

1. 分析控制台输出情况，简述过程。

```
var a = 12,
    b = 13,
    c = 14;

function fn(a) {
    var b = (c = a = 20);
    console.log(a, b, c);
}

fn(a);
console.log(a, b, c);
```

输出答案：20, 20,20
12,13,20

分析过程：

全局变量中： a=12 b=13 c=14;

执行函数 fn，实参传形参，所以 fn 中的变量 a 是局部变量，并且初始值为 12，因为 12 是基本类型，全局变量 a 和局部变量 a 传值结束后再也没有关系了。

var b=(c=a=20) 等价于 var b;（使用 var 声明，为局部变量）

c=20;（在函数中没有通过 var 声明的,是对全局变量进行重新赋值，全局变量由 14 变为 20）

a=20;(局部变量由原来的 12 变为 20)

b=20;（局部变量的值为 20）

在 fn 中的 console.log(a,b,c)输出局部变量 a=20;局部变量 b=20;全局变量 c=20;

全局中的 console.log(a,b,c)输出全局变量 a=12;全局变量 b=13;全局变量 c=20

2. 分析下列代码中两种方案是否可以修改元素的字体颜色，如果不可以是为什么。

```
<body>
  <div id="box">文本</div>
</body>
<script>
  var box = document.getElementById("box");
  box.style.color = "red"; //通过这行代码可以修改 box 元素的字体颜色

  //问题：那么下列的写法是否可以修改元素的字体颜色，如果不可以是为什么？
  //第一种方案：
  var AA = box.style;
  AA.color = "red";

  //第二种方案：
  var BB = box.style.color;
  BB = "red";
</script>
```

答案：第一种方案可以，第二种方案不可以。

分析过程：从题目中可以修改 box 的字体颜色必须修改 box 对象上的 style 属性所对应的对象上的 color 属性值。

第一种方案：box.style 是一个对象，赋值给 AA，在内存中 AA 和 box.style 指向同一块内存，对 AA 对象的 color 属性进行修改，即是对 box.style 对象上的 color 进行修改。

第二种方案：box.style.color 是一个基本类型，赋值给 BB，此时 BB 和 box.style.color 在内存中没有任何关系。所以对 BB 修改，与 box.style.color 没有影响。

3. 分析控制台输出情况，简述过程。

```
var num = 1,
    obj = { value: "111" };
function fn(obj) {
    num--;
    obj.value = "222";
    return num;
}
console.log(num && fn(obj) && num++);
console.log(num);
console.log(obj.value);
```

答案: 0
0
“222”

分析过程:

全局变量 num 的值为基本类型，并且初始值为 1；全局变量 obj 的值为引用类型，其中的 value 属性为“111”；

num && fn(obj) && num++ 逻辑与运算符，先看最左边的操作数，此时 fn 并没有执行，num 为 1，所以等价于 1&&fn(obj)&&num++；
1 的布尔类型为 true,所以继续看右操作数，此时 fn(obj)执行。在 fn 执行的时候全局变量 obj 传参给形参局部变量 obj，因为 obj 是引用类型，所以全局变量 obj 和局部变量 obj 在内存中执行同一个堆内存空间。fn 函数体中 num--，因为 fn 中没有局部变量 num，所以是访问的全局变量 num,此时全局变量 num 变成了 0；
修改局部变量 obj 的 value 属性，等同于修改全局变量 obj,所以此时全局变量 obj 和局部变量 obj 所指向的空间 value 的值由“111”变为“222”。
最后 return num; fn 中并没有局部变量 num，仍然去全局变量中找，找到全局变量 num，并且此时全局变量 num=0；所以 fn(obj)表达式的值为 0，此时等价于 1&&0&&num++，又等价于 0&&num++；
根据逻辑运算短路原则，此时忽略对右操作数的判断，所以 num++并没有执行，所以全局变量 num 仍然为 0。到此为止 num && fn(obj) && num++ 的输出结果为 0；

console.log(num); 输出全局变量 num 的值，根据上面分析可得知结果为 0；
console.log(obj.value); 输出全局变量 obj 对象的 value 值，根据上面分析可得知结果为“222”；