

The Scope Chain, Scope & Lexical Environment

Lexical Environment

It is created whenever Execution context is created.

Lexical Environment is the local memory + reference to lexical environment of parent.

The whole chain of lexical environment is known as scope chain and it defines whether a variable or function is present inside the scope or not.

Scope

It means where you can access a specific variable or function in a code.

Scope is directly depend on lexical environment.

eg: 1

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

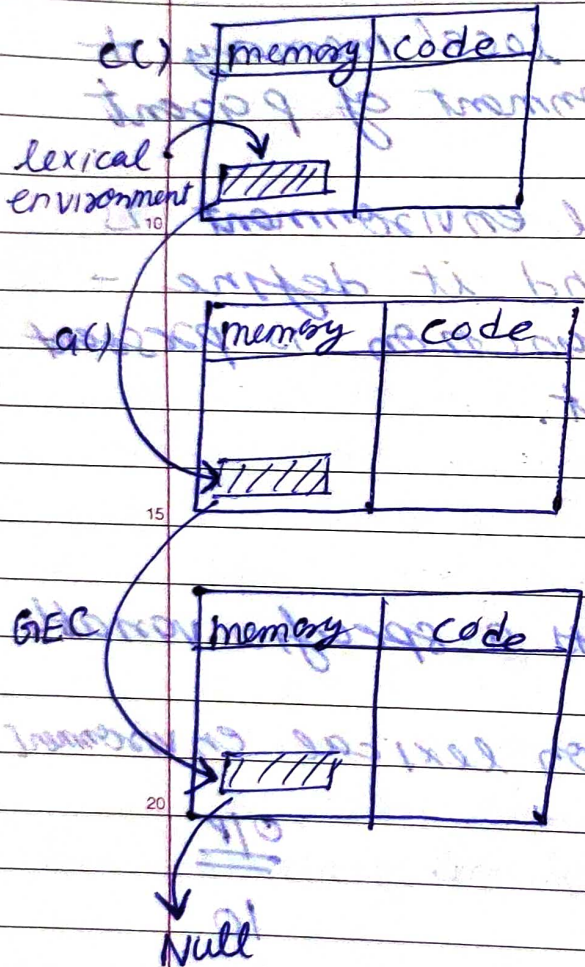
```
var b = 10;
a();
```

eg: 2

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

Will through uncaught Reference Error as scope of variable x is inside b() only.

* When local execution context is created then it holds the lexical environment of its parents along with variable and function of that Execution context.



- In code lexical environment of CC will point to AC EC and AC's lexical environment will point to GEC.
- GEC lexical environment points to null.

- Scope chain : chain of lexical environment reference.

JS engine first will search memory of that EC. If not found then through lexical environment will go parent of it and will perform search and if again not found null in GEC lexical environment. And finally JS engine will throw an error variable is not defined.