

[whenever an execution context is created then a lexical environment is also created.]

Page No.

Date

The scope chain, scope and lexical Environment

episode
or

code

```
function a() {  
  console.log(b);  
}  
var b = 10;  
a();
```

⇒ 10

code

```
function a() {  
  c();  
  function c() {  
    console.log(b);  
  }  
}  
var b = 10;  
a();
```

⇒ 10

code

```
function a() {  
  var b = 10;  
  c();  
  function c() {  
    console.log(b);  
  }  
}  
a();
```

⇒ 10

lexical

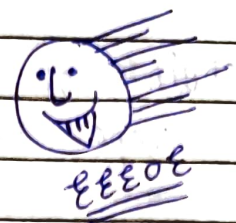
[lexical is a term means in hierarchy or in a sequence or in order]

ex. c() function is physically present in a() function
i.e. c() is lexically inside a() and
a() is lexically inside global scope

code

```

function a() {
  var b = 10;
  c();
  function c() {
  }
}
a();
console.log(b);
  
```



b is not defined.

scope - [scope means where you can access a specific variable or a function in our code.]

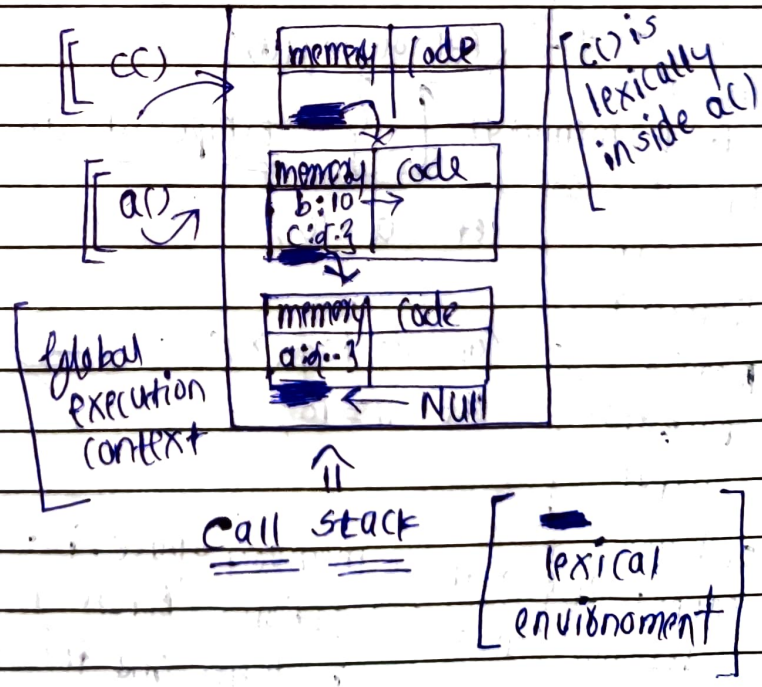
- [scope is directly dependent on lexical environment.]

*** - [lexical environment is the local memory along with lexical environment of its parent.]

code

```

example :-
function a() {
  var b = 10;
  c();
  function c() {
    console.log(b);
  }
}
  
```



10

[scope chain defines whether a variable or function is present inside these scope or not, if scope chain is exhausted and variable is not found, i.e. that means it is not inside the scope chain.]

scope chain

- [chain of lexical environment is called as scope chain.]