

Actividad A2

Ana Alba Burgués

Uoc

1. Variables personalizadas:

- Se definen en el selector `:root`, lo que facilita la reutilización y la modificación de los colores en toda la página sin tener que cambiar cada declaración de color individualmente.
- Los colores son gestionados a través de variables como `--color1` (negro), `--color2` (blanco), `--color3` (verde brillante), entre otros.

2. Estilos generales:

- El selector universal `*` elimina márgenes y padding predeterminados, asegurando un diseño consistente.
- El `body` utiliza una fuente legible y colores de fondo y texto adaptados con las variables definidas.

3. Contenedores y estructura:

- `.container` es un contenedor centralizado con un ancho máximo de 1140px.
- Los elementos de la cabecera (`header`) y pie de página (`footer`) están diseñados de manera coherente, utilizando flexbox para una disposición adecuada.

4. Tipografía:

- Se emplean tamaños de fuente y espaciado consistente, con encabezados (`h1`, `h3`) configurados para ajustarse a la jerarquía visual.
- En la sección `hero`, se sobresale el título con un tamaño y peso específico.

5. Navegación:

- La barra de navegación usa flexbox para distribuir los elementos de forma eficiente.
- Se utiliza `justify-content: space-between` para separar los elementos y `align-items: center` para alinear verticalmente los ítems de la barra.
- Se incorporan efectos de hover en los enlaces para mejorar la experiencia del usuario.

6. Diseño de pie de página:

- El pie de página se organiza utilizando flexbox, tanto en su estructura general como en las divisiones de los logos y enlaces.
- Los enlaces del pie de página tienen un estilo similar a los de la navegación, con un efecto de hover que cambia el color del texto.

7. Sección principal y formularios:

- Secciones como `#segundo` tienen márgenes amplios, mientras que `.container-main` ajusta el contenido con márgenes laterales reducidos.
- Los formularios están organizados con flexbox, asegurando una disposición clara y bien alineada de los elementos.
- Los `checkboxes` se disponen en fila, facilitando su visualización y interacción.

8. Bloques de texto:

- Los bloques de texto en `.bloques-text` están bien distribuidos con flexbox, con márgenes y padding adecuados.
- Cada bloque tiene fondo blanco y bordes redondeados para un diseño limpio y moderno.

9. Media Queries:

- El diseño es adaptativo, utilizando media queries para ajustar el layout en pantallas pequeñas (menos de 768px de ancho). Esto asegura que el sitio sea completamente funcional y bien presentado en dispositivos móviles.
- Los elementos del pie de página y la navegación se reorganizan para mejorar la experiencia en dispositivos móviles.

10. Flexbox y Responsividad:

- El uso de flexbox es una característica clave para organizar la mayoría de los elementos, permitiendo un diseño fluido y flexible.
- En las pantallas más pequeñas, los elementos se reorganizan verticalmente o se adaptan a nuevas disposiciones para mantener la accesibilidad y la estética.

En el informe, se ha utilizado el servicio de publicación de GitHub para alojar y compartir el prototipo desarrollado. La plataforma ofrece una manera eficiente de gestionar el código fuente y facilitar la publicación del proyecto en línea. A continuación, se proporciona la URL que conduce a la primera página (index.html) del prototipo: [<https://github.com/aalba5/UOC-HTML-y-CSS>].