# 第一章 JS采坑

## **1.1 语法基础**

### **1.1.1包围函数**

包围函数（function(){})的第一对括号向脚本返回未命名的函数，随后一对空括号立即执行返回的未命名函数，括号内为[匿名函数](https://www.baidu.com/s?wd=%E5%8C%BF%E5%90%8D%E5%87%BD%E6%95%B0&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3ujN9PHbzmvDdPyRvnAcY0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnH61njn1P10zrjf1nWmYnjmvn0" \t "https://zhidao.baidu.com/question/_blank)的参数。  
作用：可以用它创建命名空间，只要把自己所有的代码都写在这个特殊的函数包装内，那么外部就不能访问，除非你允许(变量前加上window，这样该函数或变量就成为全局)。各JavaScript库的代码也基本是这种组织形式。  
// 如果你不在意返回值，或者不怕难以阅读  
// 你甚至可以在function前面加一元操作符号  
  
!function () { /\* code \*/ } ();  
~function () { /\* code \*/ } ();  
-function () { /\* code \*/ } ();  
+function () { /\* code \*/ } ();

### **1.1.2 Function**

Function**构造函数** 创建一个新的Function对象。 在 JavaScript 中, 每个函数实际上都是一个Function对象。

|  |  |
| --- | --- |
| 1  2  3 | var sum = new Function('a', 'b', 'return a + b');  console.log(sum(2, 6));  // expected output: 8 |

##### **语法：**

new Function ([arg1[, arg2[, ...argN]],] functionBody)

##### 描述：

使用Function构造器生成的Function对象是在函数创建时解析的。这比你使用[函数声明](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Operators/function)或者[函数表达式](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Operators/function)(function)并在你的代码中调用更为低效，因为使用后者创建的函数是跟其他代码一起解析的。

### **1.1.3 Arguments**

其实Javascript并没有重载函数的功能，但是Arguments对象能够模拟重载。Javascrip中国每个函数都会有一个Arguments对象实例arguments，它引用着函数的实参，可以用数组下标的方式"[]"引用arguments的元素。arguments.length为函数实参个数，arguments.callee引用函数自身。

## **1.2 语言基础API采坑**

### **1.2.1 obj.hasOwnProperty(prop)**

* 介绍:

方法会返回一个布尔值，指示对象自身属性中是否具有指定的属性

所有继承了 Object 的对象都会继承到 hasOwnProperty 方法。这个方法可以用来检测一个对象是否含有特定的自身属性；和 in 运算符不同，该方法会忽略掉那些从原型链上继承到的属性

* 参数：

prop要检测的属性  字符串 名称或者 Symbol。

* 返回值：

用来判断某个对象是否含有指定的属性的 Boolean 。