Closure

In JavaScript, a closure is created when a function accesses variables outside of its immediate lexical scope. The closure retains a reference to the environment in which it was created, allowing the function to access and manipulate variables in that environment, even after the outer function has returned.

function outer()

{

  let count = 0;

  return ()=>

  {

    count++;

    console.log(count);

  }

}

const counter = outer();

counter(); //1

counter(); //2

counter(); //3