

### 3.1.4 严格模式

ECMAScript 5 引入了严格模式 (strict mode) 的概念。严格模式是为 JavaScript 定义了一种不同的解析与执行模型。在严格模式下, ECMAScript 3 中的一些不确定的行为将得到处理, 而且对某些不安全的操作也会抛出错误。要在整个脚本中启用严格模式, 可以在顶部添加如下代码:

```
"use strict";
```

这行代码看起来像是字符串, 而且也没有赋值给任何变量, 但其实它是一个编译指示 (pragma), 用于告诉支持的 JavaScript 引擎切换到严格模式。这是为不破坏 ECMAScript 3 语法而特意选定的语法。

在函数内部的上方包含这条编译指示, 也可以指定函数在严格模式下执行:

```
function doSomething(){
    "use strict";
    //函数体
}
```

严格模式下, JavaScript 的执行结果会有很大不同, 因此本书将会随时指出严格模式下的区别。支持严格模式的浏览器包括 IE10+、Firefox 4+、Safari 5.1+、Opera 12+ 和 Chrome。

## 3.2 关键字和保留字

ECMA-262 描述了一组具有特定用途的**关键字**, 这些关键字可用于表示控制语句的开始或结束, 或者用于执行特定操作等。按照规则, 关键字也是语言保留的, 不能用作标识符。以下就是 ECMAScript 的全部关键字 (带\*号上标的是第 5 版新增的关键字):

break	do	instanceof	typeof
case	else	new	var
catch	finally	return	void
continue	for	switch	while
debugger*	function	this	with
default	if	throw	
delete	in	try	

ECMA-262 还描述了另外一组不能用作标识符的**保留字**。尽管保留字在这门语言中还没有任何特定的用途, 但它们有可能在将来被用作关键字。以下是 ECMA-262 第 3 版定义的全部保留字:

abstract	enum	int	short
boolean	export	interface	static
byte	extends	long	super
char	final	native	synchronized
class	float	package	throws
const	goto	private	transient
debugger	implements	protected	volatile
double	import	public	

第 5 版把在非严格模式下运行时的保留字缩减为下列这些:

class	enum	extends	super
const	export	import	

在严格模式下, 第 5 版还对以下保留字施加了限制:

implements	package	public
interface	private	static
let	protected	yield

```
alert(message); // hi
```

这个例子省略了 `var` 操作符，因而 `message` 就成了全局变量。这样，只要调用过一次 `test()` 函数，这个变量就有了定义，就可以在函数外部的任何地方被访问到。



虽然省略 `var` 操作符可以定义全局变量，但这也不是我们推荐的做法。因为在局部作用域中定义的全局变量很难维护，而且如果有意地忽略了 `var` 操作符，也会由于相应变量不会马上就有定义而导致不必要的混乱。给未经声明的变量赋值在严格模式下会导致抛出 `ReferenceError` 错误。

可以使用一条语句定义多个变量，只要像下面这样把每个变量（初始化或不初始化均可）用逗号分隔开即可：

```
var message = "hi",  
    found = false,  
    age = 29;
```