

Null类型

如果定义的变量准备在将来用于保存对象，那么最好将该变量初始化为 `null` 而不是其他值。这样一来，只要直接检查 `null` 值就可以知道相应的变量是否已经保存了一个对象的引用，如下面的例子所示：

```
if (car != null){  
    // 对 car 对象执行某些操作  
}
```

实际上，`undefined` 值是派生自 `null` 值的，因此 ECMA-262 规定对它们的相等性测试要返回 `true`：

```
alert(null == undefined);    //true
```

尽管 `null` 和 `undefined` 有这样的关系，但它们的用途完全不同。如前所述，无论在什么情况下都没有必要把一个变量的值显式地设置为 `undefined`，可是同样的规则对 `null` 却不适用。换句话说，只要意在保存对象的变量还没有真正保存对象，就应该明确地让该变量保存 `null` 值。这样做不仅可以体现 `null` 作为空对象指针的惯例，而且也有助于进一步区分 `null` 和 `undefined`。