

# JavaScript no invasivo, no obstructivo y no intrusivo

JavaScript no invasivo, no obstructivo o no intrusivo se refieren a un enfoque de programación en el cual el código JavaScript se integra en una página web de tal manera que no interfiera con el funcionamiento básico de la página ni cause problemas de rendimiento, accesibilidad u otros inconvenientes para los usuarios que visitan el sitio.

En términos simples, implica escribir código JavaScript de manera que no cause conflictos con la estructura y el comportamiento de la página web. Algunos principios clave de JavaScript no invasivo incluyen:

1. **Separación de contenido y comportamiento:** El código JavaScript debe estar separado del contenido HTML tanto como sea posible. Esto significa evitar la mezcla de código JavaScript directamente en las etiquetas HTML y, en su lugar, enlazar el comportamiento a través de atributos, eventos o clases específicas.
2. **Uso de eventos:** En lugar de incrustar funciones de JavaScript en las etiquetas HTML (por ejemplo, usando el atributo **onclick**), se recomienda utilizar métodos para adjuntar eventos a elementos HTML de forma externa. Esto ayuda a mantener el código más organizado y permite una gestión más sencilla de los eventos.
3. **No interferencia con la accesibilidad:** El JavaScript no debe bloquear o interferir con la navegación de los usuarios ni con la accesibilidad de la página. Esto significa que debe ser posible utilizar la página con funcionalidad reducida o deshabilitada JavaScript sin que afecte negativamente a la experiencia del usuario.
4. **Graceful degradation y progressive enhancement:** Al desarrollar con JavaScript no invasivo, se sigue el principio de "degradación elegante" (graceful degradation) y "mejora progresiva" (progressive enhancement). Esto significa que se crea una base funcional que funciona incluso en navegadores que no admiten todas las características avanzadas, y luego se agregan mejoras si el navegador y la compatibilidad lo permiten.

En resumen, JavaScript no invasivo busca integrar la interacción y la funcionalidad dinámica en una página web de manera que mejore la experiencia del usuario sin comprometer la usabilidad, la accesibilidad ni la integridad del contenido básico de la página.

# null y undefined

“null” y “undefined” son dos valores especiales en JavaScript que se utilizan para representar la ausencia de un valor o la falta de definición de una variable. Aunque ambos indican la ausencia de algo, tienen diferencias clave en su uso y comportamiento:

## undefined:

- **Significado:** Indica que una variable ha sido declarada pero aún no se le ha asignado un valor. También es el valor de retorno predeterminado para funciones que no tienen una declaración **return**.
- **Uso:** Se utiliza para verificar si una variable está declarada pero no inicializada.

## null:

- **Significado:** Es un valor que indica explícitamente la ausencia de valor o la intención de que una variable no tenga ningún valor.
- **Uso:** Se puede asignar a una variable para indicar que está vacía o no contiene ningún valor útil.

## Diferencias:

“undefined” se utiliza generalmente para variables que no se han inicializado, mientras que “null” se utiliza para indicar que una variable está vacía o no tiene valor.

Al comparar, “null” es un valor asignado explícitamente, mientras que “undefined” generalmente es asignado por JavaScript en ciertos contextos.

## Precauciones al acceder a datos con estos valores:

**“undefined”:** Antes de acceder a una variable, verifique si esta definida.

**“null”:** Al igual que con “undefined”, verifica si la variable no es “null” antes de acceder a ella.