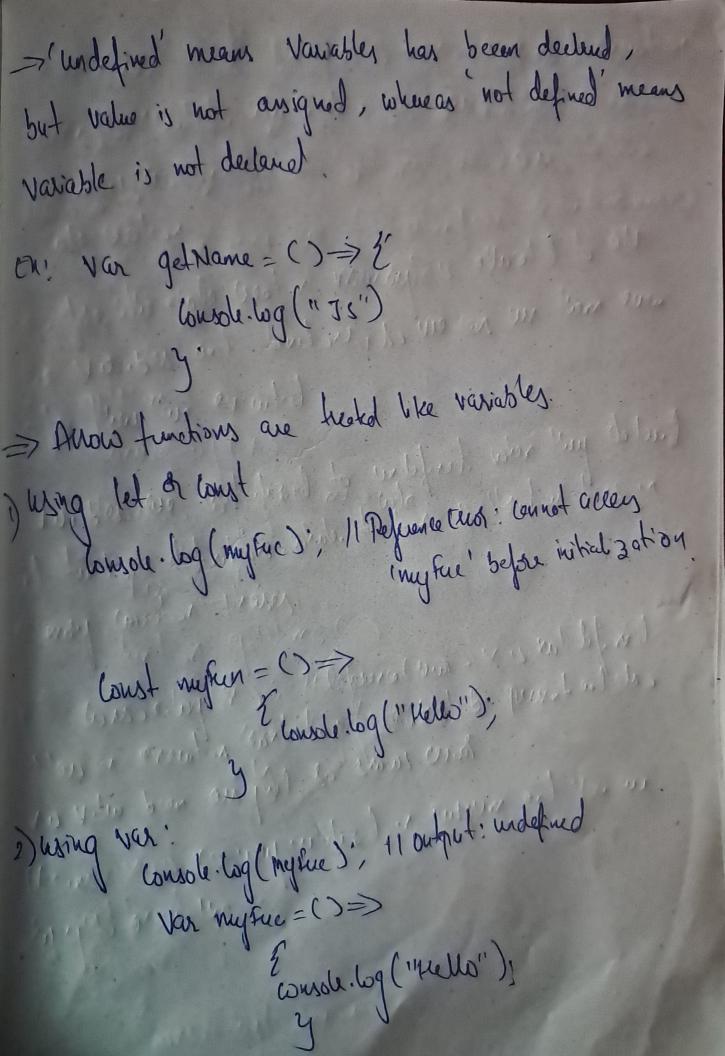
Hoisting - Hoisting in a javascript is a mechanism where Variables and function declarations are noticed to the top of their containing scape (either global or turction scape) before the execution of the code. En: getName (); Courde log (x), busole log (get Name); Var x=7; Juntion getnome () Course log ("Javascept") Op: Javasaipt undefined. get Name () 2 Cousole log ("Javascupt")

7 Whenever a JS Program is run, a global eneution Content is created, which has a phases 1118 , 98. 1814 memory acation 2) Code Creation -> Before the Code Creation, memory is allocated to all the variables and functions, to me can me them ever before they are exembed. > Variables are initialized to undefined when they declared and function definition, whole trustion is stored as it is in mendy coreation phases. Now in enewtrin content, x will shoot be present and also no value been assigned, so shows cred. Means 'X' is no way intulized in purposen, and you are tying to access the x on stop some



> Variable declarations are scanned and are mode undefined - function declaration are scanned and are mede available.

Horsting to let and court

> variables declared with let and court are bristed,

but they are xinitialized. This mean they cannot be

accend before their declaration, leading to a Represent two.

They exist in a temporal dead zone, Ist is a period in

They exist in a temporal dead zone, Ist is a period in

Swhere variables declared with let and court cannot be

account before their declaration.

Console. loge (5)

Console. loge (5)

Console. loge (5)

Console. loge (5)

11 Reference Error: commet accent b' before intelization

Function Explessions function expressions are not hoisted in the same way as tunction declaration. a only the variable declaration is hoisted, but the turntary itself it not available until the code reacher the en: bomble log(my hie); 110/p:- undefined

Vay myrue = function () Comple log ("kello"); Console log (my Fuc); 1/Reference tust: connot access

Ny fue before introlizator let nyfice = function () 2 comol·log ("Hello").

let, a=10; (ousole.log (a); is same as global object > window object M console window . b window a Because variables using let & went are stoled in different memory location location.

Tet & lough variables are not attached to global object. > At global, this = window IN Course this b this a = undefined syntantud: a has already been delend lefa=to leta=10 7 me get let a= to -7 me get some syntanteus.

Var a=10; var a=10; - me won't get any error.

Const 5;

5 = 10;

The get syntan that: Missing initializes in worst declaration.

-> Means we have to intalize the value in same line



> Syntan EMOR is similar to compile Emol > Reference Emol falls under time and

-> Syntan Euro - Violation of Js syntan > type Euse > while trying to re-initialize which is not
> Reference Euse - while trying to accent variable which is not
there in global mendy.