HTML

Para escribir mismas etiquetas varias veces, como en una lista, escribimos li\*3, por ejemplo, y se crearán tres etiquetas <li>.

Usamos h1 para señalar que se trata del título principal de la página. Hay que añadir también, que todos los h1 que marquemos tendrán el mismo tamaño y estilo por defecto, al igual si ponemos h2, etc. A menos que lo modifiquemos con CSS. Hace falta añadir que existen hasta h6 según nivel de importancia. Usamos h2 para señalar las secciones más destacadas del artículo.

La etiqueta <strong> se usa para destacar una palabra o frase de gran relevancia y con contenido semántico. La etiqueta <b>, sin embargo, se usa para remarcar una palabra o frase por razones estilísticas sin darle esa importancia semántica. Como el artículo trata sobre TikTok, y la palabra se menciona varias veces, he decidido utilizar la etiqueta <strong> para transmitir ese significado semántico especial de gran relevancia. También, en otros casos, se podría usar la etiqueta <mark class=”tiktok”> en HTML y en CSS darle estilo a esa clase.

Para los enlaces se utiliza la etiqueta <a href=”link”>con el texto o imagen que queramos que haga la función de click</a> y nos lleve al link proporcionado.

He puesto las etiquetas <h2> dentro de las etiquetas <article> ya que están relacionadas semánticamente e indican el título de ese artículo.

Para crear un enlace en estadísticas.html en la palabra Instagram, Facebook… y que lleve directamente a la palabra del artículo referente a Instagram, Facebook… del archivo redes.html, en estadísticas.html por ejemplo, en Instagram usé la etiqueta <a href=”./redes.html#Instagram”>Instagram,</a> donde #Instagram es el selector de id en redes.html de un <div>, aunque también pude haber usado <a name=”Instagram”>Instagram</a> para lograr el mismo efecto, teniendo en cuenta que, son sensibles a mayúsculas y minúsculas.

En relación a la organización de las listas anidadas del archivo estadísticas.html, para crear una lista desordenada he utilizado la etiqueta <ul> y para una lista ordenada he usado <ol>. Como había una lista dentro de otra he ido creando nuevas etiquetas <ul> para diferenciar una lista de otra, y dentro de cada <ul> he creado nuevos puntos <li>.

Para conseguir que se muestren los SVG correctamente en lugar de los puntos, he tenido que crear una clase para <li> y quitar los puntos con CSS, y crean un elemento <span> al que añadí con :before el SVG.

En el *footer* he decidido usar una lista ya que creo que hay una información organizada verticalmente, y que se podrían añadir más cosas creando un nuevo punto. Si el estilo del pie de página y la información fueran distintos, por ejemplo, todo en una línea pero separados a ambos lados de la página, probablemente habría usado la etiqueta <span >.

CSS

Teniendo en cuenta que trabajamos sin SASS para poder organizar mejor las carpetas de css según componentes, lo primero que he creado han sido reseteadores con margin: 0 y padding: 0, para evitar los márgenes exteriores e interiores que aparecen en el navegador y poder contar y trabajar con todo el tamaño de pantalla y organizarlo a mi forma, así como a:link, a:visited {text-decoration:none} y el color indicado para los enlaces de toda la página, para evitar que textos con enlace se pongan automáticamente de color azul o se subrayen. Siguiendo las reglas de especificidad de CSS y el funcionamiento en cascada de los estilos, he usado main a:link, main a:visited así como main a:hover para conseguir que los colores sean diferentes del header.

Para conseguir el estilo diferente de las e, les asigné una clase diferente con mayor especificidad, consiguiendo diferenciar las letras. Apliqué hover a las letras, pero no he conseguido que cambien de color junto al resto de la palabra cuando se pasa el ratón por el encima, sino que se activan por separado. Sospecho que el problema pueda estar en las etiquetas <span> que creé para cada “e”, pero no entiendo por qué no lo interpreta como un enlace a toda la palabra, ya que ambos <span> se encuentran dentro de la etiqueta <a>.

Siguiendo la metodología BEM, he ido creando clases como por ejemplo, nav\_\_menu y nav\_\_menu—element, etc. Que están relacionadas y se entiende que una está dentro de la otra, así como, he ido dando nombres descriptivos a las clases.

A la clase nav\_\_menu le di estilos con list-style: none evitando así que cualquier lista que creemos en nuestra página, nos vaya a mostrar puntos o números que no queramos, aplicándolo de este modo solamente a la lista que no me interesa.

He creado la clase nav\_\_menu—element y añadí márgenes para dar cierto espacio entre las palabras.

Referente a Font-size, he tenido en cuenta la herencia de CSS y apliqué Font-size:15px para todo el texto de <main> y después, para elementos específicos, le apliqué un tamaño de fuente diferente.

He usado selectores compuestos para dejar espacio entre cada párrafo en los artículos, para ello he escrito en CSS: article p{margin-bottom:20px;} consiguiendo que cada <p> dentro de una etiqueta <article> va a tener un espacio inferior de 20 píxeles.

Para personalizar los puntos de las listas, he tenido que crear una clase, y especificar el tipo de punto, así como el color.

Para conseguir las barras verticales he tenido que usar position relative en la caja padre y absolute en la hija, o poner un :before o :after.

En el <footer>, he tenido que usar selector compuesto para la lista desordenada que creé, ya que la porpiedad list-style:none no se puede aplicar directamente en <footer>.

En el footer he usado padding de 25px en una sola línea, salvo bottom que puse 15px, para compensar los 10px de padding-bottom que puse en footer li para dejar espacio entre ambos textos, consiguiendo de esta manera, un espacio uniforme en los bordes del pie de página en relación al texto.