

PEC 3 - Juan Estévez Sevilla

En esta última PEC3, como novedad a las anteriores PECs se introduce el uso de diversos métodos de diseño para estructurar el contenido del sitio web Vital360. A continuación, detallo los métodos principales empleados:

En primer lugar, la cabecera y el footer de la página, utilizan la propiedad float. Esta propiedad saca los elementos del flujo normal del documento y los pegan al lado izquierdo o derecho de su elemento contenedor padre según el valor seleccionado.

El resto del contenido por debajo del elemento flotado en el flujo normal, envuelve y rellena el espacio a la derecha o izquierda hasta la parte superior del elemento flotante.

Para evitar que este contenido se mueva hacia arriba, se debe limpiar el float. Esto se logra aplicando en el selector del elemento <main>, la propiedad clear con valor both, para eliminar cualquier elemento flotado, a la izquierda o a la derecha, .

La propiedad display con valor inline-block en los elementos de las listas no ordenadas de la cabecera y el pie de página cambia el comportamiento de los elementos (valor block) en el flujo normal. Esto provoca que estos elementos a nivel de bloque se comporten como elementos a nivel de línea.

La técnica de posicionamiento relativo y fijo también se utiliza en el sitio web.

El posicionamiento relativo desplaza el elemento de su posición predeterminada en el flujo normal. Esto se utiliza para mover los elementos con clase category y .author para que se alineen y se superpongan con la imagen. Las propiedades top y left se deben definir también con el valor asignado del desplazamiento.

El posicionamiento fijo se define en el elemento <header> con clase .header, para fijar el elemento en relación con la ventana gráfica del navegador. Así, la cabecera se quede en el mismo lugar de la pantalla mientras se desplaza el resto del contenido al hacer scroll.

Finalmente, se utiliza también como método de diseño completo principal: Flexbox.

Este método altera la forma en que se distribuyen los elementos secundarios dentro de los elementos padres. Esto facilita la disposición de los elementos en una dimensión, ya sea en fila o en columna.

El elemento padre que tiene establecido la propiedad display: flex se llama contenedor flexible y los elementos que se presentan como cajas flexibles dentro del contenedor flexible se denominan elementos flexibles.

La propiedad display con valor flex se utiliza en el elemento <div> de las páginas blog.html y videos.html. Esto produce que los hijos directos se convierten en elementos flexibles.

Se utiliza la propiedad flex con valor 1 en los elementos hijos para que todos los elementos crezcan y llenen el contenedor, en lugar de dejar espacio al final.

La propiedad `flex direction`, especifica la dirección en la que están dispuestos los elementos hijo del elemento `flexbox`. El valor por defecto, está establecido en el valor `row`, por lo que se presenta en una fila. Mientras que el valor en `flex-direction: column`, coloca los elementos en columna.

En la pagina `index.html`, los elementos flexibles en dirección inversa se utilizan con el valor `column-reverse`, para que comience desde la esquina opuesta de la ventana del navegador y fluyan al revés.

La propiedad `flex-wrap` con valor `wrap`, provoca que cuando se tiene definido un ancho determinado, los elementos flexibles salten de línea si el contenedor padre es inferior en tamaño para evitar el desbordamiento de los elementos.

Para especificar un valor de tamaño se utiliza la propiedad `flex-basis` con el valor del tamaño `390px`.

La propiedad `justify-content` con valor `center` alinea los elementos flexibles `<article>` de la página `index.html` sobre el centro del eje principal.

A continuación, paso a numerar otra serie de elementos, atributos y selectores ya utilizados en las otras pags.

El atributo `lang="es"` configura el idioma principal de la página. Esto permite que motores de búsqueda indexen el documento HTML de manera más efectiva, además de ser útil para las personas con discapacidad visual porque utilizan lectores de pantalla.

El elemento meta `charset = "utf-8"` establece que el documento HTML use la codificación UTF-8 y gestione la gran mayoría de caracteres de todos los idiomas escritos.

Los metaelementos con atributos `author` y `description` definen el autor y la descripción de la pagina. También incluyen palabras clave relacionadas con el contenido.

El elemento `<title>` establece el título de la página que aparece en la pestaña del navegador. El elemento `<link>` vincula la hoja de estilos externa, `estilos.css`, en los 4 archivos html.

En el elemento `<body>`, para implementar un marcado semántico en las secciones básicas del documento utiliza las etiquetas `<header>`, `<nav>`, `<main>` y `<footer>`.

El elemento `<header>` es el encabezado global de la página. Contiene como título principal un elemento `` con el logo Vital 360 a la izquierda y el menú de navegación a la derecha.

El elemento `<nav>` permite la navegación principal entre las páginas. Está formado por una lista no ordenada, elemento ``, que contiene dos elementos `` para representar los enlaces a las secciones del sitio: Blog y Videos.

El elemento `<main>` establece el contenido principal y el elemento `<footer>` sirve como sección final de la página, es decir, el pie de página.

El elemento `<article>` se utiliza para estructurar el contenido completo de cada página dividida en diferentes elementos `<section>`, para agrupar las diferentes partes del contenido relacionado y se establece también el elemento `<header>` como encabezado específico de cada sección.

El elemento `<h1>` representa el título principal en la página `index.html`, y los elementos `<h2>` `<h3>`, los subtítulos.

La etiqueta `` (elemento vacío, es decir, sin texto o etiqueta de cierre) requiere del atributo `src` con la ruta que apunta a la imagen que se quiere poner en la página. Para vincular semánticamente las imágenes y los `iframe`, página `videos.html`, con sus títulos de pie, se utilizan los elementos HTML5 `<figure>` y `figcaption>`.

El atributo `target` con valor `_blank`, se utiliza en los enlaces que no tienen relación directa con el sitio web, para que el navegador abra en una nueva ventana o pestaña la página vinculada en ese nuevo contexto, así el usuario puede seguir navegando en el sitio sin abandonarlo por completo.

La etiqueta `<form>` es el contenedor padre para crear el formulario. Este elemento contiene el atributo `action` para definir la ubicación (URL) donde se envían los datos recopilados del formulario, y el atributo `method`, para definir con qué método HTTP se envían los datos (método `get` o `post`).

El formulario contiene elementos tipo `<label>` y `<input>`.

La etiqueta `<label>` define una etiqueta para un widget de formulario HTML. Su atributo `for` toma como valor el atributo `id` del control de formulario al que está asociado. Esto permite a los usuarios hacer clic en la etiqueta para activar el control correspondiente, y también proporciona un nombre accesible para que los lectores de pantalla lo lean en voz alta.

La etiqueta `<input>` genera los widgets de formulario. Su atributo `type` genera los tipos de campos: texto (`type = "text"`) y casilla de verificación (`type="checkbox"`).

El atributo `name` en cada control o widget de formulario informa al navegador que nombre dar a cada dato y, en el lado del servidor, permite que el servidor maneje cada dato por su nombre. Los datos del formulario se envían al servidor como pares de nombre/valor.

Los botones de opción solo permiten una opción específica de las 3 opciones, y la casilla de verificación permite activar o desactivar la casilla anterior al elemento `<p>` (He leído y acepto los términos y condiciones).

El atributo `placeholder` genera el valor del texto indicado dentro del cuadro de entrada. Se utiliza para describir brevemente el propósito del cuadro.

Para envolver los elementos `<label>` y su widget se utilizan elementos `<p>`.

El elemento `<button>` con atributo `type` con valor `submit`, envía los datos una vez estas completados a la página web definida en el atributo `action` del elemento `<form>`. Este elemento permite usar HTML en su contenido, frente a los `input` de tipo `button`.

Las secciones de la página `index.html` presentan la clase `index` y contiene el elemento ancla con clase `category`, el elemento `<figure>` con la imagen, y un elemento no semántico `<div>` con clase `.blog` para separar el contenido textual en un bloque aparte con el elemento `<h3>` como subtítulo, el elemento `<p>` para el párrafo y otro elemento ancla con clase `read`.

Los artículos de la página `blog.html` presentan la clase `blog` y contienen el elemento ancla con clase `category`, el elemento `<figure>` con la imagen, y un elemento no semántico `<div>` con clase `.blog` para separar el contenido textual en un bloque aparte con el elemento `<h3>` como subtítulo, el elemento `<p>` para el párrafo y otro elemento ancla con clase `read`.

Las secciones de los videos de la página videos.html presentan la clase video y contienen un elemento <figure> para envolver el elemento <iframe> del video incrustado y un div que envuelve a su vez un elemento <h3> como titulo del video y un elemento p con una breve descripción.

Por otra parte en el CSS, una novedad destacable es la implementación de media queries, una técnica esencial para el diseño web responsivo.

Se define media queries para pantallas con un ancho de pantalla de 465px para móviles, 865px para tablet y 1920px para Desktop.

Esto aplica y cambia estilos específicos en los distintos elementos según las características para cada dispositivo.

Otra novedad utilizada con respecto a las otras pecs, es la propiedad transform con valor rotate (5deg), para girar las imágenes presentadas en la pagina de inicio index.html.

Asimismo, se utiliza una serie de propiedades ya utilizadas en las otras pecs como:

El modelo de cuadro alternativo en todo el documento, propiedad box-sizing: border-box y se configura todos los demás elementos para que hereden ese valor. Esto fija el ancho visible del contenedor aunque se establezca un padding o borde sobre el área del elemento.

Los tipo de letra Lato y Noto Serif se descargan del servicio de fuentes en linea de Google Fonts. El código HTML generado por la web una vez elegida la fuente y las familias 400, 600, 700 y 900, se sitúa dentro de las etiqueta <head> antes del elemento <link> para que se importe la fuente antes de usarla en el CSS.

Se establece inicialmente las propiedades line-height (altura de linea) y el color rgb(0, 59, 74) en el cuerpo del documento.

La propiedad background-image se utiliza sobre el elemento <header> con clase intro para visualizar la imagen como fondo. La propiedad background-repeat con valor no-repeat evita que la imagen de fondo se repita. La propiedad background-size ajusta el tamaño de la imagen para que quepa dentro del fondo. El valor cover permite que navegador agrande la imagen lo suficientemente para que esta cubra el área de la contenedor sin que pierda su relación de aspecto.

Aparte de la imagen de fondo, se establece también un color de fondo (background-color) que permita leer el texto por si la imagen de fondo no se carga.

La propiedad text-decoration en el selector a, elimina la linea inferior de los elementos ancla.

La clase .purple cambia el color de fondo a rgb (229,225,254) y la clase .grey cambia el color de fondo a rgb(247, 247, 247).

Varios selectores del documento utilizan las propiedades margin, border y padding para los márgenes, bordes y rellenos de los elementos.

La propiedad margin controla todos márgenes de un elemento y se utiliza para separar el elementos de los otros elementos.

Algunos elementos utilizan márgenes solamente en alguno de sus lados por medio de las propiedades: `margin-top`, `margin-right`, `margin-bottom` y `margin-left`.

La propiedad `border` dibuja una línea alrededor del elemento, en todos sus lados, entre el margen y el relleno.

La propiedad `border-radius` se utiliza para redondear las esquinas en varios elementos ancla.

La propiedad `padding` aleja el contenido del borde. Esta propiedad controla el relleno en todos los lados de un elemento. Para controlar cada lado individualmente, se usan las propiedades: `padding-top`, `padding-right`, `padding-bottom` y `padding-left`.

Se utiliza en el documento unidades en píxeles y `rem`, que es una unidad de longitud relativa. `rem` significa "el tamaño de fuente del elemento raíz".

La propiedad `font-weight` en varios selectores establece los diferentes pesos de la fuente.

Para aplicar el sentido de la dirección de los elementos flexibles en la página `index.html`, alternando sí se utiliza el selector `:nth-child` utilizado la palabra clave `odd`, para seleccionar los elementos hijos impares, y `even` para los pares.

La propiedad `text-align` establece la alineación horizontal del contenido de nivel en línea dentro de un elemento de bloque.

También, los elementos de formulario utilizan diferentes reglas de tamaño de cuadro para diferentes widgets, por lo que se utiliza la propiedad `box-sizing` para garantizar una experiencia consistente al establecer anchos y altos en elementos de formulario. Asimismo, también se establecen márgenes y relleno con valor 0 en todos los elementos.

La clase `active` cambia el color del elemento a `rgb(191, 12, 240)`.

Los elementos `li` de la lista no ordenada de la elemento `header` contiene la propiedad `border-left` para marcar un borde a la izquierda del elemento. Con el selector `.header ul li:first-child` el borde se elimina con el valor `none`.

La clase `logo` aplica un ancho de 100px en la imagen del logo del reader y el footer.

La pseudoclase `:hover` sobre varios elementos ancla, cambia el estilo de las propiedades `color` y `background` de estos elementos al pasar el cursor por encima.

La clase `read` definida en los elementos ancla de la página `blog.html`, contiene una pseudoclase `:after` para insertar el símbolo ">" con la propiedad `content` a continuación del texto contenido después de un elemento seleccionado.