**Scope Chains and Identifier Resolution** 作用域链和标识符解析

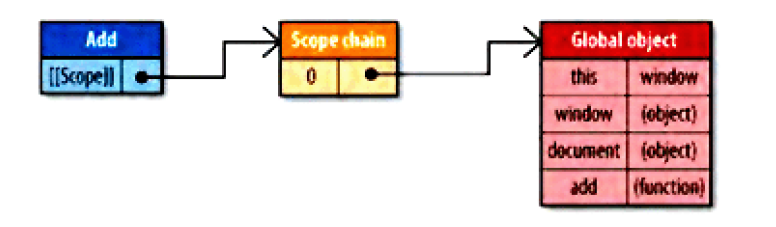
function add(num1, num2){

var sum = num1 + num2;

return sum;

}

当add()函数创建后，它的作用域链中填入一个单独的可变对象，此全局对象代表了所有全局范围定义的变量。此全局对象包含诸如窗口、浏览器和文档之类的访问接口。图2-2 指出它们之间的关系（注意：此图中只画出全局变量中很少的一部分，其他部分还很多）。



add 函数的作用域链将会在运行时用到。假设运行下面的代码：

var total = add(5, 10);

运行此add 函数时建立一个内部对象，称作“运行期上下文”。一个运行期上下文定义了一个函数运行时的环境。对函数的每次运行而言，每个运行期上下文都是独一的，所以多次调用同一个函数就会导致多次创建运行期上下文。当函数执行完毕，运行期上下文就被销毁。

一个运行期上下文有它自己的作用域链，用于标识符解析。当运行期上下文被创建时，它的作用域链被初始化，连同运行函数的[[Scope]]属性中所包含的对象。这些值按照它们出现在函数中的顺序，被复制到运行期上下文的作用域链中。这项工作一旦完成，一个被称作“激活对象”的新对象就为运行期上下文创建好了。此激活对象作为函数执行期的一个可变对象，包含访问所有局部变量，命名参数，参数集合，和this的接口。然后，此对象被推入作用域链的前端。当作用域链被销毁时，激活对象也一同销毁。图2-3 显示了前面实例代码所对应的运行期上下文和它的作用域链。

