

```
1  /*
2
3  A closure is created when a function is defined inside another function and gains access to the outer function's variables
  even after the outer function has finished executing.
4
5  */
6
7  // Clears the console to remove previous logs
8
9  console.clear();
10
11 // Function that creates and returns a counter function
12
13 function createCounter() {
14
15     // Private variable initialized to 0
16
17     let count = 0;
18
19     // Returns an inner function that has access to `count` via closure
20
21     return function () {
22
23         // Increments the private count
24
25         count++;
26
27         // Returns the updated count
28
29         return count;
30     };
31 }
32
33 // Create a new counter function instance
34
35 const counter = createCounter();
36
37 // Each call to counter() will increment and return the internal `count`
```

```
38  
39 console.log(counter()); // Output: 1  
40  
41 console.log(counter()); // Output: 2  
42  
43 console.log(counter()); // Output: 3
```