



传智播客 · 前端与移动开发学院  
<http://web.itcast.cn>

## 前端基本功—javascript 第十三天

# 目录

闭包 基础      面向对象基础

## 1.1 闭包

在程序语言中，所谓闭包，是指语法域位于某个特定的区域，具有持续参照(读写)位于该区域内自身范围之外的执行域上的非持久型变量值能力的段落。这些外部执行域的非持久型变量神奇地保留他们在闭包最初定义(或创建)时的值。

白话： 我们可以用一个函数 去访问 另外一个函数的内部变量的方式就是闭包。

内部变量 是 局部变量      那我们知道，局部变量是不可以为别人随便使用也。

```
function fun() {  
    var num = 10;  
}  
console.log(num); // 这样子就错了 num 是一个局部变量
```

### 测试题

```
1  function outerFun()  
2  {  
3      var a=0;  
4      function innerFun()  
5      {  
6          a++;  
7          alert(a);  
8      }
```

```
9   return innerFun; //注意这里
10  }
11  var obj=outerFun();
12  obj();  obj();
13  var obj2=outerFun();
14  obj2();  obj2();
15  结果是   :   1   2   1   2
```

闭包的优点：

优点：不产生全局变量，实现属性私有化。

缺点：闭包中的数据会常驻内存，在不用的时候要删掉否则会导致内存溢出。

## 1.2 对象（object）

对象是什么？

基本数据类型      string    number    boolean    null    undefined

Array      对象

对象数据类型：      对象就是带有属性和方法的 数据类型

```
var num = 10; // 变量
var arr = []; // 数组
arr.index = 10; // 数组 arr 的一个 index 属性
```

但是有个问题， 我们想要某些属性或者方法的时候，用数组不合适。

arr.lenght .

我们想要自己 id 属性和方法 。 要求这个一定是个对象才行。

### 1.2.1 声明对象

我们有两种声明对象的方式.

```
var obj = new Object();
```

但是我们更提倡用第二种方法： 字面量式声明对象

```
var obj = {};
```

```
var obj = {}; // 声明对象
obj.name = "刘德华"; // 属性
obj.age = 55;
obj.showName = function() { // 声明方法 方法一定带有 ()
    alert("俺是刘德华");
}
obj.showAge = function() {
    alert("俺今年 18 岁");
}
```

### 1.2.2 使用对象

```
console.log(obj.name); // 调用属性
console.log(obj.age);
obj.showName(); // 调用方法
obj.showAge();
```

### 1.2.3 面向对象

面向对象里面：类和对象

类是对象的抽象，而对象是类的具体实例

一切事物皆对象 JavaScript 一切皆对象

## 类和对象

### 对象 女朋友

小胡 你有对象了吗?                      泛指          女朋友

旁边坐的那个女生是你对象吗?          特指          某一个女朋友

大家喜欢吃苹果吗?                      泛指          类

你手里的那个苹果甜吗?                  特指          对象      (具体的实例)

### 面向对象的特性

#### 1. 抽象

抽象就是忽略一个主题中与当前目标无关的那些方面，以便更充分地注意与当前目标有关的方面。

#### 2. 封装

封装是把过程和数据包围起来，对数据的访问只能通过已定义的界面

#### 3. 继承

#### 4. 多态

多态是指两个或多个属于不同类的对象，对于同一个消息（方法调用）作出不同响应的方式

## 面向对象编程中常见概念深入解析



### 1.2.4 new

新的 我们经常利用 `new` 关键字 去声明新的对象

javascript

`new` 运算符的作用是创建一个对象实例。这个对象可以是用户自定义的，也可以是带构造函数的一些系统自带的对象。

`new` 关键字可以让 `this` 指向新的对象

所谓"构造函数"，其实就是一个普通函数，但是内部使用了 `this` 变量。对构造函数使用 `new` 运算符，就能生成实例，并且 `this` 变量会绑定在实例对象上。

### 1.2.5 prototype

共同的 相同的 部分

主要解决：函数因为使用非常非常多，重复执行效率太低。

```
Person.prototype.showName = function() { // 用的共同的父  
亲  
    alert("我的名字是" + this.name);  
}
```

类.prototype.方法 = function() {} 具体格式

可以把那些不变的属性和方法，直接定义在 `prototype` 对象上

使用方法：

类名.prototype.方法