



Closure y Ámbito Lexical en Javascript

1. Cierre (Closure):

Un cierre es una función que tiene acceso a las variables de su ámbito externo, incluso después de que esa función haya terminado de ejecutarse. Esto permite crear entornos privados y funciones que conservan el contexto en el que fueron creadas. Ejemplo:

```
function exterior() {  
  
    let mensaje = 'Hola';  
  
    function interior() {  
  
        console.log(mensaje);  
  
    }  
  
    return interior;  
  
}  
  
const miClosure = exterior();  
  
miClosure(); // Imprime: Hola
```

En este ejemplo, interior es un cierre que tiene acceso a la variable mensaje de su ámbito externo exterior, incluso después de que exterior haya terminado de ejecutarse.

2. Ámbito Léxico:

El ámbito léxico se refiere a la capacidad de una función para acceder a las variables de su ámbito externo, basándose en el lugar donde se declara la función. Ejemplo:

```
function padre() {  
  
    let mensaje = 'Hola desde el padre';  
  
    function hijo() {  
  
        console.log(mensaje);  
  
    }  
  
    hijo();  
  
}
```

```
padre(); // Imprime: Hola desde el padre
```

En este ejemplo, la función hijo tiene acceso a la variable mensaje de su ámbito externo padre debido al ámbito léxico.

3. Cierre para Encapsulamiento:

```
function contador() {  
  
    let count = 0;  
  
    return function() {  
  
        count++;  
  
        console.log(count);  
  
    };  
  
}
```

```
const incrementar = contador();
```

```
incrementar(); // Imprime: 1
```

```
incrementar(); // Imprime: 2
```

En este caso, contador devuelve una función que actúa como un contador. La variable count está encapsulada dentro del cierre, lo que significa que no es accesible desde fuera. Esto proporciona una forma de mantener el estado privado.

Los cierres y el ámbito léxico son conceptos poderosos que permiten construir código más modular y seguro.