***delete*** 操作符用于删除对象的某个属性。

与通常对 delete 的理解不同，delete 操作符与直接释放内存无关。内存管理 通过断开引用来间接完成的，查看内存管理页可了解详情。

delete 操作符会从某个对象上移除指定属性。成功删除的时候回返回 true，否则返回 false。但是，重要的是要考虑以下情况：

1、如果你删除的属性在对象上不存在，那么delete将不会起作用，但仍会返回true

2、如果 delete 操作符删除成功，则被删除的属性将从所属的对象上彻底消失。然后，如果该对象的原型链上有一个同名属性，则该对象会从原型链上继承该同名属性。（也就是说delete操作只会在自身的属性上起作用）

3、任何使用 var 声明的属性不能从全局作用域或函数的作用域中删除。

1）这样的话，delete操作不能删除任何在全局作用域中的函数（无论这个函数是来自于函数声明或函数表达式）

2）除了在全局作用域中的函数不能被删除，在对象(object)中的函数是能够用delete操作删除的。

4、任何用let或const声明的属性不能够从它被声明的作用域中删除。

5、不可设置的(Non-configurable)属性不能被移除。这意味着像Math, Array, Object内置对象的属性以及使用Object.defineProperty()方法设置为不可设置的属性不能被删除。

在严格模式下，如果直接对一个变量的引用、函数的参数或者函数名使用delete操作，将会抛出语法错误（SyntaxError）。

任何使用var声明的变量都会被标记为不可设置的。在下面的例子中，salary是不可设置的以及不能被删除的。在非严格模式下，下面的delete操作将会返回false。严格模式下无法删除对象的变量或属性，除非configurable属性被设为true。

function Employee() {

delete salary;

var salary;

}

Employee();

当你删除一个数组元素时，数组的 length 属性并不会变小。即便你删除了数组的最后一个元素也是如此。被删除的元素值变为undefined，如果你想让一个数组元素继续存在但是其值是 undefined，那么可以使用将 undefined 赋值给这个元素而不是使用 delete。