

JavaScript 变量提升

JavaScript 中，函数及变量的声明都将被提升到函数的最顶部。

JavaScript 中，变量可以在使用后声明，也就是变量可以先使用再声明。

以下两个实例将获得相同的结果：

实例 1

```
x = 5; // 变量 x 设置为 5

elem = document.getElementById("demo"); // 查找元素
elem.innerHTML = x;                      // 在元素中显示 x

var x; // 声明 x
```

[尝试一下 »](#)

实例 2

```
var x; // 声明 x
x = 5; // 变量 x 设置为 5

elem = document.getElementById("demo"); // 查找元素
elem.innerHTML = x;                      // 在元素中显示 x
```

[尝试一下 »](#)

要理解以上实例就需要理解 "hoisting(变量提升)"。

变量提升：函数声明和变量声明总是会被解释器悄悄地被"提升"到方法体的最顶部。

JavaScript 初始化不会提升

JavaScript 只有声明的变量会提升，初始化的不会。

以下两个实例结果结果不相同：

实例 1

```
var x = 5; // 初始化 x
var y = 7; // 初始化 y

elem = document.getElementById("demo"); // 查找元素
elem.innerHTML = x + " " + y;           // 显示 x 和 y
```

[尝试一下 »](#)

实例 2

```
var x = 5; // 初始化 x

elem = document.getElementById("demo"); // 查找元素
elem.innerHTML = x + " " + y;           // 显示 x 和 y

var y = 7; // 初始化 y
```

[尝试一下 »](#)

实例 2 的 y 输出了 **undefined**，这是因为变量声明 (var y) 提升了，但是初始化(y = 7) 并不会提升，所以 y 变量是一个未定义的变量。

实例 2 类似以下代码：

```
var x = 5; // 初始化 x
var y;      // 声明 y

elem = document.getElementById("demo"); // 查找元素
elem.innerHTML = x + " " + y;           // 显示 x 和 y

y = 7;      // 设置 y 为 7
```

在头部声明你的变量

对于大多数程序员来说并不知道 JavaScript 变量提升。

如果程序员不能很好的理解变量提升，他们写的程序就容易出现一些问题。

为了避免这些问题，通常我们在每个作用域开始前声明这些变量，这也是正常的 JavaScript 解析步骤，易于我们理解。



JavaScript 严格模式(strict mode)不允许使用未声明的变量。
在下一个章节中我们将会学习到 "**严格模式(strict mode)**"。

[← JavaScript 代码规范](#)[JavaScript 严格模式\(use strict\) →](#)**3 篇笔记**[📝 写笔记](#)