1. **输出结果是多少？为什么是这个结果。**

结果：6，5

原因：调用fn2(obj.getA)，实际上传入的是fn1的引用。所以可以看为fn2(fn1)，而传入函数引用直接在fn2函数体内直接调用，相当于window.fn()，所以此时fn1函数体内的this指向window，所以this.a指向window为5。而因为在输出console.log(a)之前函数体内有使用var定义了局部变量a为6，所以打印a为6。

1. **输出结果是多少？为什么是这个结果。**

结果：3，5，5，3

原因：obj1.foo()和obj2.foo()直接通过对象.函数隐式绑定this，this指向最后调用的对象即obj1和obj2，所以输出3和5。obj1.foo.call(obj2)和obj2.foo.call(obj1),使用了call显示绑定this为obj2和obj1，所以输出5和3.

1. **输出结果是多少？为什么是这个结果。**

结果：undefined，6

原因：首先在全局作用域执行test(5)，此时test函数内部this指向window(默认绑定)，所以第一次window.x = 5，最后函数返回this(即window)赋值给变量x，而var声明的x也会挂载在window上覆盖原来的x，所以此时第二次window.x指向全局对象window。接下来执行test(6)，this任然是默认绑定为window，所以此时第三次this.x赋值为6再次覆盖原先的window.x。最后返回window对象给变量y。综上所述此时window.x为6，window.y指向window对象，所以最后输出x.x

为undefined，y.x为6。