### 申明

JS中变量申明分显式申明和隐式申明。

Var a = 1;显式申明

a = 1;隐式申明

1.全局变量：声明在函数外部的变量（所有没有var直接赋值的变量都属于全局变量）

2.局部变量：声明在函数内部的变量（所有没有var直接赋值的变量都属于全局变量）

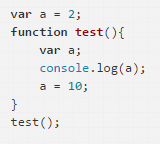
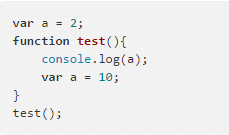
一个变量与一个对象的一个属性，有很多相似的地方，实际上，它们并没有什么本质区别。**在 Javascript 中，任何变量都是某个特定对象的属性**。

全局变量都是全局对象的属性

开始运行且没有执行 Javascript 代码之前，会有一个「全局对象」被创建

函数的局部变量.

在函数开始执行时，会有一个对应的「调用对象」被创建，函数的局部变量都作为它的属性而存储。这样可以防止局部变量覆盖全局变量



在函数体内部，局部变量的优先级比通明的全局变量要高； 变量名相同，在声明局部变量的函数体范围内，局部变量将覆盖同名的全局变量。

var scope = 'global';

function checkScope(){

var scope = 'local';

console.log(scope); // local

}

checkScope();

console.log(scope); // global



全局变量在整个上下文都有效只是在没有赋值之前调用，会输出undefin

函数作用域是针对局部变量来说的，在函数中定义的变量在函数外不能获取

JS如果用var在函数体中声明变量，那么此变量在且只在该函数体内有效，函数运行结束时，本地变量即可销毁了。

js中没有块级作用域，使用闭包实现

var name="global";

if(true){

var name="local";

console.log(name)

}

console.log(name);

### 作用域链

name="lwy";

function t(){

var name="tlwy";

function s(){

var name="slwy";

console.log(name);

}

function ss(){

console.log(name);

}

s();

ss();

}

t();

参考

<https://www.cnblogs.com/silentjesse/p/4024536.html>