**[JS中声明全局变量](https://www.cnblogs.com/signheart/p/3ca2afaca43d2dc69ce72c862a3ae81c.html)**

JS中声明全局变量主要分为显式声明或者隐式声明下面分别介绍。

**声明方式一：**

使用var（关键字）+变量名(标识符)的方式在function外部声明，即为全局变量，否则在function声明的是局部变量。该方式即为显式声明详细如下：

复制代码代码如下:

var test = 5;  //全局变量  
function a（）  
{  
  var cc=3; //局部变量  
  alert(test);  
}  
function b（）{alert(test);}

**声明方式二：**

没有使用var，直接给标识符test赋值，这样会隐式的声明了全局变量test。即使该语句是在一个function内，当该function被执行后test变成了全局变量。

复制代码代码如下:

test = 5;//全局变量   
function a()  
{  
  aa=3; //全局变量  
  alert(test);  
}

**声明方式三：**

使用window全局对象来声明，全局对象的属性对应也是全局变量，详细如下：

复制代码代码如下:

window.test;  
window.test = 5;

这种方式经常被用到一个匿名函数执行后将一些函数公开到全局。 如JQuery1.5中最末一句

复制代码代码如下:

window.jQuery = window.$ = jQuery;

**全局变量的优点：**

可以减少变量的个数，减少由于实际参数和形式参数的数据传递带来的时间消耗。

**全局变量的缺点：**（1）全局变量保存在静态存贮区，程序开始运行时为其分配内存，程序结束释放该内存。与局部变量的动态分配、动态释放相比，生存期比较长，因此过多的全局变量会占用较多的内存单元。  
  
（2）全局变量破坏了函数的封装性能。函数象一个黑匣子，一般是通过函数参数和返回值进行输入输出，函数内部实现相对独立。但函数中如果使用了全局变量，那么函数体内的语句就可以绕过函数参数和返回值进行存取，这种情况破坏了函数的独立性，使函数对全局变量产生依赖。同时，也降低了该函数的可移植性。  
  
（3）全局变量使函数的代码可读性降低。由于多个函数都可能使用全局变量，函数执行时全局变量的值可能随时发生变化，对于程序的查错和调试都非常不利。  
因此，如果不是万不得已，最好不要使用全局变量。