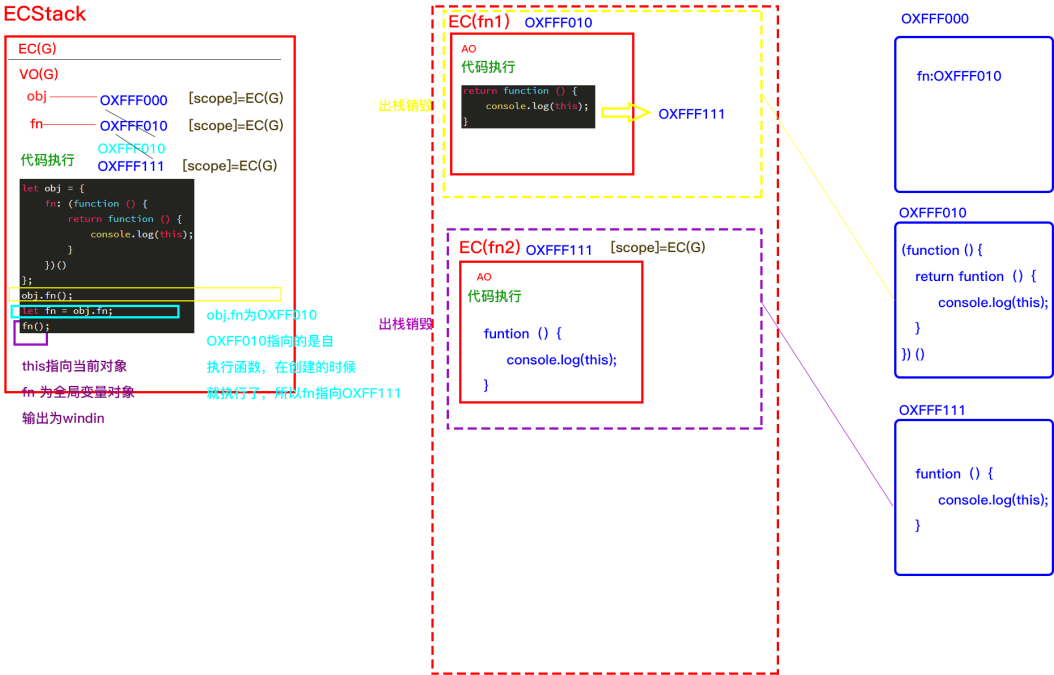


# 第一周正式课-课后作业 (3)

1.

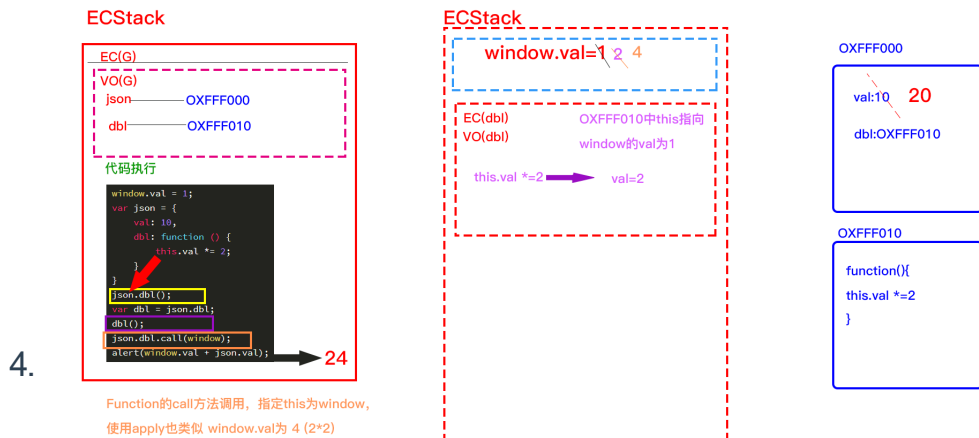


2.

“undefined”  
'language'

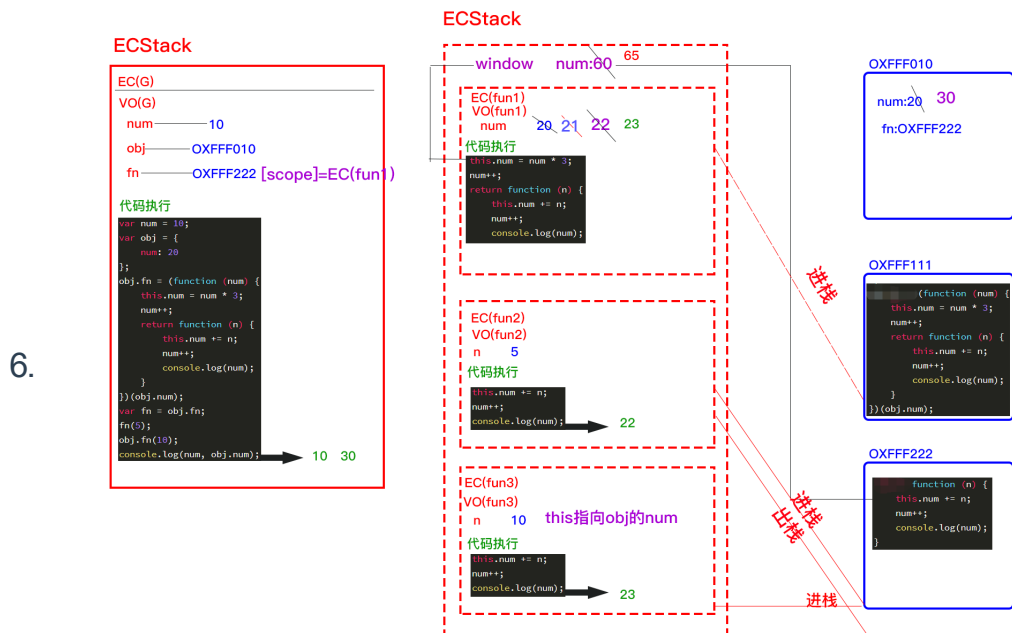
3.

'window'



5.

'12'



- 7.
- 不能实现
    - 现在的效果是都会输出5
    - 修改方法有三种
      - 把for循环中的var改成let 形成块级作用域

- 通过自定义属性来存储当前循环的索引

```
for(var i=0;i<l;i++){  
  l[i].index = i;  
  inputs[i].onclick=function(){  
    alert(this.index);  
  }  
}
```

- 通过形成闭包作用域

```
for(var i=0;i<l ; i++){  
  inputs[i].onclick=(function(i){  
    alert(i);  
  })(i)  
}
```

8.

我理解的闭包是在浏览器的大环境下 怎么能实现需求功能，怎么能有秩序，互不干扰的执行代码。这就需要闭包的私有作用域。

- 优点

- 保护作用：在这个上下文中会有一些私有的变量AO(XX)，这些私有的变量和外界的变量不会冲突（互不影响）
- 保存作用：某些情况下，上下文中的某些内容被外部占用后，当前上下文并不会出栈销毁，这样就会把上下文的一些信息存储起来

- 缺点

- 1.由于闭包会使行函数中的变量都保存在内存中，内存消耗很大，所以不能滥用闭包，否则会造成网页的性能问题，在IE中可能导致内存泄露。
- 2.闭包会在父函数外部，改变父函数内部变量的值。

9.

- VAR存在变量提升 而LET不存在

- LET是不允许重复声明的,而var允许重复声明
- 用VAR声明的变量即是全局变量,也相当于给GO(window)设置了一个属性,而且两者建立映射机制;但是用LET声明的变量仅仅是全局变量

10.

```
for (let i = 0; i < 10; i++) {  
  setTimeout(() => {  
    console.log(i);  
  }, 1000);  
}
```

11.

```
b() { b = 20;console.log(b);} 10
```

```
var b = 10;  
(function b() {  
  var b = 20;  
  console.log(b);  
})();  
console.log(b);
```