Buenas prácticas del desarrollo FrontEnd

En el desarrollo FrontEnd con el tiempo han ido surgiendo lo que se conoce como buenas prácticas, cosas que normalmente hacemos porque mejora el rendimiento o porque ayuda durante el desarrollo.

Con el tiempo desarrollando webs he ido aprendiendo estas buenas prácticas, algunas leyendo y otras descubriéndolas yo mismo mediante experimentación. A continuación explico varias de estas buenas prácticas que fui empezando a seguir.

Aviso: no es necesario seguir todas y cada una de estas buenas prácticas en todos los proyectos que uno realiza, algunas son útiles cuando se aplican en proyectos grandes y no lo son (o no tanto) en proyectos pequeños.

Comentá el HTML

Algo que la mayoría no hace es dejar comentarios en el HTML, esto es importante cuando se trabaja en equipo o cuando hay que mantener el mismo proyecto durante un tiempo para entender que es cada elemento.

Se vuelve más importante cuando tenés muchos <div> que existen solo por razones de diseño.

Minificá el HTML

La gran mayoría de sitios web no minifican el HTML que le envían el usuario, esto es muy importante ya que la minificación ayuda a que el HTML pese menos (permitiendo que el usuario lo descargue más rápido) y además elimina los comentarios que hayamos dejado.

Por último debido a que la minificación deja todo el código en una línea ayuda a evitar el espacio en blanco entre elemento al usar display: inline-block.

Cargá el CSS lo antes posible

Mientras más rápido el navegador descargue y lea el CSS más rápido va a terminar de renderizar la página. Hay que tratar de cargar los archivos CSS lo antes posible, lo ideal es que sea justo después de la etiqueta <title>.

Insertá en tu HTML el CSS crítico para tu aplicación

La carga de los archivos de CSS es una operación bloqueante, esto quiere decir que el navegador va a dejar de leer el HTML hasta que termine de descargar el archivo CSS.

Esto trae el problema de que si el archivo es muy pesado el navegador va a estar mucho tiempo sin hacer nada mientras espera que descargue. Para evitar este problema se puede detectar cuál es el CSS crítico para tu aplicación y agregar ese CSS en una etiqueta <style> directo en el HTML y el resto de estilos cargarlos de forma diferida usando JS al final del documento.

El CSS crítico básicamente es el CSS que se utiliza “above the fold”, osea que está en la parte visible del viewport al momento de entrar al sitio y sin haber hecho scroll down. Para poder

detectar cual parte de tu CSS es crítico hay herramientas como Penthouse  que tiene un plugin para Grunt  o se puede implementar con Gulp fácilmente.

Cargá el JS al final

La carga de archivos JS es una operación bloqueante por lo que detiene el renderizado de la página, si además de esto el JS se ejecuta inmediatamente al ser descargado puede dar errores si se trata de manipular el DOM antes de que se renderice el HTML.

Para evitar estos problemas es mejor siempre colocar las etiquetas <script> al final del HTML, justo antes de la etiqueta </body>. Esto permite que el usuario pueda interactuar con el resto del sitio mientras el JS se descarga y ejecuta.

Cargá cualquier JS no indispensable de forma asíncrona

La etiqueta <script> tiene un atributo llamado async que permite realizar la carga de una archivo de forma asíncrona. Para archivos como los de Google Analytics es mejor siempre cargarlos de esta forma para evitar que esta carga se convierta en una operación bloqueante.

Esto solo hay que hacerlo en archivos como Google Analytics que no son indispensables para el uso de nuestra aplicación web. Para los archivos de la aplicación web es mejor no usar async.

Elaborado a partir del artículo "Buenas prácticas del desarrollo FrontEnd" publicado en la web medium.com y que puedes encontrar completo en: <https://medium.com/@sergiodxa/buenas-pr%C3%A1cticas-del-desarrollo-frontend-40f44621841>