Closures
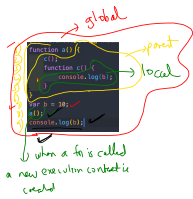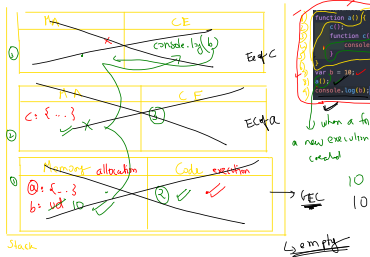
When a function is returned from another fn along
with it's lexical scope it is forms a closure

Scope, Scope chain, lexical env.

var a:
└> where can this variable/fn be accessed in
   the code.

global
→ parent
→ local

when a fn is called
a new execution context is
created

10
10

→ GEC

Stack

└> empty

1) when fn is invoked, new E.C is created

2) in top most EC: b is not define in local scope,
   we search for b in parent's scope
   this procedure goes on until global scope is reached

Reference error
b is not defined

lexical env = local memory + lexical env of parent (a) → ③
of c

lexical env of a = LM of a + LE of parent (gs) → ②

LE of global = LM of global → ①

great gf

gf

lexical env

scope

① + ②

PR → ①+② ③ ← COLE
④

scope chain ①+②+③+④

└> chain of all LE & the parent references
   COLE   PR

Memory   Code
              Code
b: vd 10   Mem   Code
a: { }   c: {}   c
                      console.log(b)

Consol
10