

Reporte de Evaluación - Fork de GitHub

Información General

Estudiante: Diego Alejandro Arango
Repositorio: Diegoarango20/act_web1_s4
Fecha de evaluación: 21/8/2025, 16:11:27
Evaluado por: Sistema de Evaluación

Resumen de Calificaciones

Calificación general: 4.5/5.0
Actividades completadas: 20/20
Porcentaje de completitud: 100.0%

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	5.0
4	Ejecutar un mensaje instantáneamente - C...	ejercicios/ejercicio_04.js	Sí	5.0
5	Contar vocales en una cadena - Crea una ...	ejercicios/ejercicio_05.js	Sí	5.0
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	5.0
13	Reemplazar espacios por guiones - Crea u...	ejercicios/ejercicio_13.js	Sí	1.0
14	Generar un ID único - Crea una función q...	ejercicios/ejercicio_14.js	Sí	5.0
15	Invertir una cadena - Crea una función q...	ejercicios/ejercicio_15.js	Sí	5.0
16	Sumar argumentos variables - Crea una fu...	ejercicios/ejercicio_16.js	Sí	1.0
17	Ejecutar operación personalizada - Crea ...	ejercicios/ejercicio_17.js	Sí	5.0
18	Validar correo electrónico - Crea una fu...	ejercicios/ejercicio_18.js	Sí	4.0
19	Retrasar ejecución de un mensaje - Crea ...	ejercicios/ejercicio_19.js	Sí	5.0
20	Calcular promedio de un arreglo - Crea u...	ejercicios/ejercicio_20.js	Sí	5.0

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 función calcula correctamente el cuadrado de un número. El código es limpio y fácil de entender. 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:

¡Excelente solución! El código es conciso, funcional y utiliza el valor por defecto correctamente. Cumple con todos los requisitos del ejercicio.

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: 5.0/5.0

Retroalimentación:

Excelente solución. El código es conciso, utiliza la función `reduce` de manera eficiente y resuelve correctamente el problema planteado.

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: 5.0/5.0

Retroalimentación:

Excelente solución. Utiliza una IIFE correctamente para imprimir el mensaje al definirse la función. El código es conciso y cumple con el objetivo.

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: 5.0/5.0

Retroalimentación:

La solución es correcta y eficiente, implementando recursividad como se solicitó. El código es limpio y fácil de entender.

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, utilizando `filter` de manera eficiente. El código es legible y funcional.

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:

Excelente solución. El código es conciso, correcto y utiliza una función flecha como se solicitó. Cumple con todos los requisitos de la actividad.

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:

¡Excelente! La solución es concisa, correcta y cumple con los requisitos del ejercicio. El uso de `setTimeout` con una función anónima es adecuado.

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 usando una IIFE y closure. La solución es concisa y funcional.

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 solución es correcta y utiliza recursión de manera eficiente para calcular el factorial. El código es claro y conciso.

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 solución es correcta, concisa y eficiente. El código es limpio y fácil de entender, demostrando una buena aplicación de los conceptos.

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: 5.0/5.0

Retroalimentación:

La solución es correcta y eficiente al utilizar `reduce`. El código es limpio y fácil de entender.

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: 1.0/5.0

Retroalimentación:

El código proporcionado no resuelve el problema planteado (reemplazar espacios por guiones). En cambio, multiplica los elementos de un arreglo. Debes implementar la función que reemplace los espacios por guiones usando 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: 5.0/5.0

Retroalimentación:

Excelente solución. El código es claro, conciso y utiliza correctamente una IIFE con closure para mantener el estado del contador, cumpliendo con la descripción del problema.

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: 5.0/5.0

Retroalimentación:

La solución es correcta y concisa, utilizando recursión de manera eficiente para invertir la cadena. El código es legible y fácil de entender.

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: 1.0/5.0

Retroalimentación:

El código no cumple con la descripción de la actividad, que solicitaba una función flecha con parámetro rest para sumar un número variable de argumentos. En cambio, presenta una función 'operacion' que recibe un callback.

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: 5.0/5.0

Retroalimentación:

La solución es correcta y concisa. El código cumple con el objetivo de la actividad y presenta una buena estructura, demostrando un entendimiento claro de las funciones de orden superior.

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: 4.0/5.0

Retroalimentación:

La función cumple con los requisitos básicos, pero es una validación muy simple. Podrías mejorarla utilizando expresiones regulares para una validación más robusta de la sintaxis del correo electrónico.

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: 5.0/5.0

Retroalimentación:

La solución es correcta y concisa. Utiliza una función anónima dentro de `setTimeout` como se esperaba. El código es legible y funcional.

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: 5.0/5.0

Retroalimentación:

La solución es correcta, concisa y utiliza la función flecha como se solicitó. El código es limpio y fácil de entender, implementando buenas prácticas con el uso de `reduce`.

Resumen General

Excelente trabajo. Completó 20/20 actividades (100%) con una calificación promedio de 4.5/5. Demuestra buen dominio de los conceptos.

Recomendaciones

- Revisar y mejorar las actividades con calificación baja