

2 → ¿Qué caracteriza principalmente a "Syntactic Sugar" en JavaScript?*

- ☐ A Cambios en la funcionalidad del código.
- ☐ B Formas más complejas de escribir operaciones.
- ☒ C Formas más simples y legibles de escribir operaciones complejas. ✓
- ☐ D Una nueva versión de JavaScript.

Aceptar

3 → ¿Qué símbolo se utiliza para definir Template Strings en JavaScript?

*

- ☐ A Comillas simples (').
- ☐ B Comillas dobles (").
- ☒ C Backticks (`). ✓
- ☐ D Ninguno de los anteriores.

Aceptar

4 → ¿Cómo se puede acceder de manera segura a ``persona.direccion.calle`` usando sugar syntax para evitar errores si ``direccion`` no está definido?

*

- ☐ A ``persona?.direccion.calle``
- ☐ B ``persona.direccion?.calle``
- ☒ C ``persona?.direccion?.calle`` ✓
- ☐ D ``persona.direccion.calle?``

Aceptar

5 → ¿Cuál de los siguientes es un ejemplo de asignación con OR lógico (`||`) en JavaScript?

*

☐ A `const resultado = valor1 && valor2;`

☒ B `const mensaje = error || "Todo bien";` ✓

☐ C `const direccion = usuario?.direccion;`

☐ D `const nombre = persona.nombre ?? "Desconocido";`

Aceptar

6 → En el contexto de funciones de JavaScript, ¿cómo se declara un valor predeterminado para un parámetro?

*

☒ A `function miFuncion(param = "default") { ... }` ✓

☐ B `function miFuncion(param: "default") { ... }`

☐ C `function miFuncion(param || "default") { ... }`

☐ D `function miFuncion(param ?? "default") { ... }`

Aceptar

7 → ¿Cómo se implementa la desestructuración de objetos en JavaScript?

*

☐ A Usando paréntesis `()``

☐ B Usando corchetes `[]`

☒ C Usando llaves `{ }` ✓

☐ D Usando el signo `\$`

Aceptar

8 → ¿Para qué se utiliza el operador de propagación (spread operator) en objetos?

*

☐ A Para iterar objetos

☒ B Para combinar propiedades de objetos ✓

☐ C Para crear bucles infinitos

☐ D Para acceder a métodos privados

Aceptar

9 → ¿Qué hace el operador rest en objetos?

*

- ☐ A Repite el objeto varias veces
- ☐ B Crea una copia exacta del objeto
- ☒ C Recoge propiedades restantes en un nuevo objeto ✓
- ☐ D Expande el objeto en función de sus métodos

Aceptar

10 → ¿Cuál de los siguientes es un beneficio del "Property Shorthand" en la creación de objetos?

*

- ☐ A Incrementa el rendimiento del código
- ☐ B Hace que el código sea más largo y detallado
- ☒ C Hace el código más legible y corto ✓
- ☐ D Permite usar nombres de propiedades duplicadas

Aceptar

11 → ¿Qué característica describe mejor la desestructuración de arrays?

*

- ☐ A Permite acceder a propiedades de objetos rápidamente
- ☒ B Asigna elementos de un array a variables de forma concisa ✓
- ☐ C Es únicamente útil para arrays de objetos
- ☐ D Crea un nuevo array con los mismos elementos pero en orden inverso

Aceptar

12 → ¿Qué realiza una función en programación?

*

- ☐ A Cambiar el color de fondo de una página web
- ☒ B Ejecutar una tarea específica ✓
- ☐ C Crear nuevas páginas web automáticamente
- ☐ D Almacenar datos de usuario en una base de datos

Aceptar

13 → ¿Qué es una función flecha en JavaScript?

*

- ☐ A Una función que apunta a otra función
- ☐ B Una función definida con la palabra clave `arrow`
- ☒ C Una sintaxis corta para definir expresiones de función ✓
- ☐ D Una función que solo puede ser usada una vez

Aceptar

14 → ¿Qué es una función anónima en JavaScript?

*

- ☐ A Una función con un nombre único
- ☒ B Una función sin un nombre ✓
- ☐ C Una función que se llama a sí misma automáticamente
- ☐ D Una función definida dentro de otra función

Aceptar

15 → ¿Cuáles son los parámetros de una función?

*

- ☐ A Los valores que una función devuelve
- ☒ B Las variables especificadas entre los paréntesis en la declaración de la función ✓
- ☐ C Los errores que una función puede generar
- ☐ D Las funciones que se pueden llamar dentro de otra función

Aceptar

16 → ¿Cómo se puede importar un módulo JavaScript en otro archivo utilizando ES6?

*

- ☒ A ``import { modulo } from 'ruta del modulo';`` ✓
- ☐ B ``require('ruta del modulo');``
- ☐ C ``include 'ruta del modulo';``
- ☐ D ``module 'ruta del modulo';``

Aceptar

17 → ¿Cuál de los siguientes es un ejemplo de función de orden superior?

*

- ☐ A Una función que no acepta parámetros
- ☒ B Una función que acepta otra función como argumento ✓
- ☐ C Una función que solo devuelve números
- ☐ D Una función que no puede ser reutilizada

Aceptar

18 → ¿Qué es una callback en JavaScript?

*

- ☐ A Una variable que almacena una función
- ☐ B Un bucle especial para funciones
- ☒ C Una función pasada como argumento a otra función ✓
- ☐ D Un error generado durante la ejecución asincrónica

Aceptar

19 → ¿Qué método de array en JavaScript pasa cada elemento del array a una función y retorna un nuevo array con los elementos transformados?

*

- ☐ A forEach
- ☐ B filter
- ☒ C map ✓
- ☐ D reduce

Aceptar

20 → ¿Cuál de los siguientes NO es un método de array que acepta una callback en JavaScript?

*

☐ A sort

☒ B splice ✓

☐ C filter

☐ D reduce

Aceptar

21 → ¿Cuál es un ejemplo de uso del método reduce()?

*

☒ A Para sumar todos los elementos de un array. ✓

☐ B Para duplicar cada elemento de un array.

☐ C Para verificar si todos los elementos cumplen una condición.

☐ D Para encontrar el primer elemento de un array que cumpla con una condición.

Aceptar

22 → ¿Para qué se utiliza la función `fetch` en JavaScript?

*

☒ A Para realizar solicitudes de red y obtener recursos ✓

☐ B Para generar números aleatorios

☐ C Para crear una nueva promesa

☐ D Para almacenar datos en el navegador

Aceptar

23 → ¿Qué método se utiliza para consumir promesas y manejar tanto los valores resueltos como los errores?

*

☐ A `then()`

☐ B `catch()`

☐ C `finally()`

☒ D `then()` y `catch()` ✓

Aceptar

24 → ¿Qué es una promesa en JavaScript?

*

☐ A Un método para ejecutar código de manera síncrona.

☐ B Una función que se ejecuta automáticamente.

☒ C Un objeto que representa la eventual finalización o fracaso de una operación asíncrona. ✓

☐ D Un tipo de bucle de eventos.

Aceptar

25 → ¿Qué permite el asincronismo en JavaScript?

*

☐ A Ejecutar todas las operaciones de manera síncrona.

☒ B Iniciar una operación en un momento y finalizarla en el futuro, sin bloquear la ejecución de otras operaciones. ✓

☐ C Ejecutar solo una operación a la vez.

☐ D Ninguna de las anteriores.

Aceptar

26 → ¿Para qué se usa el método HTTP GET?

*

☐ A Enviar datos al servidor para crear un recurso.

☒ B Solicitar datos de un recurso específico. ✓

☐ C Actualizar un recurso existente.

☐ D Eliminar un recurso especificado.

Aceptar

27 → ¿Qué es TypeScript (TS)?

*

☒ A Un superset de JavaScript que añade tipado estático. ✓

☐ B Un lenguaje de programación completamente distinto a JavaScript.

☐ C Una versión antigua de JavaScript.

☐ D Un compilador de JavaScript para otros lenguajes.

Aceptar

28 → ¿Qué característica permite TypeScript para mejorar la detección de errores en el tiempo de desarrollo?

*

☐ A Ejecución dinámica

☐ B Tipado dinámico

☒ C Tipado estático ✓

☐ D Polimorfismo

Aceptar

29 → ¿Cómo se define una interfaz en TypeScript?

*

☒ A `interface Usuario { nombre: string; }` ✓

☐ B `type Usuario = { nombre: string; }`

☐ C `define Usuario { nombre: string; }`

☐ D `class Usuario { nombre: string; }`

Aceptar

30 → ¿Cuál es la sintaxis correcta para declarar un array de números en TypeScript?

*

☒ A `const numeros: number[] = [1, 2, 3];` ✓

☐ B `const numeros: array<number> = [1, 2, 3];`

☐ C `const numeros = [number: 1, 2, 3];`

☐ D `const numeros = array<number>[1, 2, 3];`

Aceptar

31 → ¿Cómo se puede definir un tipo que puede ser un número o una cadena de texto en TypeScript?

*

☐ A `type Mixto = number && string;`

☐ B `type Mixto = number || string;`

☒ C `type Mixto = number | string;` ✓

☐ D `type Mixto: number + string;`

Enviar