

1. Sobre el Desafío

Descripción

Amigo secreto

En este desafío, desarrollarás una aplicación que permita a los usuarios ingresar nombres de amigos en una lista para luego realizar un sorteo aleatorio y determinar quién es el **"amigo secreto"**.

El usuario deberá agregar nombres mediante un campo de texto y un botón **"Adicionar"**. Los nombres ingresados se mostrarán en una lista visible en la página, y al finalizar, un botón **"Sortear Amigo"** seleccionará uno de los nombres de forma aleatoria, mostrando el resultado en pantalla.

2. Funcionalidades:

- **Agregar nombres:** Los usuarios escribirán el nombre de un amigo en un campo de texto y lo agregarán a una lista visible al hacer clic en **"Adicionar"**.
- **Validar entrada:** Si el campo de texto está vacío, el programa mostrará una alerta pidiendo un nombre válido.
- **Visualizar la lista:** Los nombres ingresados aparecerán en una lista debajo del campo de entrada.
- **Sorteo aleatorio:** Al hacer clic en el botón **"Sortear Amigo"**, se seleccionará aleatoriamente un nombre de la lista y se mostrará en la página.

3. Descargar repositorio Base [HTML & CSS]

Descripción

El propósito de este desafío es poner en práctica los conceptos adquiridos sobre lógica de programación. Para ello, hemos proporcionado el código HTML y CSS, lo que permitirá que el enfoque se centre completamente en aspectos fundamentales sobre lógica de programación como variables, condicionales, funciones y listas, entre otros.

4. Crear repositorio en GitHub

Descripción

Git y GitHub son herramientas imprescindibles para todo desarrollador. Para este desafío, comienza creando un repositorio en tu GitHub y realiza commits a medida que vayas implementando cada funcionalidad.

Si aún no conoces Git y GitHub, esta es tu oportunidad para aprender, practicar y dominar estas herramientas fundamentales. Puedes consultar la documentación oficial para guiarte en el proceso:

4.1 Crear un array para almacenar los nombres

Descripción

Inicia declarando una variable de tipo array, que almacenará los nombres de los amigos ingresados. Ejemplo:

```
let amigos = []
```

Para saber mas sobre array puedes revisar la siguiente documentación:

4.2 Implementa una función para agregar amigos

Descripción

Implementa una función para agregar amigos

Desarrolla una función, que permita al usuario ingresar un nombre en el campo de texto y añadirlo a la lista de amigos creada anteriormente.

Tareas específicas:

- **Capturar el valor del campo de entrada:** Utilizar `document.getElementById` o `document.querySelector` para obtener el texto ingresado por el usuario.
- **Validar la entrada:** Implementar una validación para asegurarse de que el campo no esté vacío. Si está vacío, mostrar un alert con un mensaje de error: *"Por favor, inserte un nombre."*
- **Actualizar el array de amigos:** Si el valor es válido, añadirlo al arreglo que almacena los nombre de amigos usando el método `push()`.
- **Limpiar el campo de entrada:** Después de añadir el nombre, restablecer el campo de texto a una cadena vacía.

4.3 Implementa una función para actualizar la lista de amigos

Descripción

Crea una función que recorra el array amigos y agregue cada nombre como un elemento `` dentro de una lista HTML. Usa `innerHTML` para limpiar la lista antes de agregar nuevos elementos.

Tareas específicas:

Obtener el elemento de la lista: Utilizar `document.getElementById()` o

`document.querySelector()` para seleccionar la lista donde se mostrarán los amigos.

Limpiar la lista existente: Establecer `lista.innerHTML = ""` para asegurarse de que no haya duplicados al actualizar.

Iterar sobre el arreglo: Usa un bucle `for` para recorrer el arreglo amigos y crear elementos de lista (``) para cada título.

Agregar elementos a la lista: Para cada amigo, crear un nuevo elemento de lista.

4.4 Implementa una función para sortear los amigos

Descripción

Escribe una función que seleccione de manera aleatoria uno de los nombres almacenados en el array amigos. Usa `Math.random()` y `Math.floor()` para obtener un índice aleatorio.

Tareas específicas:

Validar que haya amigos disponibles: Antes de sortear, comprobar si el array amigos no está vacío.

Generar un índice aleatorio: Usar `Math.random()` y `Math.floor()` para seleccionar un índice aleatorio del arreglo.

Obtener el nombre sorteado: Utilizar el índice aleatorio para acceder al nombre correspondiente en el arreglo.

Mostrar el resultado: Actualizar el contenido del elemento de resultado utilizando `document.getElementById()` e `innerHTML` para mostrar el amigo sorteado.

5. Desafío extra

Descripción

El **README** es un elemento clave en cualquier proyecto de desarrollo, ya que proporciona una descripción clara y detallada del propósito, estructura y uso del código. Cuando participas en un proceso selectivo, el README es esencial para comunicar cómo utilizar el proyecto.

Este archivo, con extensión `.md` (Markdown), es el punto de referencia inicial para cualquier persona que quiera entender y trabajar con tu código.

Un buen **README** incluye información sobre la instalación, dependencias, cómo ejecutar el proyecto, y posibles problemas o soluciones. Un **README** bien estructurado facilita la comprensión del proyecto por parte de otros desarrolladores.

Aquí hay un artículo con los pasos para crear un README asombroso:

6. Publica tu proyecto

Publica tu página en GitHub Pages!

Utiliza lo que aprendiste sobre Git e GitHub y publica tu página web utilizando el servicio **GitHub Pages**. De esta forma, podrás compartir el resultado de tu proyecto!