**闭包：**

* 理论角度

**闭包 = 函数 + 函数能够访问的自由变量**

自由变量是指在函数中使用的，但既不是函数参数也不是函数的局部变量的变量。

var a = 1;

function foo() {

console.log(a);

}

foo();

* 实践角度

1. 必须有两个执行上下文（通俗的理解就是2个函数A，B）
2. 在B中引用了A中的变量且A函数执行返回了B函数
3. A执行结束后，以某种形式保留了A的返回结果（返回结果通常是函数）
4. B函数的执行就形成了闭包

闭包的作用：

保留一个执行上下文中的变量，即使该上下文的函数已运行结束

function checkscope(){

var scope = "local scope";

function f(){

return scope;

}

return f;

}

var foo = checkscope();

foo();