Tarea 2

Buenas Practicas al escribir HTML

1. Ordenar etiquetas <div></div>

Es común ver cerrarse varias etiquetas div en la parte inferior de una página. Si bien los divs son más limpios que las tablas, dejarlos sin clasificar solo da como resultado un desorden que es difícil de limpiar. La sangría es una buena forma de organizar los cierres de etiquetas.

Sin embargo, la mejor manera de organizar sus etiquetas div es comentar cuál está cerrando. Por ejemplo, agregar un comentario como <! - # header -> ayuda al lector a comprender qué etiqueta está cerrando.

2. Utilice restablecimientos de CSS

Los restablecimientos de CSS son una práctica común en el desarrollo web, pero es sorprendente la cantidad de desarrolladores principiantes que se olvidan de hacer esto. El resultado son problemas de renderizado en varios navegadores que destruyen UX.

Un restablecimiento de CSS elimina el estilo individual de todos los elementos, y esto no deja espacio para que los navegadores anulen su elemento de estilo con sus valores predeterminados.

3. Evite la @importación

@import fue una opción popular para muchos desarrolladores cuando se trataba de importar archivos CSS. Incluir una hoja de estilo en otra es fácil con esta directiva. Alternativamente, los archivos CSS también se pueden incluir en documentos HTML.

Sin embargo, como los navegadores eliminan el soporte de importación predeterminado, los desarrolladores deben encontrar otras formas de incluir archivos HTML externos dentro del archivo principal. Cambiar el nombre de la extensión del archivo HTML externo a .shtml y usar inclusiones del lado del servidor (SSI) dentro del HTML es el mejor enfoque. Tenga en cuenta que el nombre del archivo con el comando SSI debe tener la extensión shtml.

4. Recuerde la accesibilidad

La web se está convirtiendo en un lugar mucho más inclusivo, y cada sitio web debe atender a personas con capacidades diferentes. En el pasado, los desarrolladores tenían que personalizar y reconfigurar cada elemento para hacer que sus sitios web fueran más accesibles. Un ejemplo clásico es agregar texto alternativo a todas las imágenes para que los lectores de pantalla comprendan cómo interpretar y explicar una foto a un usuario de lector de pantalla.

5. Comprimir y optimizar imágenes

La optimización de imágenes para la web es difícil incluso si utiliza los comandos "guardar para web" de Photoshop. Si no le preocupa el ancho de banda de su sitio, las imágenes no optimizadas no son un problema. Sin embargo, al crear sitios web comerciales, los desarrolladores casi siempre deben tener en cuenta el ancho de banda.

Herramientas como TinyJPG y compressjpeg son excelentes opciones gratuitas que los desarrolladores pueden usar. Smush it y RIOT son excelentes opciones de pago.

6. No mezcle CSS con HTML y Javascript

El propósito de HTML es organizar documentos web definiendo encabezados, pies de página, etc. Los desarrolladores de aplicaciones para el usuario a veces usan atributos HTML obsoletos para diseñar elementos. Sin embargo, esto a menudo se precipita en los desarrolladores que utilizan el atributo de estilo para insertar CSS directamente en HTML.

Esta práctica está bien en desarrollo o para una prueba rápida, pero difícilmente es adecuada para producción. Evite usar el atributo de estilo para insertar CSS, ya que esto va directamente en contra del espíritu de CSS. De manera similar, insertar Javascript en HTML es una mala práctica.

Usar Javascript sin obstrucciones puede parecer tedioso, pero da como resultado un HTML más limpio y mejores resultados de código.

7. Utilice HTML semántico

HTML es un lenguaje de marcado y se utiliza para crear documentos estructurados. Los desarrolladores deberían usarlo para indicar títulos, párrafos y otros elementos de diseño. Es importante utilizar HTML semántico para darle más significado a su página.

Las etiquetas semánticas hacen que su comunicación en la página sea clara para los navegadores y motores de búsqueda. Permiten una mayor personalización mediante el uso de CSS bien aplicado.

8. Construya y pruebe en paralelo

La renderización en varios navegadores es un gran problema que todo desarrollador web debe tener en cuenta. Pruebe su código mientras lo crea en lugar de renderizarlo en un navegador y ajustarlo para otros después. Ahorrará tiempo y le resultará más fácil depurar cualquier código.

Link para revisión de HTML:

<https://github.com/B4Bren/Tarea-2.git>