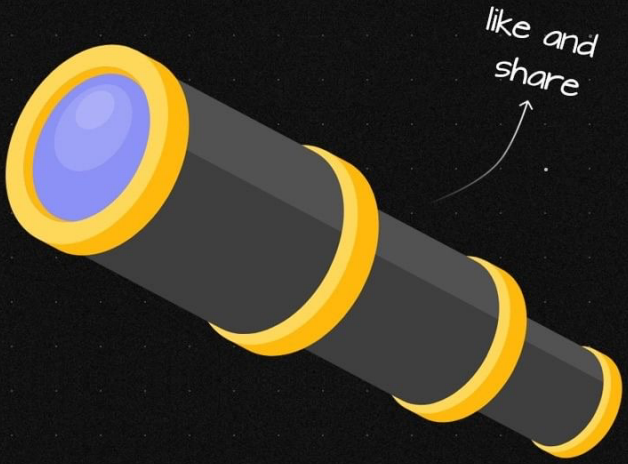
JS

Lexical scoping

in JavaScript



Code with Globa

Swipe →

JS

What is lexical scoping?

Lexical scoping is a convention used in many programming languages that sets the **scope of a variable** so that it may only be called (referenced) from within the block of code in which it is defined. During runtime, JavaScript does these two things: **parsing and execution.** Lexical scoping takes place during the **parsing phase of the js engine.**

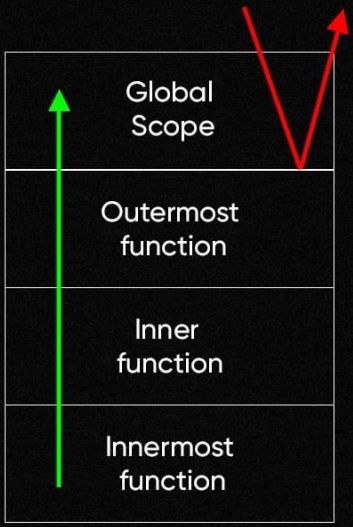
JS

The ability of a function scope to access variables from the parent scope is called Lexical scope. The **child function is lexically bonded to the parent** But the opposite is not true; the variables defined inside a functioncould not be **accessed outside that function.**

JS

Access from inner scope

Can access variables from inner scope of outer scope.



Access from outer scope

Can't access variables from outer scope of inner scope.

In the above diagram, we can see that, due to lexical scope, the functions may access all variables **from their parent scopes** up **to the global scope**, but no scope may access any variables from the functions defined inside it.

JS

In this example function `displayModel()` has access to `model` because of lexical scoping

Function `displayModel()` contains reference of function `car()`.

Variable `model` is part of the local scope and attached to function `diaplayModel()`.

```
function car(){
  var model = "Roadster";
  function displayModel(){
    console.log(model);
  };
  return displayModel;
};

var chooseCar = car();
chooseCar();
```

"Roadster"

output

Do you find it helpful?

let me know down in the comments !

Slobodan Gajić

Content Creator

Code with Sloba

FOLLOW FOR MORE