

Reporte de Evaluación - Fork de GitHub

Información General

Estudiante: David Quiroz
Repositorio: Strikys12/act_web1_s4
Fecha de evaluación: 24/8/2025, 23:37:11
Evaluado por: Sistema de Evaluación

Resumen de Calificaciones

Calificación general: 2.7/5.0
Actividades completadas: 13/20
Porcentaje de completitud: 65.0%

Detalle de Actividades

#	Descripción	Archivo	Encontrado	Calificación
1	Calcular el cuadrado de un número - Crea...	ejercicios/ejercicio_01.js	Sí	5.0
2	Saludar con nombre por defecto - Crea un...	ejercicios/ejercicio_02.js	Sí	5.0
3	Sumar elementos de un arreglo - Crea una...	ejercicios/ejercicio_03.js	Sí	3.0
4	Ejecutar un mensaje instantáneamente - C...	ejercicios/ejercicio_04.js	Sí	0.0
5	Contar vocales en una cadena - Crea una ...	ejercicios/ejercicio_05.js	Sí	1.0
6	Filtrar números mayores a 10 - Crea una ...	ejercicios/ejercicio_06.js	Sí	5.0
7	Convertir a mayúsculas - Crea una funció...	ejercicios/ejercicio_07.js	Sí	5.0
8	Temporizador con mensaje - Crea una func...	ejercicios/ejercicio_08.js	Sí	5.0
9	Crear un contador privado - Crea una fun...	ejercicios/ejercicio_09.js	Sí	5.0
10	Calcular factorial - Crea una función qu...	ejercicios/ejercicio_10.js	Sí	5.0
11	Verificar si un número es par - Crea una...	ejercicios/ejercicio_11.js	Sí	5.0
12	Multiplicar elementos de un arreglo - Cr...	ejercicios/ejercicio_12.js	Sí	4.0
13	Reemplazar espacios por guiones - Crea u...	ejercicios/ejercicio_13.js	Sí	4.0
14	Generar un ID único - Crea una función q...	ejercicios/ejercicio_14.js	Sí	2.0
15	Invertir una cadena - Crea una función q...	ejercicios/ejercicio_15.js	Sí	0.0
16	Sumar argumentos variables - Crea una fu...	ejercicios/ejercicio_16.js	Sí	0.0
17	Ejecutar operación personalizada - Crea ...	ejercicios/ejercicio_17.js	Sí	0.0
18	Validar correo electrónico - Crea una fu...	ejercicios/ejercicio_18.js	Sí	0.0
19	Retrasar ejecución de un mensaje - Crea ...	ejercicios/ejercicio_19.js	Sí	0.0
20	Calcular promedio de un arreglo - Crea u...	ejercicios/ejercicio_20.js	Sí	0.0

Retroalimentación Detallada

Actividad 1: Calcular el cuadrado de un número - Crea una función que reciba un número y devuelva su cuadrado. (Tipo de función: Declaración de función)

Archivo esperado: ejercicios/ejercicio_01.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

La solución es correcta y funcional. El código es legible y cumple con la descripción del problema. Buen trabajo.

Actividad 2: Saludar con nombre por defecto - Crea una función que salude a una persona por su nombre. Si no se proporciona un nombre, usa 'Invitado'. (Tipo de función: Expresión de función)

Archivo esperado: ejercicios/ejercicio_02.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

La solución es correcta y cumple con todos los requisitos. El código es claro y conciso, utilizando correctamente el valor por defecto en la función.

Actividad 3: Sumar elementos de un arreglo - Crea una función que sume todos los números de un arreglo. (Tipo de función: Función flecha)

Archivo esperado: ejercicios/ejercicio_03.js

Estado: Archivo encontrado

Calificación: 3.0/5.0

Retroalimentación:

La función solo funciona para arreglos de tamaño 3. Deberías usar un bucle o el método ``reduce`` para que funcione con arreglos de cualquier tamaño.

Actividad 4: Ejecutar un mensaje instantáneamente - Crea una función que imprima '¡Bienvenido!' en la consola al definirse. (Tipo de función: IIFE)

Archivo esperado: ejercicios/ejercicio_04.js

Estado: Archivo encontrado

Calificación: 0.0/5.0

Retroalimentación:

Archivo vacío - sin contenido para evaluar

Actividad 5: Contar vocales en una cadena - Crea una función que cuente las vocales (a, e, i, o, u) en una cadena. (Tipo de función: Función recursiva)

Archivo esperado: ejercicios/ejercicio_05.js

Estado: Archivo encontrado

Calificación: 1.0/5.0

Retroalimentación:

La función no implementa la recursividad y no cuenta las vocales correctamente en la cadena completa. Además, el condicional ``vocales.includes(texto)`` sólo verifica si la cadena ``texto`` completa es una vocal, no si contiene vocales.

Actividad 6: Filtrar números mayores a 10 - Crea una función que reciba un arreglo y devuelva solo los números mayores a 10. (Tipo de función: Función de orden superior)

Archivo esperado: ejercicios/ejercicio_06.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

La solución es correcta y concisa. Utiliza de manera eficiente la función ``filter`` para resolver el problema.

¡Excelente trabajo!

Actividad 7: Convertir a mayúsculas - Crea una función que convierta una cadena a mayúsculas. (Tipo de función: Función flecha)

Archivo esperado: ejercicios/ejercicio_07.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

La solución es correcta y utiliza una función flecha como se solicitó. El código es limpio y fácil de entender.

Actividad 8: Temporizador con mensaje - Crea una función que imprima un mensaje después de 3 segundos usando setTimeout. (Tipo de función: Función anónima)

Archivo esperado: ejercicios/ejercicio_08.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

La solución es correcta y concisa, cumple con los requisitos de la actividad utilizando una función anónima y setTimeout. El código es limpio y fácil de entender.

Actividad 9: Crear un contador privado - Crea una función que devuelva otra función para contar incrementos, manteniendo el contador privado. (Tipo de función: IIFE con closure)

Archivo esperado: ejercicios/ejercicio_09.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

Excelente solución. El código implementa correctamente un contador privado utilizando una IIFE y closure, cumpliendo con todos los requisitos de la actividad.

Actividad 10: Calcular factorial - Crea una función que calcule el factorial de un número. (Tipo de función: Función recursiva)

Archivo esperado: ejercicios/ejercicio_10.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

La función factorial implementada es correcta y recursiva. El código es limpio y conciso. ¡Excelente trabajo!

Actividad 11: Verificar si un número es par - Crea una función que determine si un número es par. (Tipo de función: Declaración de función)

Archivo esperado: ejercicios/ejercicio_11.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

La función esPar cumple con el objetivo y tiene un código limpio y fácil de entender. Podrías simplificar el if/else a ``return numero % 2 === 0;``

Actividad 12: Multiplicar elementos de un arreglo - Crea una función que multiplique todos los números de un arreglo. (Tipo de función: Función de orden superior)

Archivo esperado: ejercicios/ejercicio_12.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La función resuelve correctamente el problema. Sin embargo, podría implementarse usando el método ``reduce`` para cumplir mejor con el concepto de función de orden superior y mejorar la legibilidad. Considera agregar manejo de errores para casos donde el array contenga elementos no numéricos.

Actividad 13: Reemplazar espacios por guiones - Crea una función que reemplace los espacios en una cadena por guiones. (Tipo de función: Expresión de función)

Archivo esperado: ejercicios/ejercicio_13.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La solución es correcta y funcional. Se podría considerar usar ``string.replace()`` o ``string.replaceAll()`` para una solución más concisa y legible. Buen uso de una expresión de función.

Actividad 14: Generar un ID único - Crea una función que genere un ID único basado en un contador interno. (Tipo de función: IIFE con closure)

Archivo esperado: ejercicios/ejercicio_14.js

Estado: Archivo encontrado

Calificación: 2.0/5.0

Retroalimentación:

La función usa una IIFE con closure correctamente para mantener un contador. Sin embargo, genera un ID aleatorio en lugar de uno secuencial basado en el contador, lo cual no cumple con la descripción de la actividad. Se espera que retorne el valor del contador incrementado.

Actividad 15: Invertir una cadena - Crea una función que invierta una cadena de texto. (Tipo de función: Función recursiva)

Archivo esperado: ejercicios/ejercicio_15.js

Estado: Archivo encontrado

Calificación: 0.0/5.0

Retroalimentación:

Archivo vacío - sin contenido para evaluar

Actividad 16: Sumar argumentos variables - Crea una función que sume un número variable de argumentos. (Tipo de función: Función flecha con parámetro rest)

Archivo esperado: ejercicios/ejercicio_16.js

Estado: Archivo encontrado

Calificación: 0.0/5.0

Retroalimentación:

Archivo vacío - sin contenido para evaluar

Actividad 17: Ejecutar operación personalizada - Crea una función que reciba dos números y una función callback para realizar una operación. (Tipo de función: Función de orden superior)

Archivo esperado: ejercicios/ejercicio_17.js

Estado: Archivo encontrado

Calificación: 0.0/5.0

Retroalimentación:

Archivo vacío - sin contenido para evaluar

Actividad 18: Validar correo electrónico - Crea una función que valide si una cadena es un correo electrónico básico (contiene @ y .com). (Tipo de función: Declaración de función)

Archivo esperado: ejercicios/ejercicio_18.js

Estado: Archivo encontrado

Calificación: 0.0/5.0

Retroalimentación:

Archivo vacío - sin contenido para evaluar

Actividad 19: Retrasar ejecución de un mensaje - Crea una función que imprima un mensaje después de un tiempo definido por el usuario. (Tipo de función: Función anónima con setTimeout)

Archivo esperado: ejercicios/ejercicio_19.js

Estado: Archivo encontrado

Calificación: 0.0/5.0

Retroalimentación:

Archivo vacío - sin contenido para evaluar

Actividad 20: Calcular promedio de un arreglo - Crea una función que calcule el promedio de un arreglo de números. (Tipo de función: Función flecha)

Archivo esperado: ejercicios/ejercicio_20.js

Estado: Archivo encontrado

Calificación: 0.0/5.0

Retroalimentación:

Archivo vacío - sin contenido para evaluar

Resumen General

Necesita mejorar. Completó 13/20 actividades (65%) con una calificación promedio de 2.7/5. Se recomienda revisar los conceptos fundamentales.

Recomendaciones

- Revisar y mejorar las actividades con calificación baja
- Enfocarse en mejorar la documentación y comentarios del código
- Aplicar mejores prácticas de programación