Ejercicios Navidad

Descarga el proyecto que tienes en aules. Junto a él hay un archivo llamado nieve.css, en él viene generada la compilación del archivo _nieve.scss a css, los valores que hay son valores calculados y por tanto no deberán ser utilizados como valores fijos. Si nos fijamos hay 50 divs en el html, eso quiere decir que hemos usado la directiva @for para pintarlos, no generéis 50 selectores.

El proyecto contendrá al menos los siguientes ficheros:

Fichero **style.scss**

- Realiza todas las importaciones, como es la fuente Mountains of Christmas de Google.
- Quita todos los márgenes por defectos que hay
- El **body** será un flex a modo columna, con una altura del 100%.
- Tanto el **header** como el **footer** usarán un mixin que se encargará de darle el valor de fondo(rojo) y del texto(blanco). El texto estará centrado y estará por encima del contenido del body. Nota: El mixin créalo en un archivo separado.

Fichero **scss/_menu.scss**

- El **header** será un flex donde si el tamaño de los elementos no caben en la pantalla este flex estará en modo reverse.
  - ✔ El **heading** que tiene deberá tener un margen, y oculto cuando la pantalla sea muy pequeña. A sus lados habrá uno de los iconos que tenemos en el directorio /public/icons. Recordad poner la ruta como si partiéramos desde public.
  - ✔ Para cuando nos entren todos los items en la pantalla en posición horizontal (como veis no digo en modo escritorio para que lo ajustéis vosotros ese media query en función de vuestras fuentes/márgenes/etc). Haz que el nav ocupe todo el ancho restante y que así:
    - La **lista** se muestre en modo flex, sin decorados. Para cada elemento de la lista aplícale un margen interno, altura, (oculta lo “sobrante”), y dale un tamaño mínimo para luego poder jugar con el ancho de los subítems. Al hacer hover sobre estos ítems, mostraremos el contenido, y en cada elemento de la lista que hagamos hover deberemos cambiar el color del texto para saber dónde estamos.
      - Los submenus se mostarán en modo columna, usa el interpolate-size para que la altura pueda ser auto a la hora de mostrarse (y el width).
  - ✔ Cuando estamos trabajando con una resolución más pequeña, da un ancho y alto al **nav**. Muestra el icono del menú.
    - La **lista** inicialmente estará desplazada a la izquierda tantos píxeles como anchura le hayas dado en el punto anterior. Calcula su altura.
      - Inicialmente los **subítems** no se mostrarán.
    - Muestra el listado al hacer hover sobre el icono, y asegúrate que si haces hover sobre los subitems estos se “despliegan” mostrando su contenido.

En el fichero **scss/_contenido.scss**

- El main estará en una posición relativa con un margen.
  - ✔ El section será un grid que se repite con un tamaño entre 420px y 1fr. Separa los hijos entre sí. Y dale una altura del 100%
  - ✔ Los articles tienen su borde, donde el header y el footer tienen su “decorado”
  - ✔ El figure de la imagen está a small-caps y la imagen tienen un width de 400px
  - ✔ El more-info está justificado y una sangría en la primera línea.

En el fichero **scss/_nieve.scss**

- Para realizar esto deberemos importar la librería matemática de sass. Ojo es importante tener en cuenta que debéis usar interpolate, para que pueda interpretar los valores. Ejemplo: `#{$i*2}n`
- Fijáos en las propiedades que se dan en el archivo nieve.css, de él podréis coger todos los valores menos los que aparecen en el transform del keyframe y en la clases .snow.
  - ✔ Para el keyframe el valor del translateX se realizará de acuerdo a la variable que se le pasará desde la invocación de cada uno de los .snow:nth-child(X). Ese valor se detalla más adelante.
  - ✔ Realiza un bucle que se repite 3 veces donde, para el hijo $iteracion*2n deberemos aplicar un blur que coincidirá con la iterárión (en píxeles), y con un tamaño de fuente que será un valor random multiplicado por 30px.
  - ✔ Realiza un bucle que se repita 50 veces (este será el bucle para los 50 copos que hay en el html), donde:
    - Para cada .snow:nth-child($iteracion), tendremos una animación donde:
      - deberemos calcular la posiciónXinicial (posición que lee el keyframe). Esta posición será un random(20)-10vw
      - deberemos calcular la posiciónXfinal (posición que lee el keyframe). Esta posición será un random(20)-10vw
      - deberemos calcular la posición que ocupará el copo a la izquierda, para ello será una posición random del 100vw
      - Invoca a la animación donde el valor de la duración será un valor random entre 0 y 15segundos y tendrá un delay entre 0 y 9 segundos.