

Buenas prácticas con HTML 5

<!DOCTYPE html> 😊

<html> 😞 **Ahora lo correcto es poner la etiqueta “lang” para describir el lenguaje de nuestra página, quedando así <html lang=”es”> la sigla “es” corresponde al idioma español.**

<head> 😊

😞 **Es necesario colocar aquí la etiqueta “meta” con el tipo de codificación para el lenguaje correspondiente, en este caso es así <meta charset=”UTF-8”>**

<title> MI SEGUNDA APLICACION </title> 😊

<link rel='stylesheet' href='ejercicio2.css' /> 😊

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" /> 😞

Esta línea ya no va y se sustituye arriba como te declaré bien la etiqueta “meta”.

NOTA: Cuando declaramos los linkeos y los scripts del “head” es conveniente poner primero los elementos CSS, comenzando por el “normalize.css” que es el que nos normaliza que todos los elementos tengan la misma proporción de tamaño para los navegadores, posteriormente poner los CSS que le dan estilo a nuestro sitio y por ultimo será agregar los scripts de JavaScript o de jQuery que utilizará nuestra web app según sea el caso.

</head> 😊

<body> 😊

<div>

<header id="logo">

</header>

</div>

<!--inicio del menu -->

<div id="menu">

<nav>

<ul class="men" id="nav">

<li class="seleccionar">Inicio

Tutoriales

Recursos

Contacto

</nav>

</div>

- **No se recomienda utilizar un “div” que contenga la etiqueta “header” si el navegador por si solo ya interpreta esa etiqueta como el elemento que irá en la parte superior de la página.**
- **Como buenas prácticas las etiquetas que denoten elementos de tipo “img” deberán ser incluidas dentro de etiquetas “figure” Ej.**

<figure>

</figure>

- Recordar nuestro “Mockup”(Imagen llamada “Principal.jpg”) y visualizar que el menú esta dentro de nuestro <<header>>, por lo tanto la etiqueta “nav” deberá estar contenida en la etiqueta “header” y no afuera con otro “div”, ya que nos será más fácil manipular el menú por medio del CSS.
- No es practico nombrar un “id” que tenga el mismo nombre semántico que una etiqueta, como lo haces con el <<id=“nav”>> y la etiqueta “nav” mismas que puse en color rojo en el código de arriba.

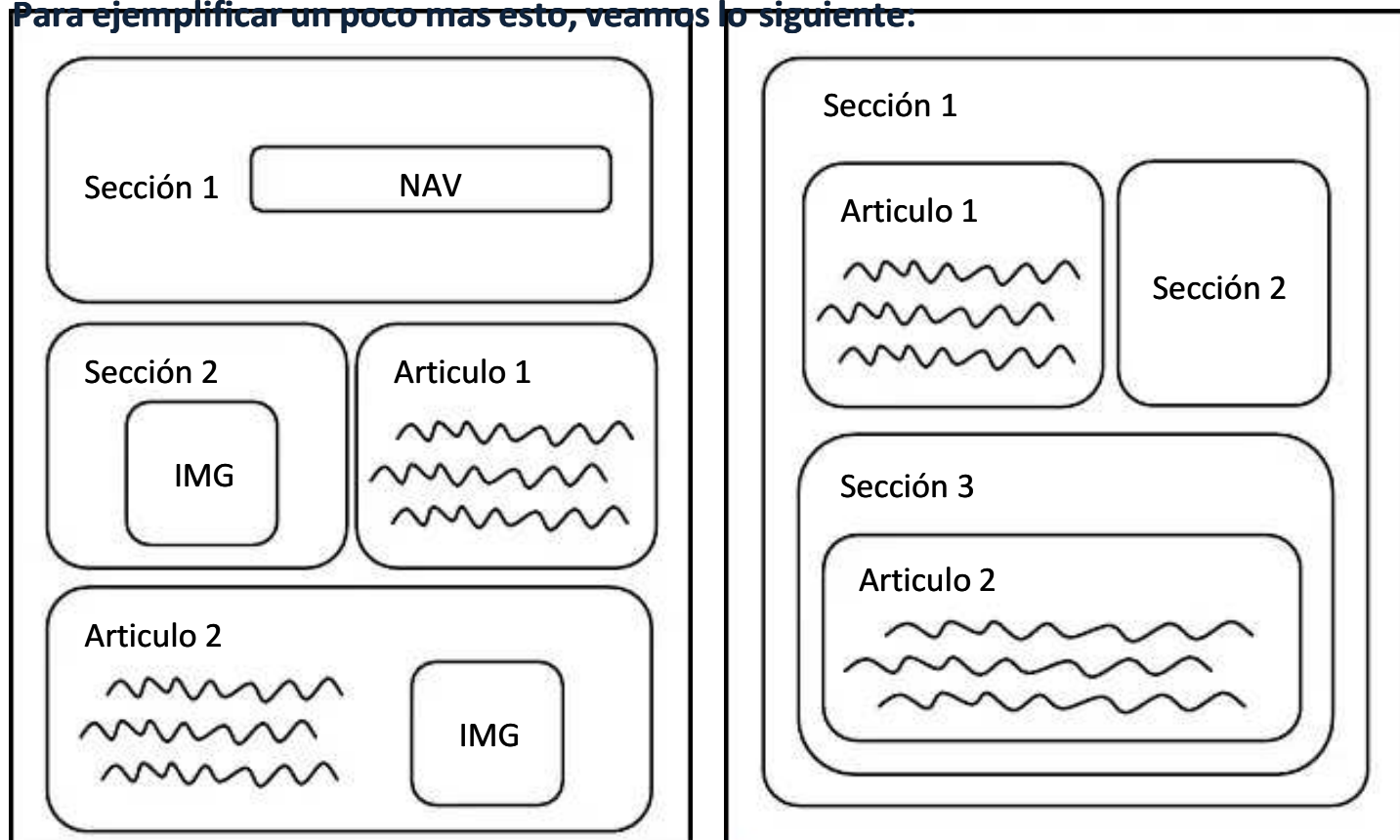
```
<!-- espacio para las imagenes -->
<div class="contenido">
```

Recordar que la etiqueta “div” no ha muerto, pero no por ello debemos abusar de ella, ya que en el nuevo paradigma de html 5 existen nuevas etiquetas tales son el caso de “<section>” y “<article>”, pero te preguntarás, ¿Cuándo utilizar “section” y cuando “article”?

Bueno, muy fácil, “section” como su nombre lo dice (sección) lo estaremos utilizando cuando hagamos una o varias secciones dentro de nuestra página que hagan referencia a contenidos diferentes, y “article” será empleado principalmente cuando demos a denotar un artículo, como el uso de textos.

No obstante, las etiquetas antes mencionadas pueden incluir cualquier otra etiqueta ya sea una etiqueta de la misma semántica u otra diferente como un “figure” o “div”

Para ejemplificar un poco mas esto, veamos lo siguiente:



```
<div id="logo2">
  <div>
    
  </div>
</div>
```

NOTA: Recordar que las etiquetas “img” deben ser contenidas por las etiquetas “figure”.

<-----Apartir de aquí esto debería aparecer dentro de un "article"

```
<div id="bienvenida">
  <h1>Bienvenidos!!</h1>
</div>
<div id="texto">
  <p>En este sitio encontrarás ayuda en video tutoriales de <br> los siguientes temas de
  interés:</p>
```

NOTA: Poner "br" para simular el hecho de que es una línea nueva no es muy recomendable, para ello es mejor jugar un poco con el CSS y obtener el mismo resultado, como por ejemplo hacer más chico la proporción del elemento que lo contenga.

```
    <div id="temas">
      <p> -Photoshop </p>
      <p> -After Effects </p>
      <p> -Cinema 4D </p>
      <p> -Aprendizaje en java </p>
      <p> -Mantenimiento preventivo y correctivo </p>
      <p> -Configuraciones e instalaciones de windows 7 </p>
      <p> -Introducción al mundo LINUX </p>
    </div>
  </div>
```

<-----> Hasta aca deberá terminar el article

Como se dijo anteriormente, no debemos de abusar de las etiquetas <<div>>, ya que si estamos dentro de la etiqueta "article" le podremos poner un "id" para identificarlo y así manipular los elementos por medio del CSS.

Como recomendación a veces es bueno analizar qué es lo que se quiere realizar y de qué manera se obtiene el mejor resultado, en tu caso utilizaste la etiqueta "p" para poner cada uno de los temas, y está bien, en mi caso utilicé una lista desordenada con elementos "li", lo importante de todo esto es obtener el mismo resultado que hayas creado en tu "Mockup".

```
    <div id="fotos1">
      
    </div>
    <div id="fotos2">
      
    </div>
    <div id="fotos3">
      
    </div>
    <div id="fotos4">
      
    </div>
    <div id="fotos5">
      
    </div>
    <div id="fotos6">
      
    </div>
  </div>
```

No es necesario denotar bastantes "id's" si a la hora de analizar nos damos cuenta de que son campos que comparten los mismos valores, por lo cual solo resta pelearnos con el CSS.



NOTA: Recordar que las etiquetas "img" deben ser contenidas por las etiquetas "figure".

</body> 😞

OJO: no es muy recomendable que una etiqueta esté fuera del “body” ya que si lo recuerdas “body” es el cuerpo de nuestra web app, el cual, debe contener a todos los elementos que integra nuestro sitio.

Siendo este el caso, una de dos, o haces un “section” que contenga todo a excepción del footer o bien en el CSS juegas con el “width” según el porcentaje a que corresponda, como lo hice yo.

<footer> 😊

<p>Tuto. Todos los derechos reservados</p> 😊

</footer> 😊

</html> 😊