JAVA SCRIPT

EP-7

->SCOPE CHAIN, SCOPE &LEXICAL ENVIORMENT

* SCOPE=LEXICAL ENVIORNMENT

**Code:**

function a()

{

console.log(b);

}

var b =10;

a();

**Console:**

10

* In function, b can access the b which is outside the function.

**Code:**

function a()

{

c();

function c(){

console.log(b);

}

}

var b =10;

a();

**Console:**

10

**Code:**

function a()

var b =10;

c();

function c{

}

console.log(b);

a();

**Console:**

Not Defined

* Scope: The place where we can access specific variable and function in the code.
* Lexical environment:
* Whenever execution context is created then a lexical environment is also created.
* Lexical environment is the local memory along with the lexical environment of its parent.(Lexical environment=local memory+ reference to the lexical environment of parent)

**Code:**

function a()

{

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | MEMORY | CODE | | |  | | --- | | Lexical Environment of its parent.(lexical parent of c is a because c is lexically inside a) | |  |   a()   |  |  | | --- | --- | | MEMORY | CODE | | b:10  c:{…}     |  | | --- | | Lexical Environment of its parent.(lexical parent of a is global) | |  |   GLOBAL EXECUTION CONTEXT   |  |  | | --- | --- | | MEMORY | CODE | | a:{..}   |  | | --- | | Lexical Environment of its parent.(lexical parent of gec is null) | |  | |

var b =10; call stack:

c();

function c(){

}

}

a();

console.log(b);

* Function is lexically inside a function.
* Where that specific code is present inside the code:a is lexically inside global scope
* c is lexically inside a.

**SCOPE CHAIN:**

The chain of lexical environment and parent references.

If JS engine does not find anything in the local memory it goes to the

next level of the scope chain.