JavaScript的=、==和=== - 风声水起 - 博客频道 - CSDN

JavaScript支持“=”、“==”和“===”运算符。你应当理解这些（赋值、相等、恒等）运算符之间的区别，并在编码过程中小心使用。

JavaScript对象的比较是引用的比较，而不是值的比较。对象和其本身是相等的，但和其他任何对象都不相等。如果两个不同的对象具有相同数量的属性，相同的属性名和值，它们依然是不相等的。相应位置的数组元素是相等的两个数组也是不相等的。

严格相等运算符“===”首先计算其操作数的值，然后比较这两个值，比较过程没有任何类型转换：

- 如果两个值类型不相同，则它们不相等。
- 如果两个值都是null或者都是undefined，则它们不相等。
- 如果两个值都是布尔值true或false，则它们相等。
- 如果其中一个值是NaN，或者两个两个值都是NaN，则它们不相等。NaN和其他任何值都是不相等的，包括它本身！！！通过x!==x来判断x是否为NaN，只有在x为NaN的时候，这个表达式的值才为true。
- 如果两个值为数字，且数值相等，则它们相等。如果一个为0，另一个为-0，则它们同样相等。
- 如果两个值为字符串，且所含的对应位上的16位数完全相等，则它们相等。如果它们的长度或内容不同，则它们不等。两个字符串可能含义完全一样且所显示出手字符也一样，但具有不同编码的16位值。JavaScript并不对Unicode进行标准化的转换，因此像这样的字符串通过"==="和"=="运算符的比较结果也不相等。
- 如果两个引用值同一个对象、数组或函数，则它们是相等的。如果指向不同的对象，则它们是不等的。尽管两个对象具有完全一样的属性。

相等运算符“==”和恒等运算符相似，但相等运算符的比较并不严格。如果两个操作数不是同一类型，那么相等运算符会尝试一些类型转换，然后进行比较：

- 如果两个操作数的类型相同，则和上文所述的严格相等的比较规则一样。如果严格相等，那么比较结果为相等。如果它们不严格相等，则比较结果为不相等。
- 如果两个操作数类型不同，“==”相等操作符也可能会认为它们相等。检测相等将会遵守如下规则和类型转换：

6. 如果一个值是null，另一个是undefined，则它们相等。
7. 如果一个值是数字，另一个是字符串，先将字符串转换为数字，然后使用转换后的值比较。
8. 如果其中一个值是true，则将其转换为1再进行比较。如果其中一个值是false，则将基转换为0再进行比较。
9. 如果一个值是对象，另一个值是数字或字符串，则将对象转换为原始值，然后再进行比较。对象通过toString()方法或valueOf()方法转换为原始值。JavaScript核心的内置类首先尝试使用valueOf()，再尝试使用toString()，除了日期类，日期类只使用toString()转换。那些不是JavaScript语言核心中的对象则通过各自的实现中定义的方法转换为原始值。
10. 其他不同类型之间的比较均不相等.