



## Curso de Closures y Scope en JavaScript



### Aprobaste el curso

Felicitaciones, ya puedes acceder a tu [diploma digital](#)

# 9.29

Calificación

# 13 / 14

Aciertos

1. El scope es el alcance de una variable en determinada parte de un programa

Verdadero



2. ¿Qué es Code Runner?

Es un plugin de Visual Studio Code que permite ejecutar el código dentro del editor



3. El uso de variables globales en nuestro código, ¿debería ser evitado?

Verdadero



4. ¿Qué representa el siguiente código?

```
const fruits = () => { var fruit = 'apple'; console.log(fruit); }
```

## Function Block Scope

## REPASAR CLASE

5. Analiza el siguiente código e identifica la variable declarada en el alcance de la función

```
const fruits = () => {  
  if (true) {  
    var fruit1 = 'apple';  
    const fruit2 = 'banana';  
    let fruit3 = 'kiwi';  
  }  
}
```

var fruit1 = 'apple';



6. Variables declaradas con "let" y "const" son de "block scope"

Verdadero



7. ¿Cuál de los siguientes códigos es un closure?

```
const buildCount = (i) => { let count = i; const displayCount = () => {  
  console.log(count++); }; return displayCount; }
```



8. JavaScript solo utiliza el hoisting en declaraciones, mas no en inicializaciones

Verdadero



9. ¿Qué sucederá si ejecutamos el siguiente código?

```
nameOfDog("Elmo");  
  
function nameOfDog(name)  
  
{ console.log(name); };
```

Se muestra en consola el siguiente mensaje: "Elmo"



10. Las variables declaradas con `let` y `const`, **NO** se pueden volver a declarar en el mismo ámbito.

Verdadero



11. ¿La herramienta de Google Chrome que nos permite hacer debugging se llama?

Chrome DevTools



12. ¿Cuál es la sección de Chrome DevTools donde podemos depurar el scope?

Sources



13. ¿Cuál de las siguientes opciones **NO** se puede considerar como una variable global?

```
var = "I am global";
```



14. Analiza el siguiente código:

```
const helloWorld = () => { globalVar = "I'm global"; } helloWorld();
```

¿Cómo se crea la variable `globalVar`?

globalVar se crea como una variable global



---

**REGRESAR**