1. Investigue a qué se le llama JavaScript no invasivo, no obstructivo o no intrusivo (unobtrusive JavaScript). Muestre un ejemplo.

JavaScript no obstructivo es la forma de escribir código JavaScript en el que separamos adecuadamente el contenido del documento y el contenido del script, lo que nos permite hacer una clara distinción entre ellos. Del mismo modo que deberíamos separar nuestra estructura y presentación colocando todos los CSS en un archivo separado y evitando el uso de atributos de estilo u otro marcado de presentación, también debemos separar nuestra estructura HTML y el comportamiento de JavaScript. Las razones son las mismas: separa tus preocupaciones, mantiene limpio tu código y te permite trabajar en JavaScript sin tocar HTML o CSS. Por lo tanto, básicamente se trata de separar el comportamiento o javascript de la presentación o html. Este enfoque es útil de muchas maneras, ya que hace que nuestro código sea menos propenso a errores, fácil de actualizar y depurar.

El concepto básico de la programación no obstructiva es que JavaScript debe usarse como una mejora de nuestra página web en lugar de un requisito absoluto. Si no necesita JavaScript, no lo use; su contenido estático se mostrará bien solo con HTML y CSS. Muchos desarrolladores cometen el error de importar bibliotecas de códigos incluso antes de determinar si serán necesarios. jQuery, por ejemplo, ha sido mal utilizado cuando todas nuestras necesidades básicas pueden satisfacerse fácilmente mediante CSS y código JavaScript puro. La idea de JavaScript no obstructivo es cambiar el diseño de la vieja escuela: en lugar de crear páginas web dinámicas 100% basadas en JavaScript, crear páginas web estáticas 100% regulares y luego, casi como una ocurrencia tardía, agregar una capa "también" de JavaScript.

Ejemplo:

```
<input type="button" id="btn" onclick="alert('Test')" />
```

Ese no es un JavaScript no obstructivo porque el comportamiento y la presentación son mixtos. El onclick no debería estar allí en html y debería ser parte de javascript en sí, no de html.

Con el ejemplo anterior, puede ser no obstructivo de esta manera:

JavaScript:

```
var el = document.getElementById('btn');
el.onclick = function(){
  alert('Test');
};
```

2. Investigue qué es Null y undefined en JavaScript

El tipo primitivo de JavaScript null representa una ausencia intencional de un valor; por lo general, se establece a propósito para indicar que una variable se ha declarado, pero aún no se le ha asignado ningún valor.

Esto contrasta null con el valor primitivo similar undefined, que es una ausencia involuntaria de cualquier valor de objeto.

Esto se debe a que una variable que se ha declarado, pero no se le ha asignado ningún valor es undefined, no null.

Desafortunadamente, typeof regresa "object" cuando se le solicita un valor null, debido a un error histórico en JavaScript que nunca se solucionará.

Eso significa que la comprobación de null no se puede realizar usando typeof.

Una forma de verificar null en JavaScript es verificar si un valor es aproximadamente igual al null usando el operador de doble igualdad: == .

Esto puede ser útil para verificar la ausencia de valor, null y undefined ambos indican una ausencia de valor, por lo tanto, son más o menos iguales (tienen el mismo valor a pesar de que son diferentes tipos).

Entonces, cuando se programa para verificar si una variable tiene algún valor antes de intentar procesarla, puede usar == null para verificar si hay null o undefined.

En cambio, si se desea constatar con exactitud un valor null, excluyendo cualquier undefined, se debe usar el operador de triple igualdad: ===.