深入剖析JavaScript作用域OR闭包

铺 垫：

1、作用域是指程序源代码中定义变量的区域。作用域规定了如何查找变量，也就是确定了当前执行代码对变量的访问权限。

2、执行上下文栈（Execution Context Stack，ECS）。

3、当JS解析器执行一段可执行代码（Executable Code）时,会创建对应的执行上下文（Execution Context）。并且压入执行上下文栈。

4、变量对象（Variable Object）。

、全局上下文下的变量对象。

全局对象（Global Object）。（当js编译器开始执行的时候会初始化一个Global Object用于关联全局的作用域。全局上下文中的变量对象就是全局对象）

、函数上下文下的变量对象。（在函数上下文中，我们用活动对象来表示变量对象）

活动对象（Activation Object）。（只有进入到一个执行上下文中，这个执行上下文的变量对象才会被激活，也就是活动对象是在进入函数上下文时被创建的，它通过函数的arguments属性初始化。）**函数上下文的变量对象初始化只包括Arguments对象。**

.1执行上下文的代码会分成两个阶段进行处理（1、进入执行上下文；2、代码执行）

1、进入执行上下文时会给变量对象添加形参、函数声明、变量声明初始的属性值。

2、在代码执行阶段，会再次修改变量对象的属性值

5、作用域链（Scope Chain）

作用域链：它在解释器进入到一个执行环境时初始化完成并将其分配给当前执行环境。每个执行环境的作用域链由当前环境的变量对象及父级环境的作用域链构成。

下面，从一个函数的创建和激活两个时期来解释作用域链是如何创建和变化的。

函数创建：函数的作用域在函数定义的时候就决定了（函数有一个内部属性[[scope]]，当函数创建的时候，就会保存所有父变量对象到其中）。

函数激活：进入函数上下文，创建AO对象后，就会将活动对象添加到作用域链的前端。

一、执行全局代码，创建全局执行上下文，全局上下文被压入执行上下文栈。

ECS = [globalcontext]

二、全局上下文初始化，同时a函数被创建，保存作用域链到函数的内部属性[[scope]]。

三、执行a函数。创建a函数执行上下文，a函数执行上下文压入执行上下文栈。

ECS = [acontext,globalcontext]

四、a函数执行上下文初始化

1、复制函数[[scope]]属性创建作用域链。

2、用arguments创建活动对象

3、初始化活动对象，即加入形参。函数声明、变量声明

4、将活动对象压入a作用域链顶端