Day-17()

Saturday, 1 February 2025 5:06 PM

Mentality Approach

Lexical Scope = Local Memory + Parents LS

LM + PLS

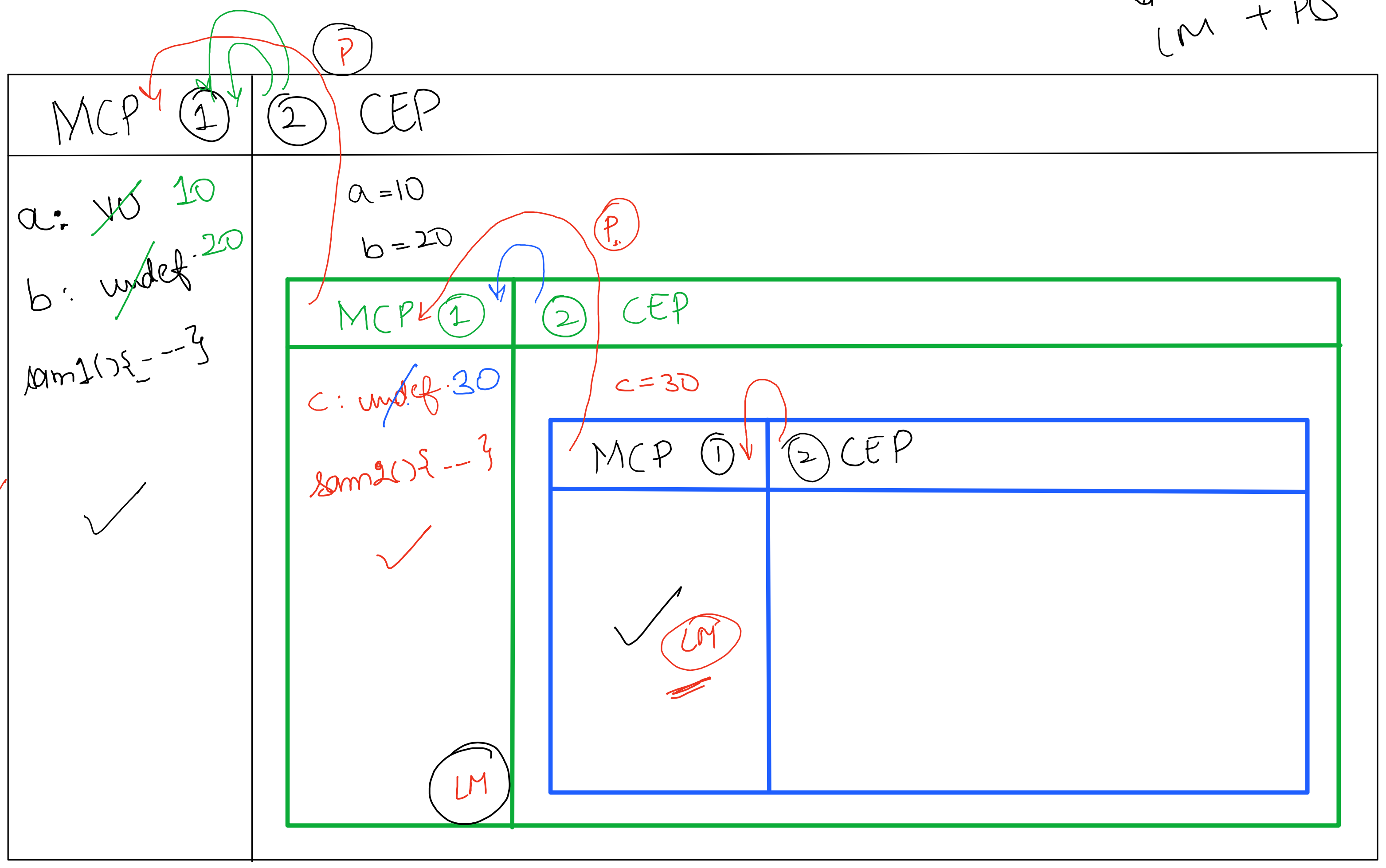
LM + PLS

```
let a = 10;
var b = 20;
function sam1(){
    var c = 30;
    function sam2(){
        console.log(a);
        console.log(b);
        console.log(c);

    }
    sam2()
}
sam1()
```

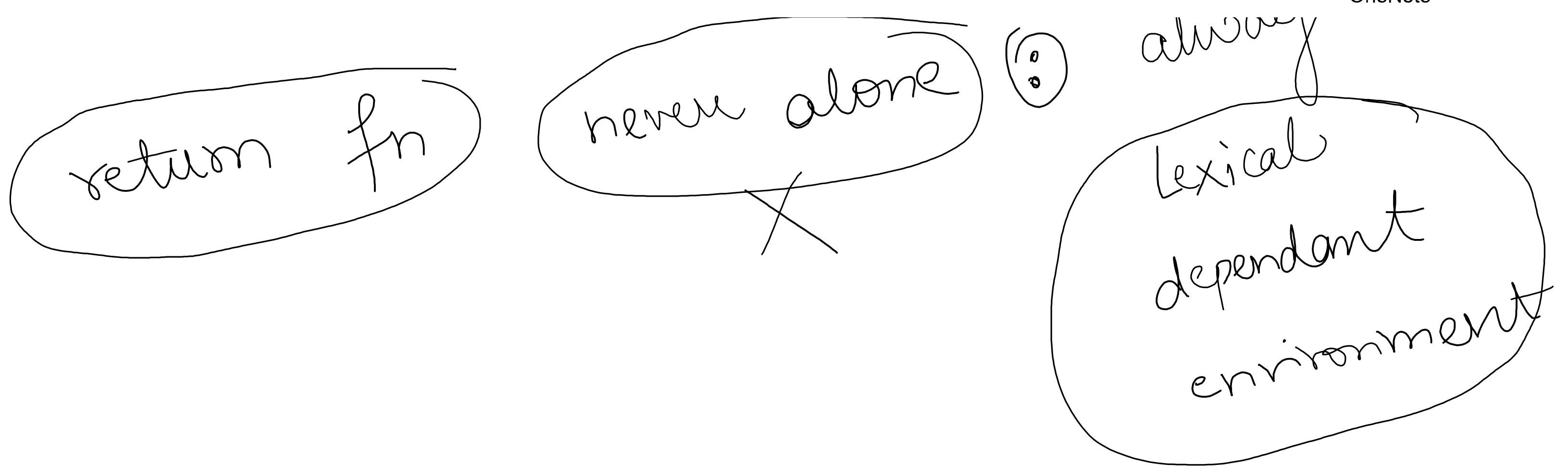
10
20
30

outer

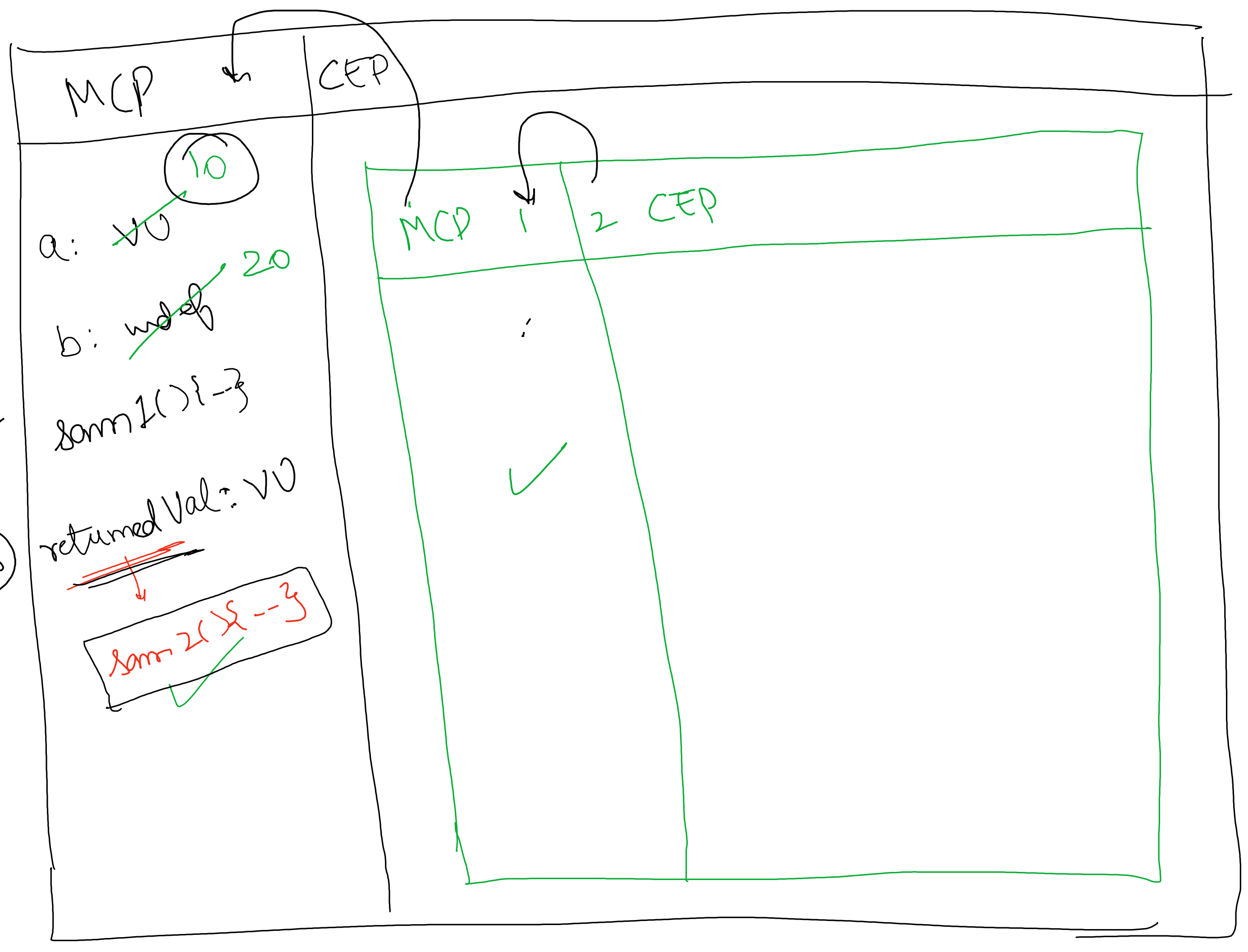


closures

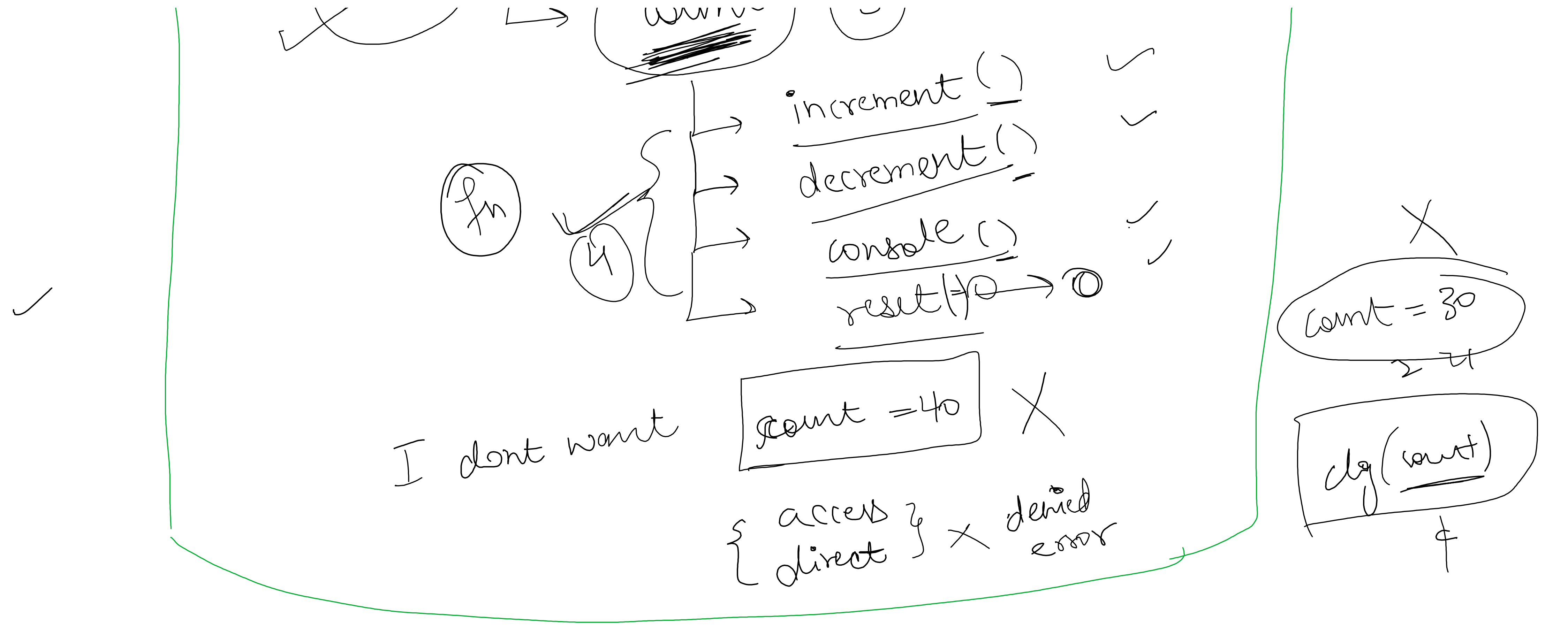
returns with



```
let a = 10;
var b = 20;
function sam1(){
    var c = 30;
    function sam2(){
        console.log(a);
        console.log(b);
        console.log(c);
    }
    return sam2;
}
let returnVal = sam1();
console.log(returnVal);
returnVal();
```



```
10
20
error
```



Higher Order ① fn (HOF)

Callback fn

HOF

① → if (A) fn can have another fn (B) as an argument
HOF

② if (A) fn can return another fn (B) from it
then its HOF.

→ HOF ✓

→ First class fn ✗

```
function outer() {
    - -
    - - -
    return inner;
}
```

✓

✓

HOF ✓

```
let outer = function() {
    - - fn
    - .
    return inner
}
```

first class fn ✓

✓

Array Methods

1. push()
   pop()
   shift()
   unshift()

✓

let arr = [10, 20, 30]

$\boxed{\text{arr.length}}$ = 3

= arr.push(10)

(20)

= arr . [illegible]