
两个不同的数组是完全不同的,即使他们具有完全相同的元素。对于存放对象、数组、函数的引用的变量来说,只有当他们引用的是同一个对象、数组、函数的时候,他们才相等。  
  
注意!应遵循的原则:  
  
当两个运算数的类型不同时:将他们转换成相同的类型,  
  
1)一个数字与一个字符串,字符串转换成数字之后,进行比较。  
  
2)true转换为1、false转换为0,进行比较。  
  
3)一个对象、数组、函数 与 一个数字或字符串,对象、数组、函数转换为原始类型的值,然后进行比较。(先使用valueOf,如果不行就使用toString)  
  
4)其他类型的组合不星等。  
  
想两个运算数类型相同,或转换成相同类型后:  
  
1)2个字符串:同一位置上的字符相等,2个字符串就相同。  
  
2)2个数字:2个数字相同,就相同。如果一个是NaN,或两个都是NaN,则不相同。  
  
3)2个都是true,或者2个都是false,则想同。  
  
4)2个引用的是同一个对象、函数、数组,则他们相等,如果引用的不是同一个对象、函数、数组,则不相同,即使这2个对象、函数、数组可以转换成完全相等的原始值。  
  
5)2个null,或者2个都是未定义的,那么他们相等。  
  
“!=”非等同运算符:与等同运算符检测的结果相反。  
  
(二)“===”“!===”全同运算符与非全同运算符号:  
  
全同运算符遵循等同运算符的比较规则,但是它不对运算数进行类型转换,当两个运算数的类型不同时,返回false;只有当两个运算数的类型相同的时候,才遵循等同运算符的比较规则进行比较。  
  
“!==”非全同运算符与全同运算符比较的结果相反。如果两个运算数的类型或者值不同的时候,返回true。  
  
六、比较运算符:  
  
这些比较运算符用于比较不同类型的值,结果返回一个布尔值。  
  
“<”“>”“<=”“>=”  
  
注意规则:比较的运算数可以是任意类型的,但是只能在数字和运算数上执行。不是数字和字符串的运算数将被转换成数字或字符串。  
  
1)如果两个运算数都是数字,或者都可以转换成数字,则按照数字的大小进行比较;  
  
2)如果两个运算数都是字符串,或者都可以转换成字符串,则将按照字母顺序进行比较;  
  
3)如果字符串遇到数字,则将字符串转换成数字进行比较。  
  
4)如果运算数既不能转换成数字,也不能转换成字符串,则返回结果false。  
  
七、字符串运算符:  
  
并没有专用的字符串运算符,只是有些运算符在遇到字符串运算数的时候,表现不同。  
  
(一)“+”连接2个字符串;  
  
1)当2个运算数都是字符串的时候,连接起来;  
  
2)当其中有一个是数字的时候,将数字转换成字符串,连接起来;  
  
(二)“>”这样的比较运算符通过比较确认两个字符串的顺序,比较采用字符的顺序,较小的位于教大的前面,大写字母位于小写字母之前。  
  
(三)“+”的作用方法取决于计算顺序,  
  
如:s = 1 + 2 +"var" 则:返回结果3var; 因为先计算1+2,然后将结果3转换成字符串与"var"连接;  
  
如:s = "var" + 1 + 2 则:返回结果var12; 因为先计算var与1的连接,在将结果var1与转换成字符串的2连接起来。  
  
八、逻辑运算符:  
  
用于执行布尔运算,常常和比较运算符一起使用,来表示复杂的比较运算。  
  
“&&” 逻辑与运算、“||”逻辑或运算符、“!”逻辑非运算符  
  
(一)“&&”当两个运算数都是布尔值的时候,逻辑与对他们进行and操作,即:当且仅当两个布尔值都是true的时候,返回结果true,否则返回false。  
  
注意:实际表现  
  
“&&”将检测第一个表达式运算数布尔值,如果第一个运算数表达式返回false,则返回左边第一个运算数表达式的值:false;否则将继续检测第二个即右边的运算数表达式,然后返回第二个运算数表达式的值;  
  
例如:if (a = b) stop(); 与 (a = b) && stop(); 等价  
  
不赞成使用这个方法,因为运算符右边的代码不能保证会被执行,  
  
例如:if ((a < b) && (b++ <10) ) stop();如果运算符右边的增量必须的话,却有可能由于前面的返回值而不被执行,一般反对在&&的右边使用具有其他作用的表达式(赋值、函数调用、增减量);  
  
将“&&”看作是一个布尔代数的运算符,会更简单与安全。  
  
(二)“||”当两个运算数都是布尔值的时候,逻辑或对他们进行or操作,即:当两个布尔值有一个是true的时候,返回结果true,否则返回false。  
  
注意:实际表现  
  
“||”将检测第一个表达式运算数布尔值,如果第一个运算数表达式返回true,则返回左边第一个运算数表达式的值:true;否则将继续检测第二个即右边的运算数表达式,然后返回第二个运算数表达式的值;  
  
同样不赞成使用这个方法,因为运算符右边的代码不能保证会被执行,  
  
一般反对在||的右边使用具有其他作用的表达式(赋值、函数调用、增减量);  
  
将“||”看作是一个布尔代数的运算符,会更简单与安全。  
  
(三)“!”逻辑非是一个一元运算符,放在运算数之前,它的目的是对运算数取反。  
  
九、赋值运算符:  
  
(一)“=”是赋值运算符;他总是期望左边的运算数是一个变量、数组的一个元素或对象的一个属性;  
  
期望右边是一个任意类型的任意的值;  
  
从右到左的结合性,如果一个表达式中有多个赋值运算符,则从最右边开始计算。  
  
注意:每一个赋值表达式都有一个值,就是运算符右边的值;  
  
(二)可以使用带操作的赋值运算  
  
“+=” 左边的值加上右边的值后,赋值于左边的值。“-=”“/=”“\*=”方法一样;  
  
十、其他运算符:  
  
“?:”条件运算符是唯一的一个三元运算符;  
  
一个表达式的布尔结果?表达式1(任意类型的任意值):表达式2(任意类型的任意值);  
  
根据第一个运算数的布尔值结果,如果为true,则执行第二个运算数表达式,返回第二个运算数表达式的值;如果第一个运算数的布尔值结果是false,则执行第三个运算数表达式,返回第三个运算数表达式的值。