

MEMORIA DOCUMENTAL

XXI JORNADAS DE PERIODISMO

HTML



INDICE

1. Introducción
2. Paso a paso
3. Explicación de las entidades HTML y CSS utilizadas
 - 3.1 Etiquetas utilizada en HTML
 - 3.1.1 Etiquetas comunes
 - 3.1.2 Etiquetas de index.html
 - 3.1.2 Etiquetas de programacion.html
 - 3.1.3 Etiquetas de inscripcion.html
 - 3.2 CSS utilizado, cuando, porque y para que?
 - 3.2.1 Selectores
 - 3.2.1 Declaraciones y propiedades
4. Cambios significativos
5. Verificación de HTML y CSS
 - 5.1 Archivo index.html
 - 5.2 Archivo programacion.html
 - 5.3 Archivo inscripcion.html
 - 5.4 Archivo estilos.css

1. Introducción

En la siguiente practica vamos a realizar una página informativa de unas jornadas de periodismo, repasando así los conceptos vistos en las unidades anteriores, como son los formularios, tablas, listas, listas anidadas, videos, imágenes y algunas mas.

Demostrando así:

- Conocer la estructura semántica de un documento HTML.
- Conocer el uso adecuado de las tablas en HTML y formularios.
- Conocer como crear estructuras de tabas de forma semántica.
- Conocer las etiquetas relacionadas con las tablas y formularios.
- Conocer como crear menús de navegación con listas.
- Conocer como utilizar CSS

He realizado en el entorno de desarrollo Visual Studio Code, ya que es el utilizado en el modulo.

2. Paso a paso

1. Dibujar un esquema de la distribución de la web que se propuso. Para ello he utilizado una captura de pantalla y el paint para dibujar la posición de los bloques.
2. Crear “index.html”.
3. Declarar el doctype y el head.
4. Crear una carpeta CSS donde se almacenara el archivo “estilos.css” y otra img, donde se almacenarán las imágenes necesarias.
5. Escribir el body con su estructura:
 - header
 - nav
 - main
 - footer
6. Crear los documentos “programacion.html” y “inscripcion.html”, aprovechando la misma estructura del index.
7. Estructurar respecto al esquema de distribución y escribir toda la información de cada pagina.
8. Ajustar los elementos con el archivo CSS y añadir en el HTML las clases necesarias para que no se solapen las instrucciones.
9. Reajustar los cambios propuestos por el profesorado en la practica, siendo la mayor parte en el CSS. También se ha sustituido la carpeta img por una llamada media que contiene ademas de las imagenes utilizadas un video que debe reproducirse en el index.
10. Por ultimo he pasado los archivos por el verificador para cerciorarme que no tienen errores.

3. Explicación de entidades utilizadas

3.1 Etiquetas utilizadas en HTML

3.1.1 Etiquetas comunes

`<!DOCTYPE>`

He utilizado esta etiqueta para informar al navegador qué versión se usó de HTML o XML para escribir el documento.

Lo he declarado con `<!DOCTYPE html>` en la línea 1.

Según la documentación, esta etiqueta no distingue entre mayúsculas y minúsculas por lo que `<!Doctype html>`, `<!DocType html>` y `<!doctype html>` también se considera correcto.

`<html></html>`

Define el documento entero de una página y el resto de los elementos que compondrá el sitio web.

`<head></head>`

Dentro de esta etiqueta he colocado los `<meta>`, `<link>`, `<title>` y `<script>` de la web que son informaciones que no se visualizan en la web pero contienen información necesaria para el navegador.

`<link>`

Esta etiqueta la he utilizado en el head para hacer un hipervínculo con “estilos.css”.

`<title></title>`

Dentro de esta etiqueta he puesto el título de la página que es el siguiente: XXI Congreso de periodismo del CEEDCV

`<script></script>`

Esta etiqueta se utiliza para llamar a un javascript, en este caso el necesario para el kit de iconos de fontawesome.com

`<body>`

Etiqueta que engloba la parte del documento que se va a visualizar en el navegador.

`<header>`

Utilizo esta etiqueta semántica que utilizo para englobar la cabecera. Dentro he usado un `` para la imagen y un `<div>` que contiene un `<h1>` para el título y un `<p>` que aporta lugar y fecha del evento.

`<nav>`

Etiqueta semántica que utilizo para englobar el menú de la web, constituido por una lista desordenada `` con tres ítems ``. Los cuales incluirá cada uno de ellos un hipervínculo con el atributo href ``.

`<main>`

Etiqueta semántica que he utilizado para englobar el contenido principal del body. Lo divido en un `<article>` y dos `<section>`.

`<footer>`

Etiqueta semántica que he utilizado para albergar el pie de página y contiene un párrafo `<p>`

3.1.1 Etiquetas de index.html

<article>

Etiqueta semántica en la cual he puesto un <h2> Titulo del articulo y un <div> que contiene un párrafo <p> y un video con los atributos src para decirle el origen del video, autoplay y muted para que se reproduzca automáticamente silenciado ya que si no está silenciado los navegadores no lo reproducirán de forma automática<video src="" autoplay muted>

<section>

Etiqueta semántica que he utilizado para las siguientes dos secciones.

La primera sección la he construido con una lista desordenada con 2 items .

El primer item es un h3 y una lista desordenada anidada que contienen 4 items. En esa lista anidada tenemos en los dos primeros item información sobre donde se realiza, en el tercer item he puesto el link al correo electrónico con un y en el cuarto item con un que no lleva a ningún sitio, he puesto los iconos importados con la etiqueta <i> que importa con el js que esta en el head con un<script>.

El segundo item es otro h3 y otra lista anidada de 5 items, de los cuales, los 4 primeros contienen información de cuando se realiza el evento y el 5 item contiene un a inscripcion.html.

La segunda sección contiene un <h2> con el titulo de la sección y seis <div> que contiene cada uno de ellos una imagen con atributos src(para el origen de la imagen y un texto alternativo) y un párrafo <p>. Esos 6 div estarán divididos en dos divs, por lo que dentro del primer div habrá 3 divs, y dentro del segundo otros 3.

3.1.2 Etiquetas de programacion.html

<section>

La primera sección del documento contiene un titulo <h2> y un <div> que contiene un párrafo<p> y una tabla<table>.

<table>

Esa tabla tiene 11 filas <tr> y 5 columnas y contiene información de la programación del evento.

La primera fila contiene 5 <th> que hace referencia a los encabezados de la tabla(uno por cada columna) como son fecha, taller, ponente, horario y ubicación.

La segunda fila contiene 5<td> que son datos de tabla (uno por columna). El primero de ellos tiene el atributo rowspan con valor 4 por lo que esa celda y las 3 inferiores serán combinadas para contener la información de los talleres del día 11 de marzo.

De la tercera a la quinta contienen 4<td> ya que la primera celda de la izquierda ya está creada desde la segunda a la quinta fila.

Se completa la tabla hasta la fila 10 utilizando el mismo método con la información del día 12 y 13 de marzo

En la fila 11 se coloca un colspan con valor 5 para combinar toda la celda para poner un hipervínculo a inscripcion.html.

La segunda sección del documento se ha estructurado con un <div> con un <h2> y tres listas ordenadas , anidadas a una lista desordenada .

La lista desordenada tiene 3 items que son los títulos <h3> de las listas ordenadas .

Cada lista ordenada , contiene la programación de cada día, por lo que cada una tiene tantos items como programación tenga ese día. (en este caso, 4).

3.1.3 Etiquetas de inscripcion.html

<section>

En este documento he decidido utilizar solo una sección que contiene un <h2> y un formulario que envía los datos por url gracias al método GET <form method="GET" action="">. En este caso el formulario contiene dos <fieldset> y un <div>.

<fieldset>

Un fieldset es un grupo de campos y en este caso tenemos dos y un tercero dentro de un <div> compartido con un botón <button>.

En el primer fieldset he colocado con <legend> la leyenda del fieldset, después un <div> que contiene dos inputs, de nombre, apellidos con el atributo de tipo texto, id, nombre y sus restricciones <input type="text" id="nombre" name="nombre" minlength="2" maxlength="50" size="25" required pattern="A-Za-z]+"> y sus correspondientes etiquetas <label> (se utiliza el div para que con CSS se puedan poner en la misma línea los dos inputs).

Fuera del <div> están los inputs tipo email, tel y file.

El segundo fieldset contiene igualmente de un <legend>, un área de texto con un alto de 5 líneas y un ancho de 100 caracteres que el cliente está obligado a escribir, dando los motivos de su interés en participar <textarea name="motivos" rows="5" cols="100" required>, un select con tamaño 1 para que el desplegable solo visualice una opción <select name="sector" id="sector" size="1"> con todas sus opciones siguiendo la misma estructura <option value="bachiller">, Estudiante bachillerato </option>, dos listas ordenadas con sus items para cada una de las conferencias con un input radial para confirmar asistencia que tiene chequeado el "no" <input type="radio" name="asistencia" id="dia11" checked> y que están anidadas sobre una lista desordenada con 3 items que contiene cada uno los días del evento como título <h4>, otra lista desordenada independiente con 3 items los cuales son radiales que dan a elegir el día de asistencia, y por último un <div> que contiene un tercer <fieldset> con un input tipo checkbox para aceptar las condiciones con su label correspondiente <input type="checkbox" name="legal" id="legal" required> <label for="legal"> Declaro haber leído y aceptar las condiciones generales del programa y la normativa sobre protección de datos. </label> y un botón tipo submit con el texto enviar <button type="submit"> Enviar </button>.

3.2 CSS utilizado, cuando, porque y para que?

3.2.1 Selectores

Genéricos:

*{} //apunta a todo por lo que lo que haya aquí dentro lo tomará como predeterminado.

body{} //apunta a todo lo que haya dentro del body.

h1{} //apunta al título de la página.

h2{} //apunta: dentro del index a el título del artículo y al título de la sección de eventos, dentro de programación a el título de la sección tabla y al título de la sección de la programación y dentro de inscripción al título de la sección de inscripción.

h3{} //apunta: dentro del index a título ¿Dónde se realiza? y a ¿Cuándo se realiza?, dentro de programación a los títulos que indican el día de la programación.

.negrita{} //clase creada para los textos y los títulos que van en negrita y apunta: dentro del index al ítem de la lista ¿Cuándo se realiza? que tiene un hipervínculo a inscripción, dentro de inscripción a los títulos de los días del programa y dentro de programación a las horas de la tabla.

.sub{} //clase creada para poner un subrayado punteado color amarillo que apunta a algunos hipervínculos de la web.

Header:

header{} //apunta a la cabecera de la página en los tres documentos. En el cual se ha puesto el tamaño de padding, el tipo de diseño

header img{} //apunta a todas las imágenes que hayan dentro del header, en este caso solo hay una, que sería el logo.

header div{} //apunta a todos los divs que hay en la cabecera, que en este caso solo es uno.

header p{} //apunta a todos los párrafos que hay dentro de la cabecera. En este caso solo hay uno situado debajo de la cabecera.

Nav:

nav{} //apunta a la barra de navegador de la web

nav ul{} //apunta a todas las listas desordenadas que hay en el nav. En este caso solo hay una.

nav li{} //apunta a todos los ítems de listas que hay en el nav. En este caso hay 3 ítems.

nav a{} //apunta a todos los hipervínculos que hay dentro del nav. En este caso 3, uno por cada ítem de lista.

nav a:hover{} //apunta a los hipervínculos que tenga el puntero encima y que estén dentro del nav.

.a_select //Primera clase que nos encontramos en la web y en este caso apunta a el hipervínculo de la página en la que nos encontramos.

Article:

article{} //apunta a todos los artículos de la web. En este caso solo hay uno

article h2{} //apunta a todos los titulos 2 del articulo. En este caso solo hay uno

article div{} //apunta a todos los divs que contiene el articulo. En este caso solo hay uno.

article p{} //apunta a todos los párrafos que contiene el articulo. En este caso solo hay uno.

article video{} //apunta a todos los videos que contenga el articulo. En este caso solo hay uno.

Sections:

Al existir 5 sections he tenido que crear una clase para cada uno de ellos:

.donde_y_cuando{} //apunta a la sección donde se explica el donde y cuando se hacen las jornadas que existe en el index.

.eventos{} //apunta a la sección donde se encuentran los eventos interesantes que están en el index.

.tabla{} //apunta a la sección donde se encuentra la tabla de ponencias que hay en la pagina programación.

.programa{} //apunta a la sección donde se encuentra el programa de los tres días en la pagina programación.

.inscripcion{} //apunta a la sección donde se encuentra el formulario de inscripcion.

Para no hacer mas clases dentro de cada una de esas clases se ha decidido trabajar con el modelo de hijos ">"

Footer:

footer{} //apunta a todo el pie de pagina

footer > p{} //apunta a todos los párrafos que sean hijos de footer

3.2.2 Declaraciones y propiedades

@import url(""); //Utilizado para importar en este caso los de letra a nuestro archivo CSS

box-sizing: border-box; //Utilizado para que el navegador tome en cuenta los bordes como tamaño máximo incluido el grosor del borde y el relleno que se añada.

display //Utilizado para maquetar la web, en este caso lo he utilizado con el valor "flex" que permite usar el modelo de cajas flexibles. Se coloca esta declaración a los elementos padre para que los elementos hijo se comporten de la manera especificada con otras declaraciones como son las dos siguiente. También lo he utilizado con el valor inline para que los items de algunas listas en concreto se alineen en horizontal.

justify-content //Utilizado para especificar el comportamiento que deben adoptar los elementos hijo que se alinearan el vertical u horizontal, como puede ser los valores center, flex-start o flex-end.

align-items //Utilizado para justificar los elementos hijo en vertical.

padding //Utilizado para dar relleno. Lo he utilizado en el "*" para que sea 0 como predeterminado ya que algunas etiquetas pueden generar rellenos no deseados. También se ha asignado esa declaración a varios selectores que se ha necesitado poner relleno. Algunos como padding, y en otros solo dándole el relleno de la

dirección en la que se toma como seria: padding-top(superior), padding-left(izquierda), padding-right(derecha) o padding-bottom(inferior).

margin //Utilizado para poner un margen. Lo he utilizado en el “*” para que sea 0 como predeterminado ya que algunas etiquetas pueden generar márgenes no deseados, Al igual que el padding lo he utilizado en varios selectores y también se pueden asignar por top, left, right o bottom.

border //Utilizado para poner bordes a los elementos necesarios que en este caso son los que tienen el borde inferior punteado de color amarillo. Se pueden asignar de forma general o con -top, -left, -right y bottom(en este caso border-bottom).

width //Utilizado para darle un ancho, y en este caso lo he utilizado en el body con el valor 100% para que utilice el 100% de la ventana del navegador. En otros selectores les he puesto los píxeles del ancho que creía oportunos para cada selector.

height //Utilizado para darle un alto, en este caso lo he utilizado para darle un alto mayor al del contenido en algunos de los elementos como por ejemplo el selector footer.

background-color //Utilizado para poner un color de fondo al elemento

font-family //Utilizado para poner los tipos de letra de la web. Lo he utilizado en el “body” para que lo herede toda la web y después sustituirla en los selectores concretos que tengan otro tipo de letra por especificidad.

font-weight //Utilizado para darle un ancho a la fuente, en este caso lo he utilizado para poner algunos textos o títulos en negrita con el valor “bold”.

font-size //Utilizado para dar tamaño a la fuente del selector al que está asociado.

text-decoration //Utilizado para decorar los textos, ya sea una línea inferior, tachados o una línea superior con sus diferentes estilos. En esta web lo he utilizado para quitar el subrayado de los hipervínculos con el valor “none” y para ponerlo al hipervínculo de el documento en el que estemos dentro de la web con el valor underline.

line-height //Utilizado para darle un ancho de línea mayor al predeterminado, en este caso lo he usado en el footer para separar una línea de otra algo más que al resto de párrafos.

color //Utilizado para darle color a los textos y titulares que hay en los selectores que tienen dicha declaración.

transform: rotate(-20deg); //En este caso he usado esta declaración para rotar una imagen 2D en su eje central. La he utilizado en la imagen del header.

list-style //Se utiliza para darle estilo a las listas. En este caso lo he utilizado para quitar el estilo de las listas con el valor “none”, ya que no me interesaban.

5. Verificación de HTML y CSS

5.1 Archivo index.html

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 11 milliseconds.

5.2 Archivo programacion.html

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 12 milliseconds.

5.3 Archivo inscripcion.html

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 14 milliseconds.

5.4 Archivo estilos.css

¡Enhorabuena! No error encontrado.

¡Este documento es [CSS versión 3 + SVG](#) válido!