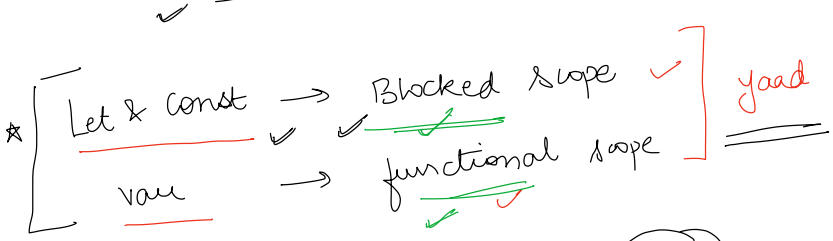
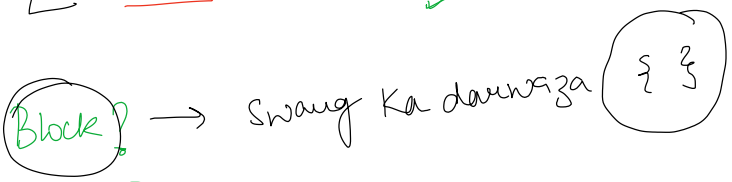
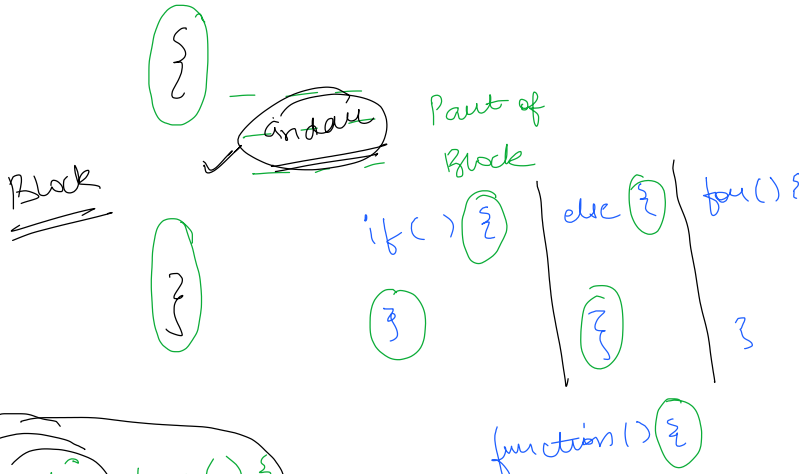
Day23(hof , callback , scope , let var and const deep dive, block , script)

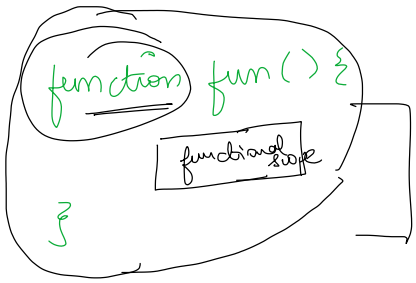
Wednesday, 5 April 2023 7:37 PM

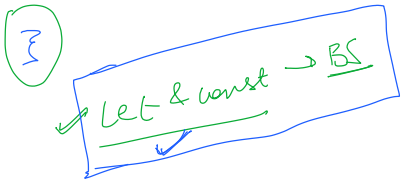
Let, var & const declaration





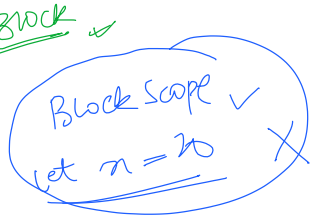






GEC | EC

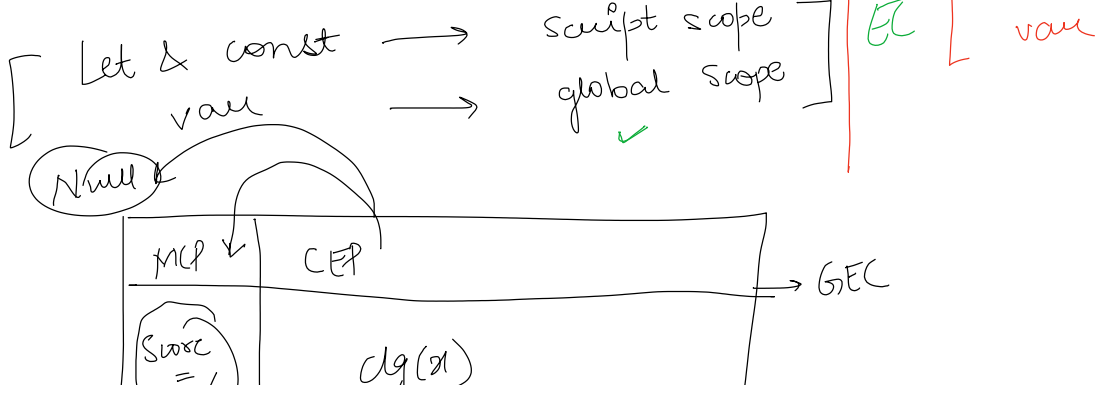
```
1  // scope
2  let score = 50;
3  if(score > 33){
4      let x = 20;
5  }
6  console.log(x);
```

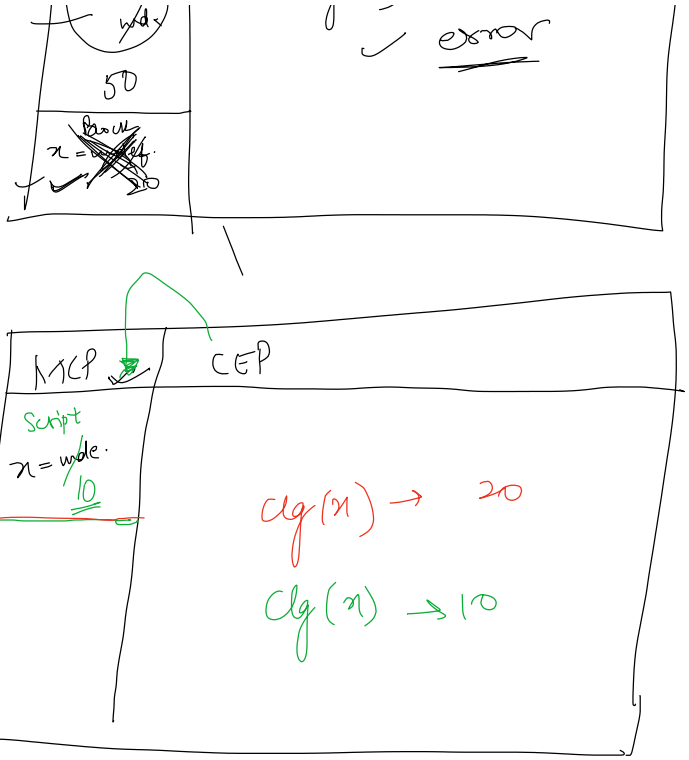


x ✗ error

if declared variable in GEC with :

Important : GEC






```
1  let x = 10;
2  {
3      let x = 20
4      console.lo
5  }
6  console.log(
```

20
10

When you access a variable / function even before its declaration that concept is called hoisting.

hoisting

var → Normal

let, const → DTZ

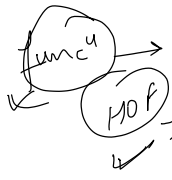
Higher Order function

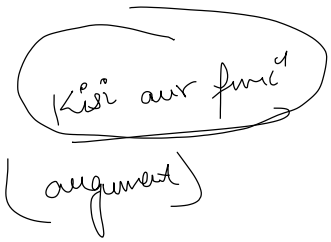
functions that operate on other functions either by

1. taking them as an argument (or)
2. returning them (func°)

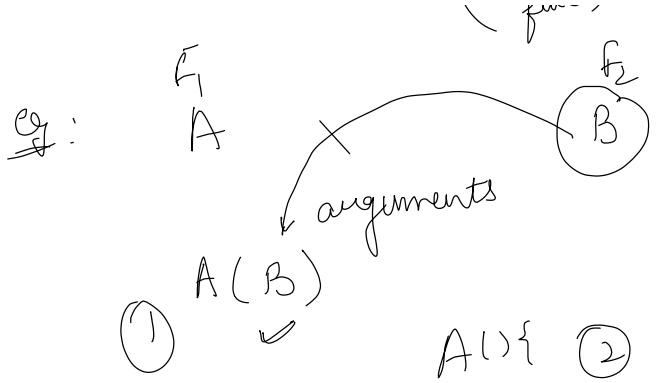
are called HOF.

Hindi

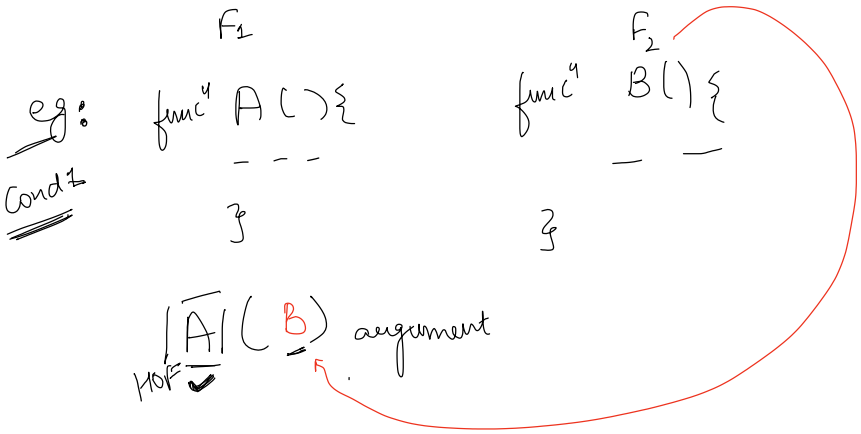




return (poora ka poora func°)




```
A(){ (2)
return B
}
```



or

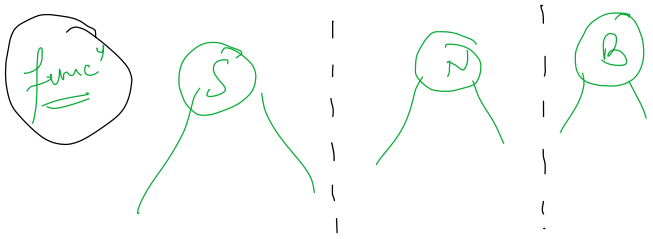
Cond2

```
HOF
function A(){
    function B(){
    }
    return B;
}
```

returning a func^n

Real life

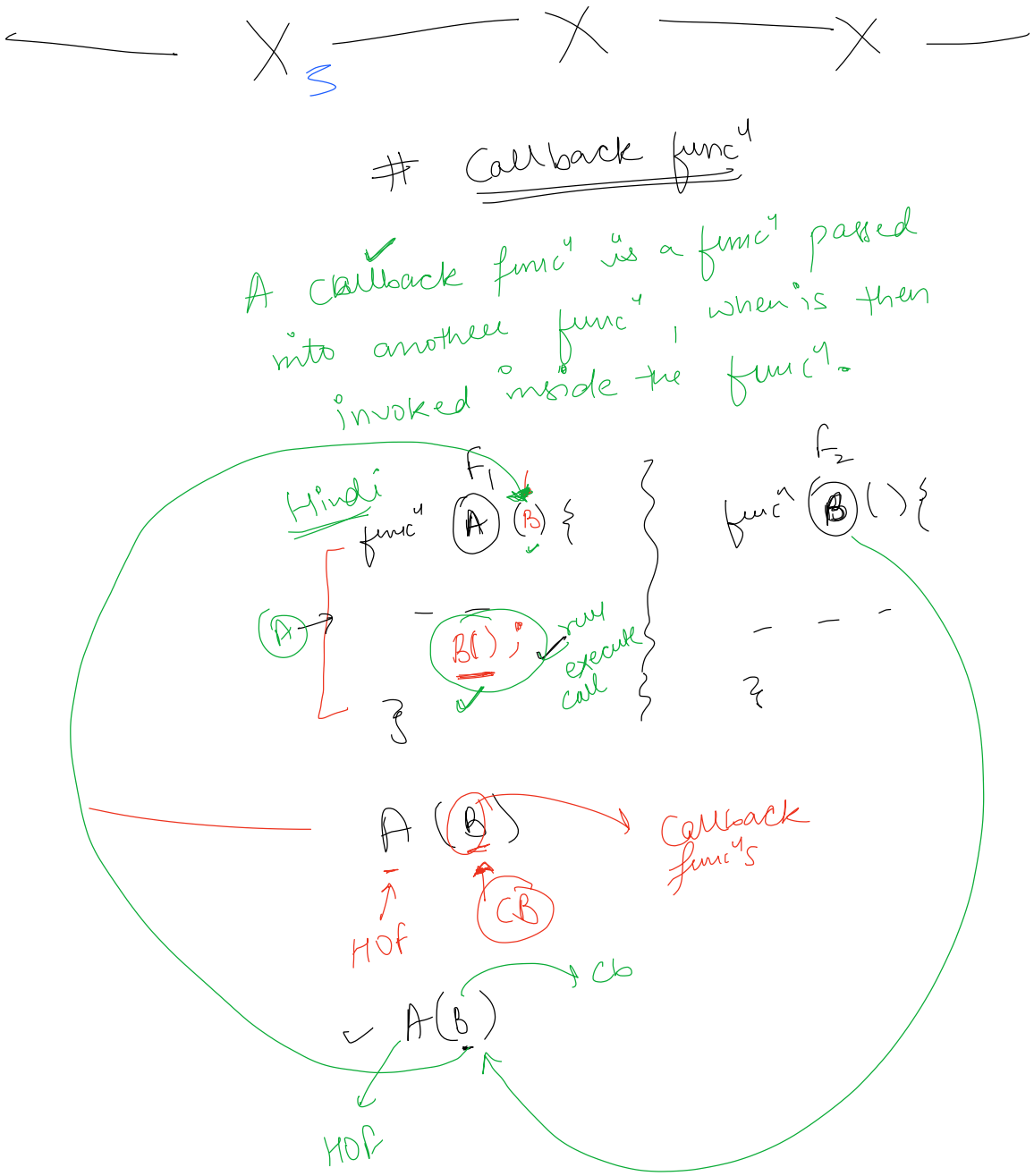
Let arr = ['Sam', 'Samantha', 'Bob']



f(S)

f(N)

f(B)



A callback function is a function passed as an argument into another function, but the only condition is that the passed function (means callback fun.) must be call inside the function which accept it as parameter.

Let and Const => Script scope (if globally declared => GEC).
Let and Const => Block scope (if declared inside block).
Eg : Block => while , if, for, {} etc.

Var => Global scope (if globally declared => GEC).
Var => Functional / Local scope (if declared inside function).
Otherwise var has global scope.
All variable have local / functional scope inside function
i.e var , let and const