



Curso de JavaScript Desde Cero

¡No te rindas!

Necesitas una **calificación mínima de 9.0** para aprobar.

Vuelve a intentarlo en 05 horas, 38 minutos, 19 segundos

6.88

Calificación

11 / 16

Aciertos

Resumen

1. ¿Cuál sería el resultado de la ejecución del siguiente código?

```
let fruta = { naranja: '🍊' }  
let vegetal = fruta  
vegetal.naranja = '🥬'  
console.log(fruta)
```

{naranja: '🥬'}



2. ¿Cuál sería el resultado de la ejecución del siguiente código?

```
const fruta = '🍏'  
fruta = '🍌'  
console.log(fruta)
```



[REPASAR CLASE](#)

3. ¿Los siguientes tipos de datos son complejos?

```
const medallas = [🏆, 🏅]  
console.log(typeof valorInicial)  
object console.log(typeof medallas)
```

Falso. null es primitivo.



4. ¿Qué operador se utiliza para concatenar dos strings en JavaScript?

join(' ')

[REPASAR CLASE](#)

5. ¿Qué valor de índice se obtiene con el siguiente código?

```
const evento = 'PlatziConf'  
console.log(evento.indexOf('c'))
```

-1



6. ¿Cómo podemos obtener la raíz cuadrada de numero ?

Math.sqrt(numero)



7. ¿Cuál sería el resultado de la ejecución del siguiente código?

```
const numeroEntero = 3  
console.log(!!numeroEntero)
```

TRUE



8. Son ejemplos de lenguajes de programación compilados

C, C++, Rust, Go, Swift



9. Al realizar las siguientes conversiones, ¿cuál es el tipo de dato que se imprime?

```
const numero = '596'  
const numeroConvertido = parseInt(numero)  
console.log(typeof numero)  
console.log(typeof numeroConvertido)
```

string number



10. ¿Cuántas expresiones hay en $(4*2) + (4*5)$

4

[REPASAR CLASE](#)

11. Es un nombre simbólico para un valor que se pasa a una función. Es parte de la declaración de la función.

Argumento

[REPASAR CLASE](#)

12. Es el valor real que se pasa a la función cuando esta se invoca. Es el valor que se asigna a un parámetro específico.

Parámetro

[REPASAR CLASE](#)

13. Son fragmentos de código que podemos utilizar una y otra vez en diferentes partes de nuestro programa.

Funciones



14. Es un ejemplo de una función pura

```
function sum(a, b) { return a + b }
```



15. ¿Cual de las siguientes muestra una arrow function?

```
const greeting = (name) => ( return `Hi, ${name}` )
```



16. ¿A qué hace referencia el principio DRY o Don't Repeat Yourself?

Sugiere que el código debe ser modular y reutilizable



[Ver menos](#)

Siguiente curso que podría interesarte

Asincronismo con JavaScript

Por Óscar Barajas



Curso de Asincronismo con JavaScript

Por: Oscar Barajas Tavares

Gestión de Paquetes y Dependencias en JavaScript

Por Óscar Barajas



Curso de NPM: Gestión de Paquetes y Dependencias en JavaScript

Por: Oscar Barajas Tavares

[Curso de Asincronismo con JavaScript →](#)

[Volver al Home](#)