

¿Qué es GIT?

Git es un sistema de control de versiones distribuido, diseñado por Linus Torvalds. Está optimizado para guardar cambios de forma incremental. Permite contar con un historial, regresar a una versión anterior y agrega funcionalidades.

GIT | ¿Cómo funciona?

Git almacena instantáneas de un mini sistema de archivos. Cada vez que confirmamos un cambio, Git toma una "foto" del aspecto del proyecto en ese momento y crea una referencia a esa instantánea. Si un archivo no cambió Git sólo crea un enlace a la imagen anterior idéntica que tiene almacenada.

Js es un lenguaje interpretado o compilado? Justifica tu respuesta

JavaScript es interpretado porque se ejecuta directamente sin compilación previa, aunque algunos motores usan técnicas de optimización durante la ejecución.

En el index.html vincula el archivo helpers.js antes que el game.js, debe llevar ese orden preciso? Justifica tu respuesta.

Sí, helpers.js debe cargarse antes que game.js si game.js depende de él, para evitar errores por funciones o variables no definidas.

¿Qué tipo de variable es "target"? Explica y ejemplifica

target es una variable que puede ser un objeto HTMLElement en eventos de DOM (representa el elemento que activó el evento) o cualquier otro tipo de objeto o destino en diferentes contextos de programación.

Identifica 3 funciones definidas por javascript y 3 funciones definidas por el usuario (identificar y definir las funciones flecha). Explica la utilidad de cada una.

alert()

- Definición: Muestra un cuadro de diálogo con un mensaje.
- Utilidad: Se utiliza para mostrar mensajes simples al usuario, útil para depuración y notificaciones rápidas.

- Ejemplo:

```
alert('Encontraste el tesoro en ${clicks} z
```

getRandomNumber

- **Definición:** Se asume que esta función genera un número aleatorio dentro de un rango dado.
- **Uso:** Se usa para generar las coordenadas x e y del objeto target.
- **Ejemplo de uso:**

```
let target = {  
    x: getRandomNumber(WIDTH),  
    y: getRandomNumber(HEIGHT)};
```

getDistance

- **Definición:** Calcula la distancia entre dos puntos. En el contexto del código, uno de los puntos es el clic del usuario (e) y el otro es el target.
- **Uso:** Se usa para determinar la distancia entre la ubicación del clic y el tesoro.
- **Ejemplo**

```
let distance = getDistance(e, target);
```

FUNCIÓN FLECHA EN EL CÓDIGO:

En el código proporcionado, no se utiliza una función flecha. Todas las funciones que aparecen son funciones importadas desde un módulo (getRandomNumber, getDistance, getDistanceHint) o una función anónima que se pasa como callback al addEventListener.

```
$map.addEventListener('click', function (e) {  
  
    console.log('click');  
  
    clicks++;  
  
    let distance = getDistance(e, target);  
  
    let distanceHint = getDistanceHint(distance);  
  
    $distance.innerHTML = `<h1>${distanceHint}</h1>`;
```

```
if (distance < 20 ) {  
    alert(`Encontraste el tesoro en  ${clicks} clicks!`);  
    location.reload();  
}  
});
```