

Reporte de Evaluación - Fork de GitHub

Información General

Estudiante: Brandon Ciro Ortiz
Repositorio: OBrandonC/act_web1_s3
Fecha de evaluación: 11/9/2025, 20:56:35
Evaluado por: Sistema de Evaluación

Resumen de Calificaciones

Calificación general: 4.0/5.0
Actividades completadas: 20/20
Porcentaje de completitud: 100.0%

Detalle de Actividades

#	Descripción	Archivo	Encontrado	Calificación
1	Crea un programa que use un ciclo while ...	ejercicios-while/ejercicio_while_01.js	Sí	1.0
2	Utiliza un ciclo while para calcular la ...	ejercicios-while/ejercicio_while_02.js	Sí	5.0
3	Implementa un sistema que use un ciclo w...	ejercicios-while/ejercicio_while_03.js	Sí	3.0
4	Crea un programa que use un ciclo while ...	ejercicios-while/ejercicio_while_04.js	Sí	3.0
5	Desarrolla una función que use un ciclo ...	ejercicios-while/ejercicio_while_05.js	Sí	5.0
6	Dado un array de nombres, usa un ciclo w...	ejercicios-while/ejercicio_while_06.js	Sí	4.0
7	Implementa un programa que use un ciclo ...	ejercicios-while/ejercicio_while_07.js	Sí	5.0
8	Utiliza un ciclo while para generar los ...	ejercicios-while/ejercicio_while_08.js	Sí	4.0
9	Crea un juego que use un ciclo while don...	ejercicios-while/ejercicio_while_09.js	Sí	5.0
10	Desarrolla un programa que use un ciclo ...	ejercicios-while/ejercicio_while_10.js	Sí	5.0
11	Usa un ciclo for anidado para crear las ...	ejercicios-for/ejercicio_for_01.js	Sí	5.0
12	Implementa un programa que use ciclos fo...	ejercicios-for/ejercicio_for_02.js	Sí	3.0
13	Dado un array de 20 números aleatorios, ...	ejercicios-for/ejercicio_for_03.js	Sí	5.0
14	Utiliza ciclos for anidados para encontr...	ejercicios-for/ejercicio_for_04.js	Sí	5.0
15	Crea un programa que use un ciclo for pa...	ejercicios-for/ejercicio_for_05.js	Sí	5.0
16	Implementa una función que use un ciclo ...	ejercicios-for/ejercicio_for_06.js	Sí	5.0
17	Usa ciclos for anidados para crear una m...	ejercicios-for/ejercicio_for_07.js	Sí	5.0
18	Desarrolla un programa que use un ciclo ...	ejercicios-for/ejercicio_for_08.js	Sí	2.0
19	Utiliza un ciclo for para generar la sig...	ejercicios-for/ejercicio_for_09.js	Sí	1.0
20	Implementa un programa que use un ciclo ...	ejercicios-for/ejercicio_for_10.js	Sí	5.0

Retroalimentación Detallada

Actividad 1: Crea un programa que use un ciclo while para mostrar una cuenta regresiva desde 20 hasta 1, mostrando cada número en la consola. Al final debe mostrar '¡Despegue!'.

Archivo esperado: ejercicios-while/ejercicio_while_01.js

Estado: Archivo encontrado

Calificación: 1.0/5.0

Retroalimentación:

El código no resuelve la actividad descrita. En lugar de una cuenta regresiva desde 20 hasta 1, implementa un ciclo que itera del 1 al 50 mostrando números divisibles por 3 y 5.

Actividad 2: Utiliza un ciclo while para calcular la suma de todos los números pares entre 1 y 50. Muestra el resultado final y cuántos números pares se sumaron.

Archivo esperado: ejercicios-while/ejercicio_while_02.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

La solución es correcta, clara y concisa. El código cumple con los requisitos del problema y utiliza buenas prácticas. Excelente trabajo.

Actividad 3: Implementa un sistema que use un ciclo while para solicitar una contraseña al usuario hasta que ingrese 'admin123'. Debe mostrar cuántos intentos fallidos hubo antes del acceso correcto.

Archivo esperado: ejercicios-while/ejercicio_while_03.js

Estado: Archivo encontrado

Calificación: 3.0/5.0

Retroalimentación:

El código no implementa la descripción original del ejercicio (ingresar 'admin123'). En cambio, valida la complejidad de la contraseña. Debería corregirse para cumplir con la descripción inicial.

Actividad 4: Crea un programa que use un ciclo while para generar números aleatorios entre 1 y 100 hasta que salga un número mayor a 95. Muestra cada número generado y al final indica cuántos números se generaron.

Archivo esperado: ejercicios-while/ejercicio_while_04.js

Estado: Archivo encontrado

Calificación: 3.0/5.0

Retroalimentación:

El programa no cumple con la condición de detenerse al encontrar un número mayor a 95. Además, el objetivo es encontrar un número mayor a 95, pero el código busca el número 50.

Actividad 5: Desarrolla una función que use un ciclo while para calcular el factorial de un número dado. Debe mostrar paso a paso cómo se va calculando el factorial.

Archivo esperado: ejercicios-while/ejercicio_while_05.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

Excelente solución. El código resuelve correctamente el problema del factorial mostrando los pasos intermedios. Bien hecho.

Actividad 6: Dado un array de nombres, usa un ciclo while para buscar un nombre específico. El programa debe mostrar en qué posición se encontró el nombre o indicar si no existe.

Archivo esperado: ejercicios-while/ejercicio_while_06.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

El código funciona correctamente y cumple con la descripción del problema. Sin embargo, falta la inicialización del array 'array'. Declarar `let array = [];` al inicio mejoraría la claridad y evitaría errores.

Actividad 7: Implementa un programa que use un ciclo while para contar cuántos dígitos tiene un número entero positivo. Por ejemplo, 12345 tiene 5 dígitos.

Archivo esperado: ejercicios-while/ejercicio_while_07.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

La solución es correcta y eficiente. El código es limpio y fácil de entender. Considera validar que la entrada sea un número positivo.

Actividad 8: Utiliza un ciclo while para generar los primeros 15 números de la secuencia de Fibonacci. Muestra cada número de la secuencia.

Archivo esperado: ejercicios-while/ejercicio_while_08.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

El código funciona correctamente y genera la secuencia de Fibonacci. La descripción del problema solicitaba 15 números, pero el código genera 20, lo cual es un pequeño error de correspondencia. Considera usar nombres de variables más descriptivos.

Actividad 9: Crea un juego que use un ciclo while donde el usuario debe adivinar un número entre 1 y 50. Proporciona pistas ('muy alto', 'muy bajo', 'cerca') y limita a 7 intentos máximo.

Archivo esperado: ejercicios-while/ejercicio_while_09.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

Excelente solución. El código es claro, funcional y cumple con todos los requisitos del problema. Bien implementado el manejo de intentos y las pistas para el usuario.

Actividad 10: Desarrolla un programa que use un ciclo while para procesar calificaciones de estudiantes. Debe continuar pidiendo calificaciones hasta que se ingrese -1, luego calcular y mostrar el promedio, la calificación más alta y más baja.

Archivo esperado: ejercicios-while/ejercicio_while_10.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

Excelente solución. El código es claro, bien estructurado y cumple con todos los requisitos. Se manejan correctamente las validaciones y los casos borde.

Actividad 11: Usa un ciclo for anidado para crear las tablas de multiplicar del 1 al 10. Cada tabla debe mostrarse claramente separada y formateada.

Archivo esperado: ejercicios-for/ejercicio_for_01.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

Excelente solución. El código es limpio, funcional y cumple con el objetivo del ejercicio, generando las tablas de multiplicar del 1 al 12 correctamente formateadas. ¡Bien hecho!

Actividad 12: Implementa un programa que use ciclos for anidados para crear un patrón de asteriscos en forma de diamante: * ****** ******* ******** ********* ********* ******** ******* ****** *****

Archivo esperado: ejercicios-for/ejercicio_for_02.js

Estado: Archivo encontrado

Calificación: 3.0/5.0

Retroalimentación:

El código genera una pirámide, no un diamante como se solicitaba. La lógica de la pirámide es correcta, pero debes agregar la parte inferior invertida para completar el diamante.

Actividad 13: Dado un array de 20 números aleatorios, usa un ciclo for para encontrar: el número mayor, el menor, la suma total, el promedio, y cuántos números son pares e impares.

Archivo esperado: ejercicios-for/ejercicio_for_03.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

Excelente solución. El código es claro, conciso y resuelve el problema planteado de manera eficiente. El uso de `toFixed(2)` para el promedio es un buen detalle para la presentación.

Actividad 14: Utiliza ciclos for anidados para encontrar y mostrar todos los números primos entre 1 y 100. Debe indicar cuántos números primos se encontraron.

Archivo esperado: ejercicios-for/ejercicio_for_04.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

Excelente solución. El código es claro, eficiente y cumple con todos los requisitos de la actividad. Bien hecho.

Actividad 15: Crea un programa que use un ciclo for para invertir una cadena de texto carácter por carácter. Muestra tanto la cadena original como la invertida.

Archivo esperado: ejercicios-for/ejercicio_for_05.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

La solución es correcta, clara y funcional. El código es legible y cumple con los requisitos de la actividad. Buen uso del ciclo for para invertir la cadena.

Actividad 16: Implementa una función que use un ciclo for para calcular la potencia de un número sin usar Math.pow(). Debe mostrar el proceso paso a paso.

Archivo esperado: ejercicios-for/ejercicio_for_06.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

Excelente solución. El código es claro, bien estructurado y maneja correctamente los casos especiales (exponente cero, exponente negativo, base cero). La salida paso a paso es muy útil.

Actividad 17: Usa ciclos for anidados para crear una matriz 5x5 donde cada elemento sea la suma de sus índices (fila + columna). Muestra la matriz formateada.

Archivo esperado: ejercicios-for/ejercicio_for_07.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

Excelente solución. El código es claro, conciso y cumple con todos los requisitos del problema. La matriz se construye y se imprime correctamente.

Actividad 18: Desarrolla un programa que use un ciclo for para contar cuántas vocales (a, e, i, o, u) hay en una frase dada. Debe mostrar el conteo de cada vocal por separado.

Archivo esperado: ejercicios-for/ejercicio_for_08.js

Estado: Archivo encontrado

Calificación: 2.0/5.0

Retroalimentación:

El código cuenta la frecuencia de todos los caracteres, no solo las vocales como lo pide el ejercicio. Además, la salida no es la esperada (conteo de cada vocal por separado). Debe enfocarse en filtrar solo las vocales y mostrar el conteo individual de cada una.

Actividad 19: Utiliza un ciclo for para generar la siguiente secuencia: 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024. Debe mostrar cada término y la suma total de la secuencia.

Archivo esperado: ejercicios-for/ejercicio_for_09.js

Estado: Archivo encontrado

Calificación: 1.0/5.0

Retroalimentación:

El código presentado no resuelve el problema planteado en la descripción de la actividad. En su lugar, simula el lanzamiento de dos dados y calcula la frecuencia de sus sumas.

Actividad 20: Implementa un programa que use un ciclo for para verificar si una palabra o frase es un palíndromo (se lee igual de izquierda a derecha que de derecha a izquierda). Ignora espacios y mayúsculas.

Archivo esperado: ejercicios-for/ejercicio_for_10.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

Excelente solución. El código es claro, conciso y funcional. La lógica para limpiar el texto y verificar el palíndromo es correcta y eficiente. Muy buen trabajo.

Resumen General

Excelente trabajo. Completó 20/20 actividades (100%) con una calificación promedio de 4.0/5. Demuestra buen dominio de los conceptos.

Recomendaciones

- Revisar y mejorar las actividades con calificación baja