

# Agenda

→ Lexical Scope

→ Closure

	re-declare	re-initialise
var	✓	✓
let	✗	✓
const	✗	✗

var is functional scope and let/const is block scope

```
let a = 10;

function foo() {
  console.log("Inside:", a); // 10
}

foo();
```

Area A  
call stack

Destroyed



Area B  
Global memory

a = 10

foo = { fn }

global

## Lexical Scope

Lexical scope is the ability of a function to access variables from the parent scope. We say that the child function is "lexically bound" by the parent function.

```
function grandparent() {  
  let a = 20;  
  
  function parent() {  
    console.log("Parent: ", a); // 20  
  
    function child() {  
      console.log("Child: ", b); // 20  
    }  
  
    child();  
  }  
  
  parent();  
}  
  
grandparent();
```

scope chaining

Closures : A function that has access to its parent scope and can use its variables after the enclosing function is terminated