20/06/2024.

* [Para obtener información adicional, puedes leer la documentación sobre JavaScript.](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Lexical_grammar)

Clase #4

Los instructores Christian y Leo están implementando un juego donde el usuario Los instructores Christian y Leo están implementando un juego donde el usuario debe adivinar un número secreto.

Ellos consideran que sería muy difícil para el usuario adivinar el número sin recibir pistas sobre si el número es mayor o menor.

Por lo tanto, deciden implementar esa funcionalidad de dar pistas al usuario.

Utilizan condicionales anidados para verificar si el número ingresado por el usuario es mayor o menor que el número secreto, y dar la pista correspondiente.

Prueban el juego con diferentes números y confirman que la funcionalidad de dar pistas está funcionando correctamente.

Sin embargo, se dan cuenta de que el juego solo permite una oportunidad de juego, y discuten cómo podrían hacer que el juego sea repetitivo, lo cual será abordado en el próximo video.

debe adivinar un número secreto.

Ellos consideran que sería muy difícil para el usuario adivinar el número sin recibir pistas sobre si el número es mayor o menor.

Por lo tanto, deciden implementar esa funcionalidad de dar pistas al usuario.

Utilizan condicionales anidados para verificar si el número ingresado por el usuario es mayor o menor que el número secreto, y dar la pista correspondiente.

Prueban el juego con diferentes números y confirman que la funcionalidad de dar pistas está funcionando correctamente.

Sin embargo, se dan cuenta de que el juego solo permite una oportunidad de juego, y discuten cómo podrían hacer que el juego sea repetitivo, lo cual será abordado en el próximo video.

Los instructores Christian y Leo están implementando un juego donde el usuario debe adivinar un número secreto.

Ellos consideran que sería muy difícil para el usuario adivinar el número sin recibir pistas sobre si el número es mayor o menor.

Por lo tanto, deciden implementar esa funcionalidad de dar pistas al usuario.

Utilizan condicionales anidados para verificar si el número ingresado por el usuario es mayor o menor que el número secreto, y dar la pista correspondiente.

Prueban el juego con diferentes números y confirman que la funcionalidad de dar pistas está funcionando correctamente.

Sin embargo, se dan cuenta de que el juego solo permite una oportunidad de juego, y discuten cómo podrían hacer que el juego sea repetitivo, lo cual será abordado en el próximo video.

Implementar pistas de "mayor" o "menor" para ayudar al usuario a adivinar el número secreto

Usar condicionales anidados para verificar si el número del usuario es mayor o menor que el número secreto

Probar el juego con diferentes números para confirmar que las pistas funcionan correctamente

Identificar que el juego solo permite una oportunidad de juego

Discutir cómo hacer que el juego sea repetitivo, lo cual se verá en el próximo video

Puntos importantes a anotar:

Uso de condicionales anidados

Implementación de pistas "mayor" y "menor"

Necesidad de hacer el juego repetitivo

Clase # 5.

Se introdujo el uso de bucles o loops en JavaScript para permitir que el juego del número secreto se repita hasta que el usuario acierte el número.

Se utilizó el bucle "while" para repetir el código mientras el número del usuario sea diferente al número secreto.

Es importante declarar las variables fuera del bucle "while" para que puedan ser accedidas dentro del mismo.

El operador "!=" (diferente) se usa para comparar si el número del usuario es diferente al número secreto, lo que mantiene el bucle activo.

La identación del código dentro del bucle "while" ayuda a visualizar la jerarquía y dependencia del código respecto al bucle.

Puntos clave:

Uso del bucle "while" para repetir el código hasta que se cumpla una condición.

Declaración de variables fuera del bucle para que puedan ser accedidas dentro.

Uso del operador "!=" para comparar si los números son diferentes.

Importancia de la identación para visualizar la estructura del código.

Clase #6.

Resumen:

En esta clase, los instructores Christian y Leo explican cómo implementar un bucle "while" en JavaScript para mejorar el juego del número secreto. Inicialmente, el juego solo permitía un intento, pero con el uso del bucle "while", el juego se vuelve más interactivo, permitiendo que el usuario siga intentando hasta que adivine el número correcto.

Puntos clave:

El bucle "while" en JavaScript permite repetir un bloque de código mientras se cumpla una determinada condición.

Es importante declarar las variables fuera del bucle "while" para evitar problemas de ámbito.

El operador "!==" (diferente) se utiliza para comparar si el número del usuario es diferente al número secreto.

La indentación del código dentro del bucle "while" ayuda a visualizar la jerarquía y la estructura del programa.

El uso del bucle "while" mejora la usabilidad del juego, permitiendo que el usuario tenga múltiples intentos hasta adivinar el número secreto.

Clase #7.

Puntos clave:

Creamos una variable "intentos" para llevar la cuenta de los intentos que hace el usuario.

Inicializamos la variable "intentos" en 1, ya que el primer intento se cuenta como uno.

Incrementamos el valor de "intentos" cada vez que el usuario no adivina el número secreto.

Utilizamos template strings para mostrar un mensaje al usuario indicando en cuántos intentos acertó el número.

Identificamos un problema con el mensaje cuando el usuario acierta en el primer intento, ya que decía "Realizaste la acción en 1 veces".

Para solucionar esto, creamos una variable "palabraVeces" que almacena la palabra "vez" o "veces" dependiendo de la cantidad de intentos.

Actualizamos el mensaje final para que use la variable "palabraVeces" y se muestre correctamente.

Resumen:

En esta lección, aprendimos a llevar un contador de intentos del usuario y a mostrar un mensaje final que indique en cuántos intentos acertó el número secreto. Además, solucionamos un problema de gramática en el mensaje cuando el usuario acertaba en el primer intento.