



# 《一头扎进 JavaScript(高级篇)》视频教程

## 第一章 Function 对象

**Java1234\_小锋**

扣扣:527085608

**Java1234 官方群 1, 2, 3, 4, 5, 6: (已满)**

**Java1234 官方群 7: 177914304**

## 第一节：Function 对象引入

Function 是 js 的方法对象，可以用 Function 实例化出任何 js 方法对象；

## 第二节：用 Function 创建方法对象

## 第三节：Function 对象属性

## 第四节：Function 对象方法



# 《一头扎进 JavaScript(高级篇)》视频教程

## 第二章 JavaScript 闭包

**Java1234\_小锋**

扣扣:527085608

**Java1234 官方群 1, 2, 3, 4, 5, 6: (已满)**

**Java1234 官方群 7: 177914304**

## 第一节：Js 变量的作用域

## 第二节：从外部读取方法内部的局部变量

## 第三节：闭包的概念

各种专业文献上的“闭包”（closure）定义非常抽象，很难看懂。我的理解是，闭包就是能够读取其他函数内部变量的函数。

由于在 Javascript 语言中，只有函数内部的子函数才能读取局部变量，因此可以把闭包简单理解成“定义在一个函数内部的函数”。

所以，在本质上，闭包就是将函数内部和函数外部连接起来的一座桥梁。

## 第四节：闭包的用途

一个是前面提到的可以读取函数内部的变量，另一个就是让这些变量的值始终保持在内存中。

## 第五节：闭包的使用注意点

- 1) 由于闭包会使得函数中的变量都被保存在内存中，内存消耗很大，所以不能滥用闭包，否则会造成网页的性能问题，在 IE 中可能导致内存泄露。解决方法是，在退出函数之前，将不使用的局部变量全部删除。
- 2) 闭包会在父函数外部，改变父函数内部变量的值。所以，如果你把父函数当作对象（object）使用，把闭包当作它的公用方法（Public Method），把内部变量当作它的私有属性（private value），这时一定要小心，不要随便改变父函数内部变量的值。



# 《一头扎进 JavaScript(高级篇)》视频教程

## 第三章 JavaScript 面向对象实现

**Java1234\_小锋**

**扣扣:527085608**

**Java1234 官方群 1, 2, 3, 4, 5, 6: (已满)**

**Java1234 官方群 7: 177914304**

## 第一节：面向对象三大特征

## 第二节：JavaScript 自定义对象

创建对象方式：方式一，对象初始化器方式；方式二，构造函数方式；  
对象属性定义：私有属性；对象属性；类属性；  
对象方法定义：私有方法；对象方法；类方法；

## 第三节：JavaScript 实现封装特性

## 第四节：JavaScript 实现继承特性

Apply() 实现属性和方法的继承；  
Prototype 实现原型的继承；

## 第五节：JavaScript 实现多态特性