

# 第一周正式课-课后作业（1）

1

- `undefined`

13

---

- **报错：** `a is not defined` 后面的代码不会执行
- 

- **报错** `Uncaught ReferenceError` `let` 没有变量提升，而且必须要求等`let`语句执行完之后，变量才能使用
- 

2

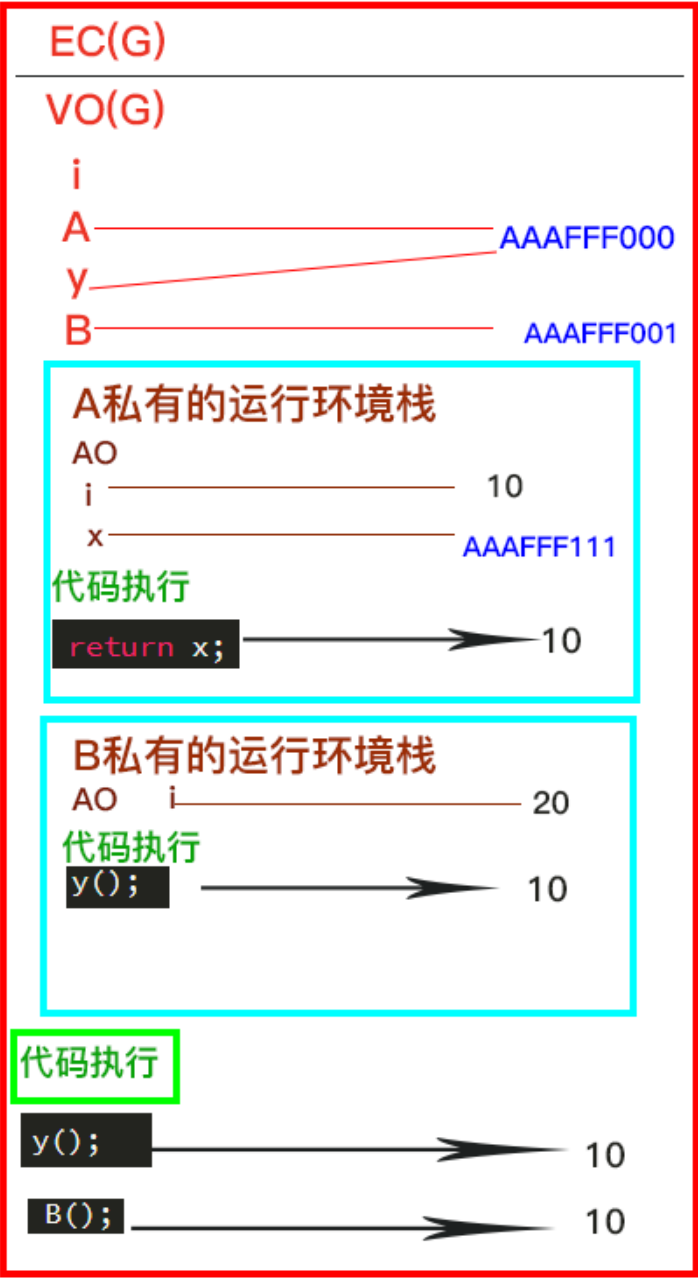
# ECStack



# ECStack



# ECStack



全局fn堆内存AAAAFF000

```
var i = 10;
function x() {
  console.log(i);
}
return x;
```

全局fn堆内存AAAAFF001

```
var i = 20;
y();
```

A私有函数x堆内存AAAAFF111

```
console.log(i);

return x;
```

# ECStack

window

obj2 ——— AAAFFF111  
a2 ——— 1

EC(G)

VO(G)

a ——— 1  
obj ——— AAAFFF111  
fn ——— AAAFFF000

fn私有的运行环境栈

AO  
a2 ——— 1

代码执行

```
obj2 = obj;  
a2 = a;  
obj2.name = "jack";
```

给GO添加一个obj2的属性  
给GO添加一个a2的属性

全局fn堆内存AAAFFF000

```
var a2 = a;  
obj2 = obj;  
a2 = a;  
obj2.name = "jack";
```

全局obj堆内存AAAFFF111

```
"name": "tom"  
"jack"
```

代码执行

```
var a=1;
```

定义

```
var obj = {  
  "name": "tom"  
}
```

定义

```
fn();
```

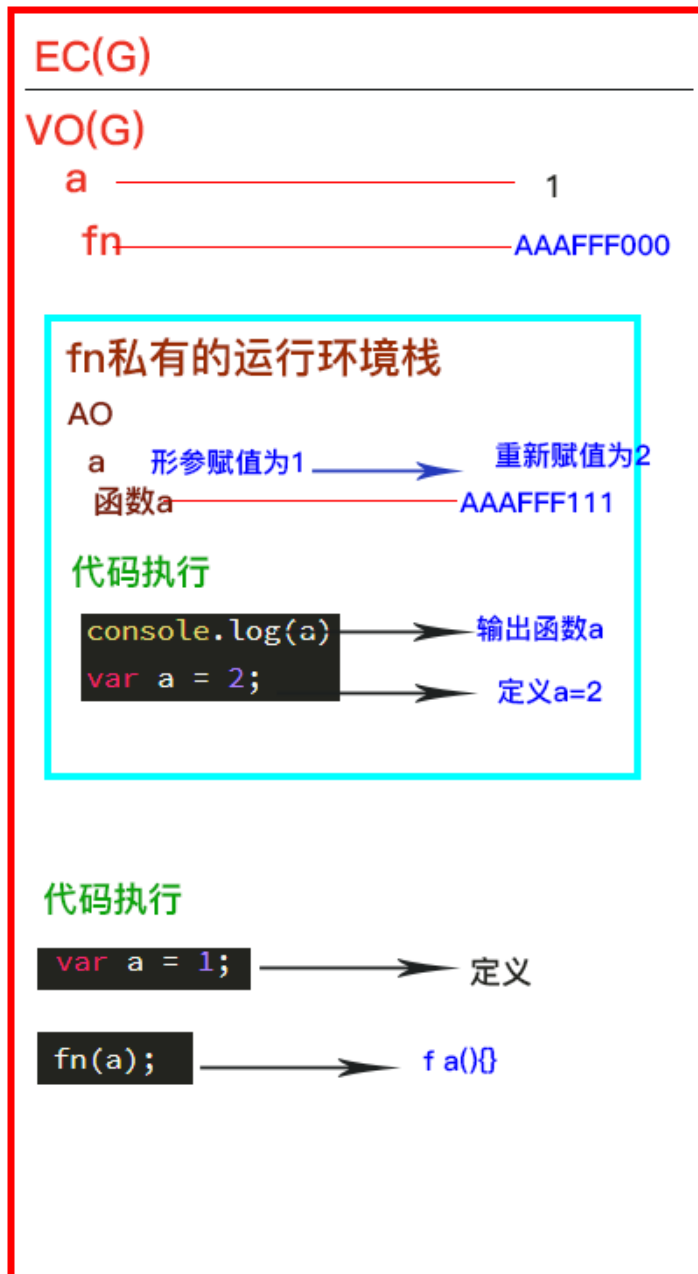
```
console.log(a);
```

1

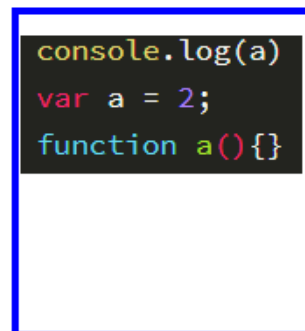
```
console.log(obj);
```

"name": "jack"

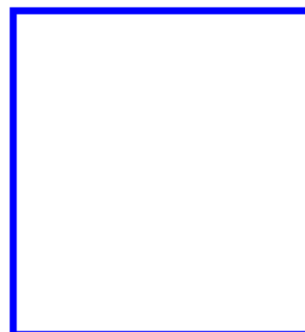
# ECStack



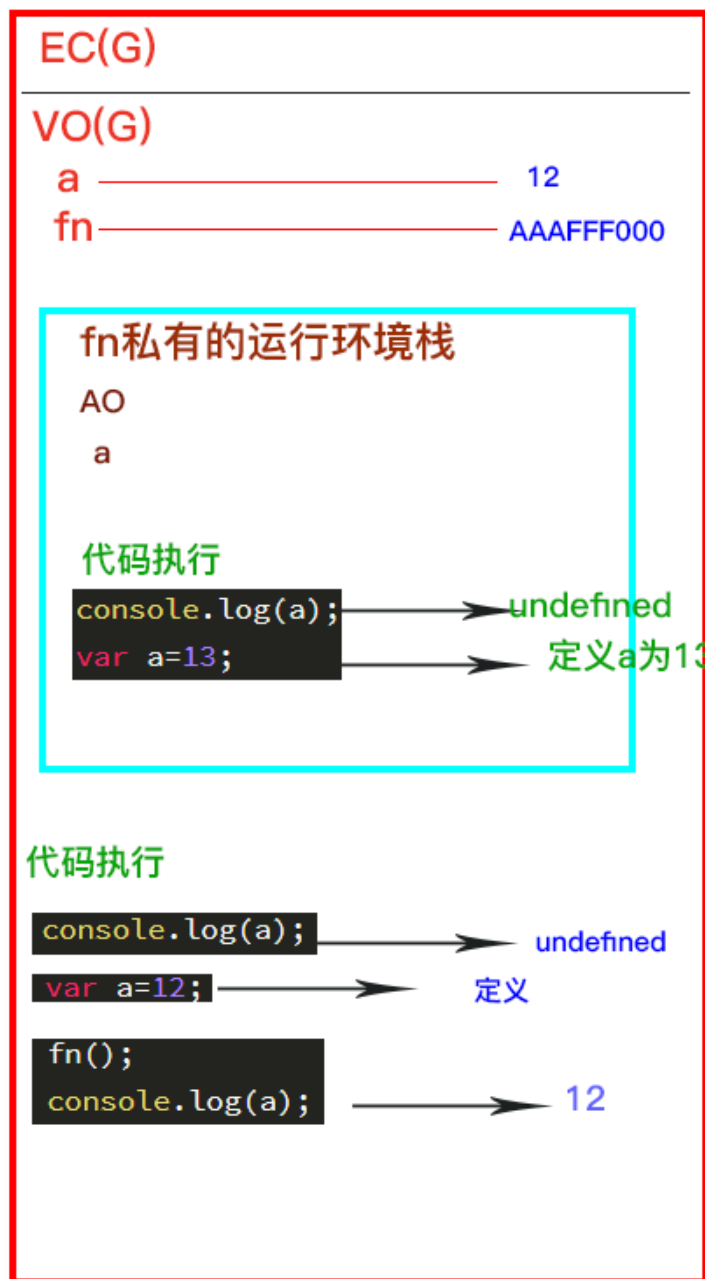
全局fn堆内存AAAFFF000



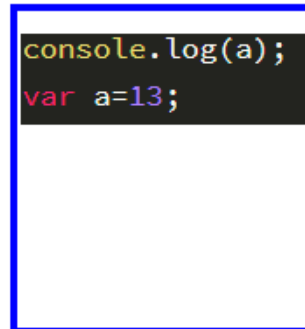
fn私有函数堆内存AAAFFF111



# ECStack



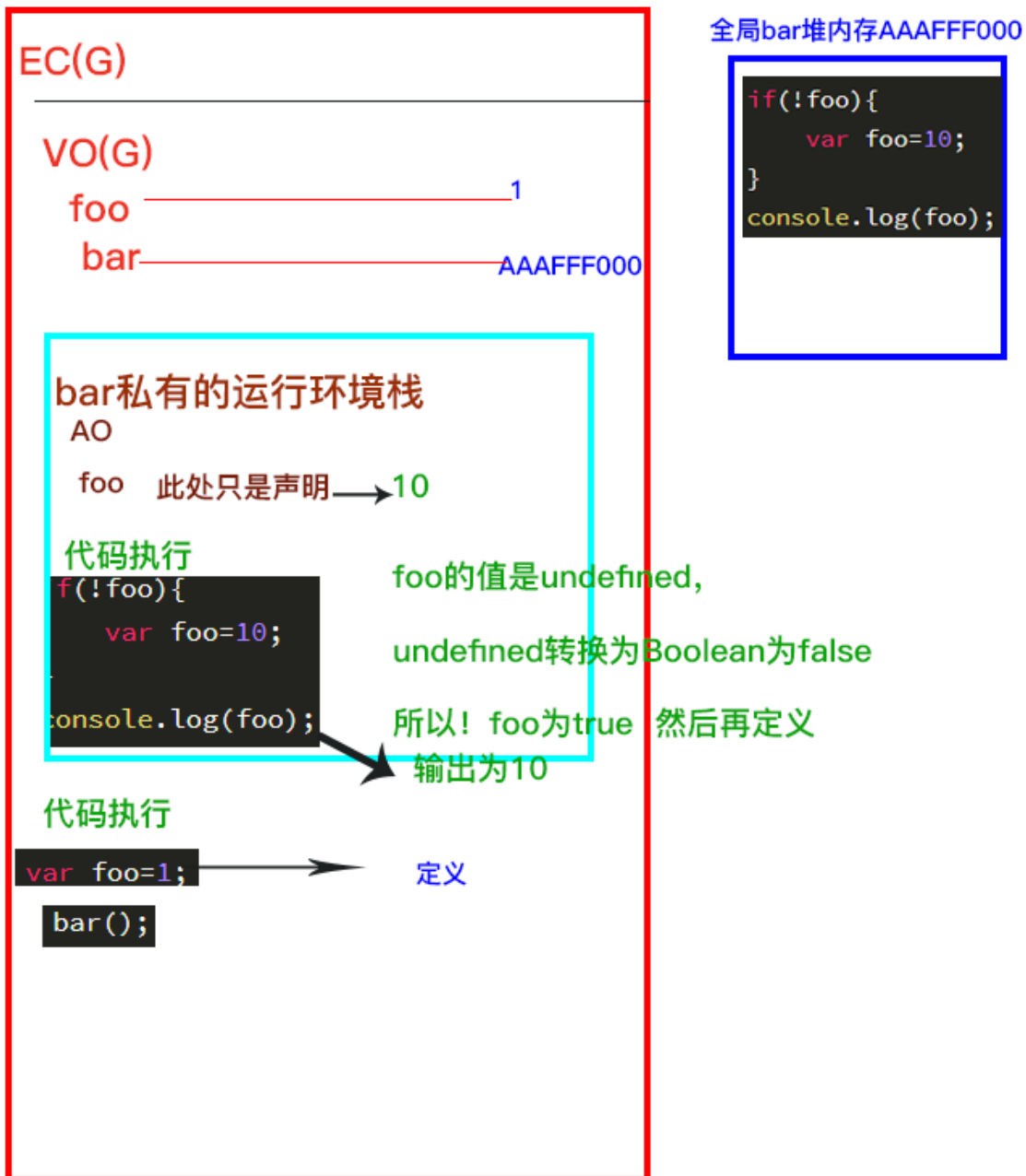
全局fn堆内存AAAFFF000



第二个 undefined 12 13 a=13 改变全局变量VO中a的值 然后相应的会修改VO中的a的值

第三个 直接报错 a没有声明

# ECStack



9

- 10
- 11
- 3

10



undefined

11

- 3
- 10
- **undefined** 因为函数没有返回值 所以a没有被定义

12

“hello”

“hello”

“hello”

自执行函数在创建的时候就已经执行了，因为`var foo = “hello”`在自执行函数的上面，声明和定义已经完成，传入函数中就都是“hello”，`||`表示如果`foo`没有被定义则为“world”