* Every thing is happened in “execution context” execution context has two parts
  + - Memory component/ Variable Environment
    - code component/ thread of execution
  + Memory component/ Variable Environment:

All variables are declared in this part in the form of key and value pair.

* + code component/ thread of execution:  
     All code written and execute into this part.
  + **Memory part**
    - First jascript read the whole program and allocate the memory to then all the variables and functions into the memory part before start the execution of code component, in variables javascript stores “undefined” and in the case of function javascript store whole copy of code.
  + Code Execution is the second phase
  + **Call Stack:** 
    - call stack contain execution contexts, at bottom there is global or parent execution context.
    - It maintains the execution order or execution contects.
  + **Hoisting:**
    - Access the function or variable before initialize it is known as hoisting.
    - Function hoisting can be get by proper function not by arrow function. Arrow funtion behave like a variable and in memory space “undefined” is stored , while in the case of proper function copy of whole code is stored into memory space.
  + **Undefined Vs not defined**
    - not defined means viable is not declared
    - undefined is and place holder , which assigned to variable at the time of variable memory allocation. It’s mean variable is declared but new value is not assigned.