

MP_0373. Lenguajes de marcas y sistemas de gestión de información

UF2. Lenguajes para la visualización de la información

2.4. Caso práctico



Índice

Objetivos	3
Creación de una página web	4
Preparando los archivos	4
Creando la maqueta inicial HTML5	4
El <i>head</i>	5
El <i>body</i>	6
El menú	10
Metiendo algo de contenido.....	14
Haciendo nuestra página responsive.....	17
Despedida	18

Objetivos

En esta sesión vamos a poner en práctica los conocimientos adquiridos a lo largo de la unidad.

Creación de una página web

Preparando los archivos

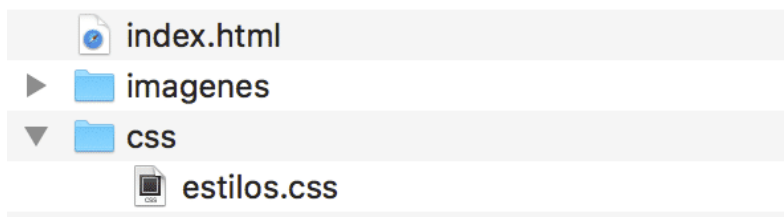
Para empezar nuestro proyecto, lo primero es crear los archivos y ordenarlos.

Aunque la página que vamos a crear no constará de muchos archivos, lo normal en un proyecto real es que tengamos que manejar una gran cantidad de ellos. Por eso, mantener una estructura de directorios adecuada es de suma importancia.

Archivos y carpetas

Aunque no existe una única forma de crear la estructura, en este caso aconsejamos separar los archivos HTML de los CSS con los estilos y de los de imágenes.

En un futuro podríamos querer crear funciones JavaScript u otros archivos, para los cuales también crearíamos su propia carpeta.



Estructura inicial de nuestra organización de archivos.

El archivo de inicio de nuestra página es `index.html`. Este nombre de archivo es el usado por defecto por los servidores para localizar el comienzo de la jerarquía del sitio web (la *home* o página de inicio).

Creando la maqueta inicial HTML5

El siguiente paso es crear la estructura principal de la página. En nuestro caso vamos a continuar con el esquema de la típica página web en formato blog.

Donde tenemos:

- Una cabecera (*head*) para poner el nombre del sitio, logos, etc.
- Un menú para insertar los diferentes enlaces a otras páginas del sitio.
- Una sección principal, donde pondremos el contenido principal.
- Un *aside* (sección de contenido relacionado) para los elementos secundarios como *banners*, *call to action*, listados, etc.

- Un *footer* (pie de página) donde insertaremos el *copyright* y otros elementos como menús secundarios, acceso a redes sociales, etc.

El *head*

Como ya hemos comentado, toda página se divide en un *head* (cabecera), donde pondremos el título, las llamadas a archivos externos, etc. y el *body* (cuerpo), donde irá el contenido visible de la página. En este caso veamos nuestro *head*.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Mi primera página web</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="keywords" content="Palabras clave para ayudar a saber el
tema sobre el que trata la página" />
    <meta name="description" content="Descripción de la página" />

    <link rel="stylesheet" type="text/css" href="css/estilos.css">

  </head>
  <body>

  </body>
</html>
```

En el código podemos ver cómo hemos insertado, además de la llamada al archivo CSS, etiquetas *meta* con información del documento como la codificación de caracteres, la escala inicial del documento, la descripción del contenido de la página, etc.

Aunque la *meta keywords* ya no es relevante para el posicionamiento en buscadores, la *description* sí es de utilidad para el SEO.

El *body*

Ahora insertaremos las etiquetas semánticas HTML5 para poner luego el contenido. En este caso empezamos el diseño pensando en la visualización para escritorio "ancho fijo", es decir le daremos un ancho para que se ajuste a resoluciones grandes, dejando márgenes a los lados si es necesario.

```
<body>

  <div id="contenedor">

    <header>
      <!-- Cabecera de la página-->
    </header>
    <nav>
      <!-- Menú de la página -->
    </nav>

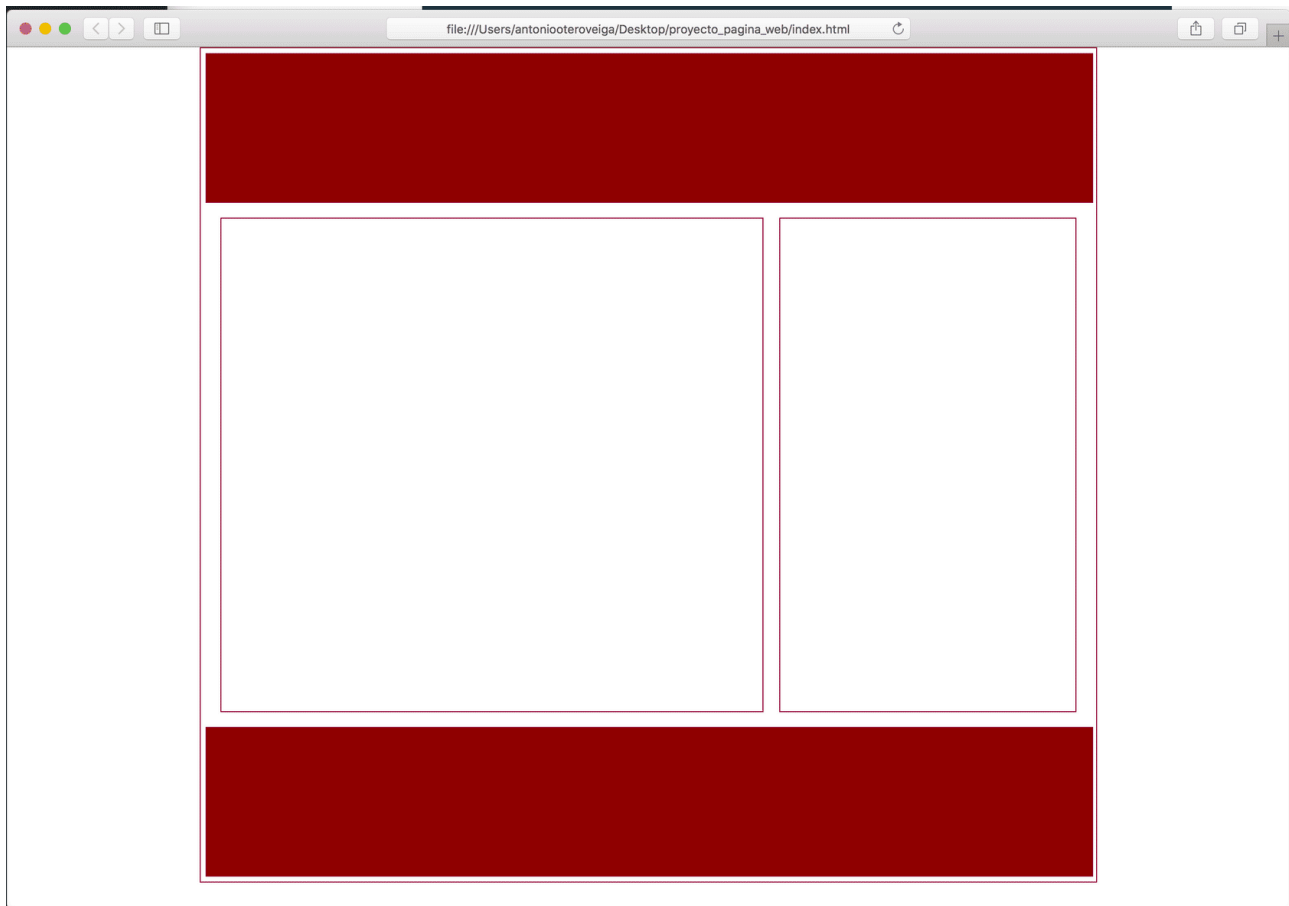
    <section>
      <!-- Contenido principal -->
    </section>
    <aside>
      <!-- información menos relevante de la página -->
    </aside>
    <footer>
      <!-- Pie de la página -->
    </footer>

  </div>

</body>
</html>
```

Aplicando CSS a los elementos principales

Ahora vamos a crear con los estilos CSS la estructura de nuestra página. Introduciremos bordes, fondos y altos a los elementos con la intención de que vayamos comprobando la estructura mientras la hacemos, aunque en algunos casos los eliminaremos cuando el documento esté terminado.



Veamos cómo quedaría la hoja de estilos.

```
@charset "UTF-8";
/* CSS Document */

*{
margin:0px;
padding:0px;
font-family:Arial, Helvetica, sans-serif;
}

#contenedor{
border: 1px solid #903;
margin-left:auto;
margin-right:auto;
width:900px;
background-color:#FFF;
padding: 5px;
}

header {
border: 1px solid #903;
width:100%;
margin-bottom: 15px;
height: 150px;
background-color:rgba(153,0,0,1);
}
```

```

        overflow: hidden;
    }

    section{
        border: 1px solid #903;
        width: 550px;
        float: left;
        height: 500px;
        margin-left: 15px;
        margin-bottom: 15px;
    }

    aside{
        border: 1px solid #903;
        margin-right: 15px;
        width: 300px;
        float: right;
        height: 500px;
        margin-bottom: 15px;
    }

    footer {
        border: 1px solid #903;
        clear: both;
        width: 100%;
        height: 150px;

        background-color: rgba(153,0,0,1);
    }

```

Archivo estilos.css.

Veamos ahora cada regla con más detalle.

```

*{

    margin:0px;
    padding:0px;
    font-family:Arial, Helvetica, sans-serif;

}

```

La primera regla que hemos incorporado está con el selector universal, dado que deseamos que se aplique a todos los elementos.

Ponemos los márgenes y *padding* a 0 para eliminar las herencias no deseadas incorporadas en párrafos, listas, etc.

Es importante entender que al eliminar estas herencias propias de los elementos HTML ganaremos en libertad a la hora de diseñar el documento, pero nos obliga a incorporar luego todos los estilos de los elementos según nuestra preferencia.


```
#contenedor{  
    border: 1px solid #903;  
    margin-left:auto;  
    margin-right:auto;  
    width:900px;  
    background-color:#FFF;  
    padding: 5px;  
}
```

Ya comentamos que hemos puesto todos los elementos en un contenedor para facilitar el diseño y hacerlo de "ancho fijo". Para este elemento hemos usado un selector de *id*.

Para centrar el elemento en la ventana del explorador los márgenes derecho y izquierdo los ponemos en *auto*, esto forzará el centrado del contenedor.

```
header {  
    border: 1px solid #903;  
    width:100%;  
    margin-bottom: 15px;  
    height: 150px;  
    background-color: rgba(153,0,0,1);  
    overflow: hidden;  
}
```

Mediante un selector de *tag* (etiqueta), damos estilo al *header*, de momento esta vacío.

En este caso se le ha puesto un alto e indicado que el contenido se adapte a su tamaño gracias al *overflow*.

```
section{  
    border: 1px solid #903;  
    width: 550px;  
    float: left;  
    height: 500px;  
    margin-left: 15px;  
    margin-bottom: 15px;  
}
```

Para el *section* le hemos puesto un ancho de 550px, también podríamos jugar con otras unidades de medida como los %.

Para la versión de escritorio se ha indicado que se coloque a la izquierda del documento poniendo *float:left*.

```
aside{
  border: 1px solid #903;
  margin-right: 15px;
  width: 300px;
  float: right;
  height: 500px;
  margin-bottom: 15px;
}
```

El *aside* es similar al *section*, aunque con diferente tamaño y flotación. En este caso lo colocamos a la derecha y hemos puesto un tamaño de 300px.

Es importante dejar aire entre los elementos si el diseño nos lo permite. En el contenedor hemos puesto un ancho de 900px, y la suma del *aside* y *section* son (550px + 300px) 850px, por lo que hemos dejado entre los elementos un espacio de 50px. Esta medida dependerá del diseño que estés realizando.

```
footer {
  border: 1px solid #903;
  clear: both;
  width: 100%;
  height: 150px;

  background-color: rgba(153,0,0,1);
}
```

Para el *footer* le damos un tamaño del 100% del contenedor.

Es importante aplicar la regla *clear:both*, dado que al estar después de dos elementos flotantes, debe recuperar su posición sin flotación.

El menú

Un sitio web consta de varias páginas, por lo que resulta útil disponer de un menú que nos permita la navegación entre los diferentes documentos.

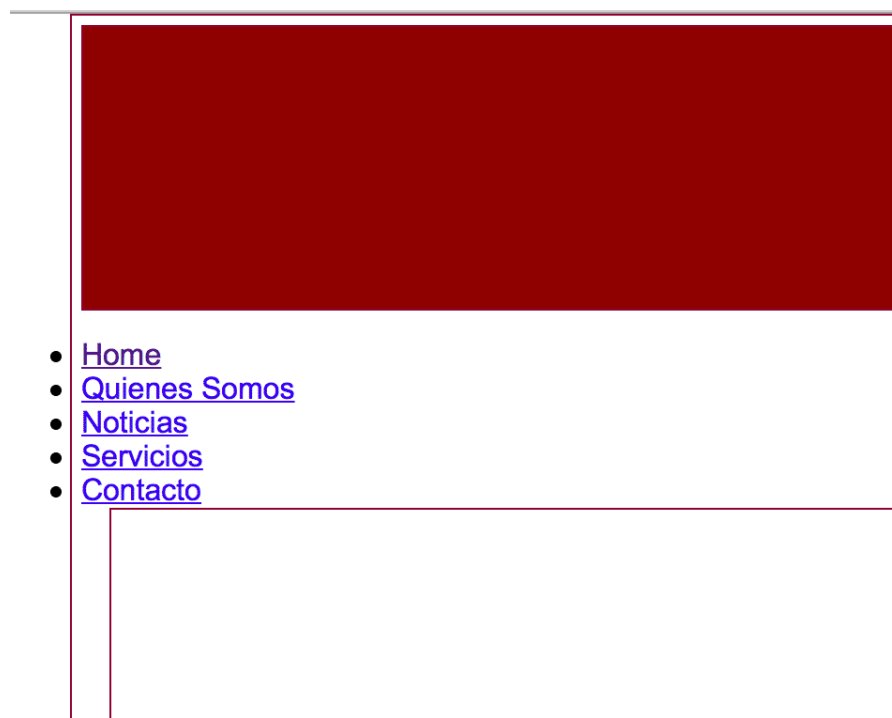
Para ello vamos a insertar una etiqueta `<nav>`, donde colocaremos los enlaces que facilitarán esta labor de navegación. Para el ejemplo que estamos elaborando hemos decidido que nuestro sitio constará de las siguientes páginas:

- Home (index.html).
- Quiénes somos (quienes.html).
- Noticias (noticias.html).

- Servicios (servicios.html).
- Contacto (contacto.html).

Primero insertamos en el HTML la etiqueta *nav* y una lista con los enlaces.

```
<nav>
    <!-- Menú de la página -->
    <ul>
        <li><a href="index.html">Home</a> </li>
        <li><a href="quienes.html">Quiénes somos</a> </li>
        <li><a href="noticias.html">Noticias</a> </li>
        <li><a href="servicios.html">Servicios</a> </li>
        <li><a href="contacto.html">Contacto</a> </li>
    </ul>
</nav>
```



Una vez tenemos la lista con nuestros enlaces, procederemos a aplicar los estilos para hacerlo más agradable.

```
nav{
    width: 100%;
}

nav ul{
    list-style: none;
}

nav ul li{
```

```
float: left;
width: 20%;
text-align: center;
}

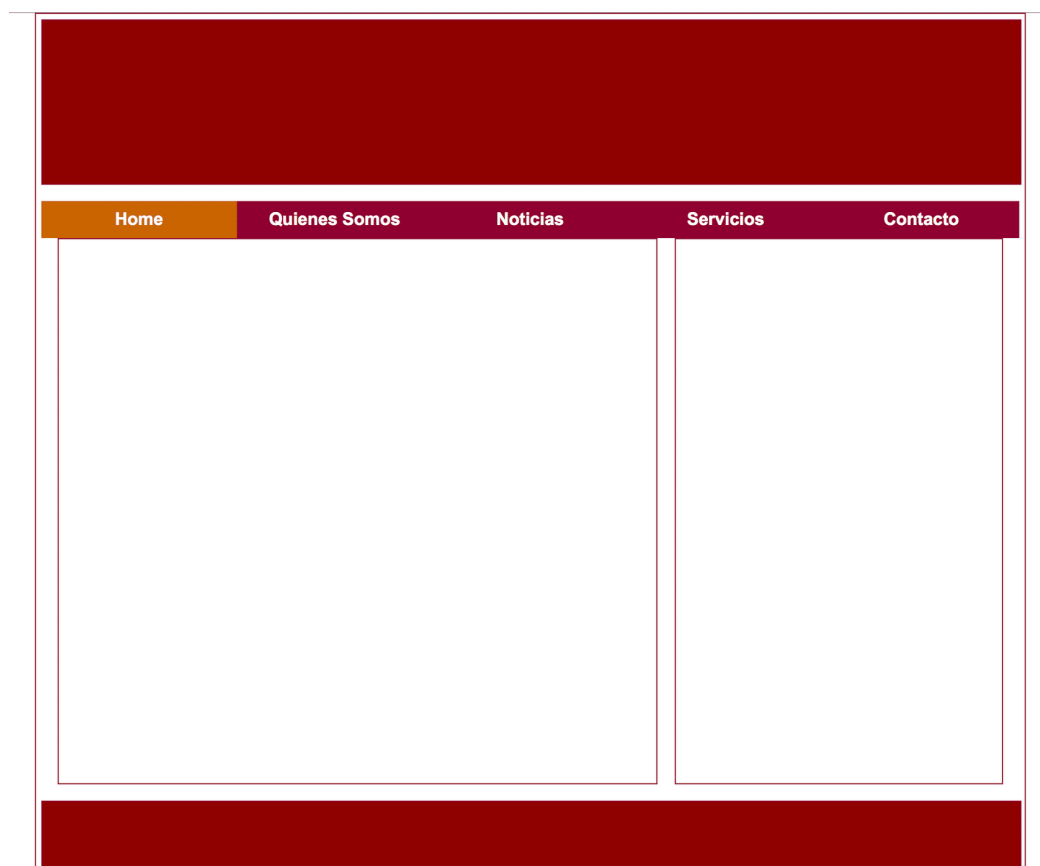
nav ul li a{

display: block;
padding-top: 8px;
padding-bottom: 8px;
font-weight: bold;
color: white;
background: #903;
text-decoration: none;

}

nav ul li a:hover{
background-color: chocolate;
}
```

Una vez aplicados los estilos, este sera su nuevo aspecto.



Examinemos con más detalle lo que hemos hecho.

```
nav{
  width: 100%;
}
```

Indicamos a la etiqueta `<nav>` que ocupe el 100% del contenedor.

```
nav ul{
  list-style: none;
}
```

Quitamos los elementos visuales que tienen las listas, como el punto.

Las listas también traen herencias como el *margin* y *padding*, no lo estamos eliminando aquí, pues ya lo hicimos en el selector universal (*), donde pusimos todos los *margin* y *padding* a 0.

```
nav ul li{
  float: left;
  width: 20%;
  text-align: center;
}
```

Dado que se trata de un menú horizontal, hacemos que los elementos `` floten hacia la izquierda.

Dado que son 5 elementos en total, le hemos puesto un tamaño del 20% a cada uno para que cubran todo el espacio.

Al usar el *text-align* indicamos que el texto quede centrado dentro de cada etiqueta ``.

```
nav ul li a{
  display: block;
  padding-top: 8px;
  padding-bottom: 8px;
  font-weight: bold;
  color: white;
  background: #903;
  text-decoration: none;
}
```

Para que el enlace ocupe toda la superficie del espacio reservado por la etiqueta ``, convertimos la etiqueta `<a>` en bloque. De no hacerlo, no podría ocupar más que el propio texto.

Otra regla que es importante mencionar es la de *text-decoration: none*, donde quitamos herencias no deseadas, como el subrayado que traen los enlaces por defecto.

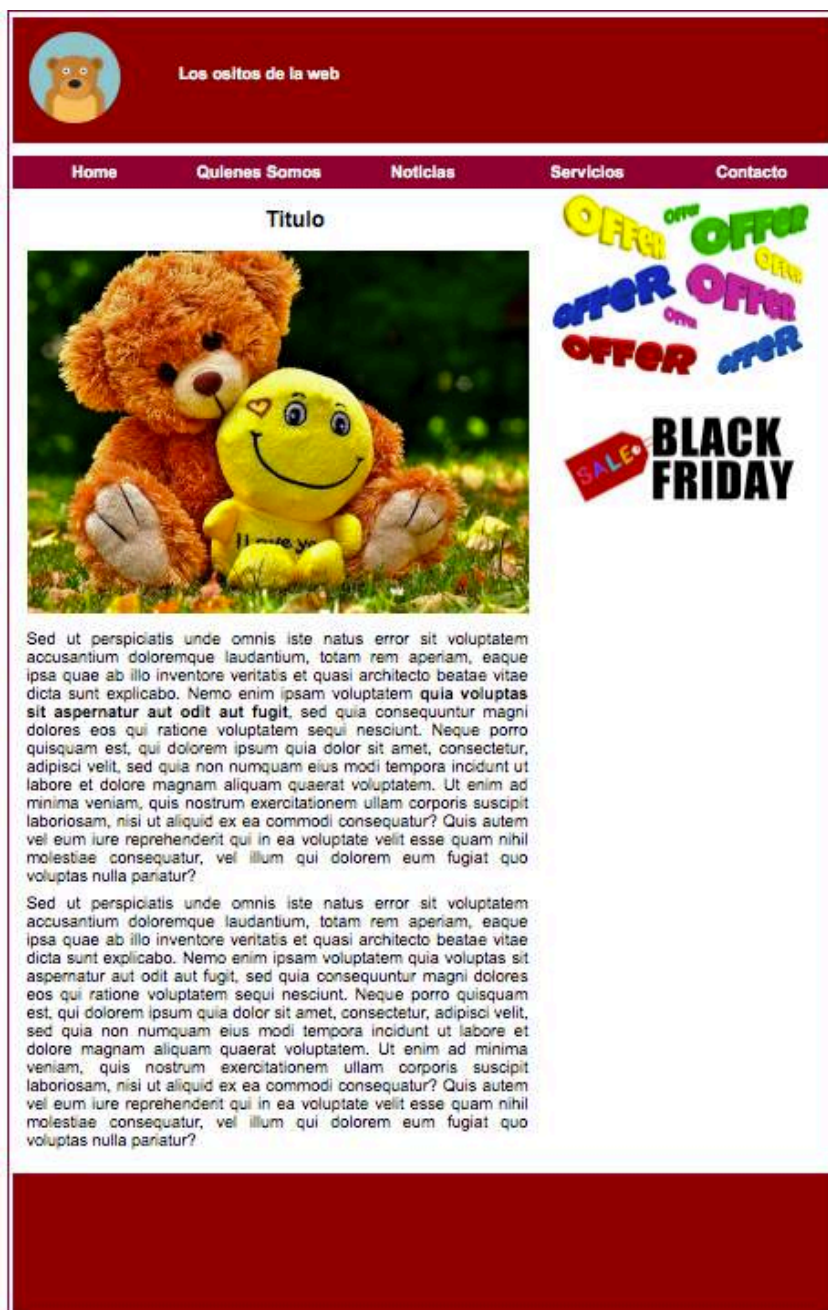
```
nav ul li a: hover{
  background-color: chocolate;
}
```

Para dotar de algo de dinamismo al menú le incluimos el comportamiento `:hover`. De esta manera, cuando el cursor del ratón se sitúe encima del elemento aplicará las reglas indicadas; en este caso un cambio del color de fondo del botón.

Metiendo algo de contenido

Dado que resultaría algo tedioso realizar un proyecto real, vamos a dar algo de personalidad a nuestra página, aunque sea con algo de texto ficticio.

En el `<header>` vamos a incluir un texto y una imagen a modo de logo. Además meteremos texto en el apartado `<section>`, junto con alguna foto.



Llenando el <header>

Como ejemplo, en el <header> insertamos un logotipo y un texto de encabezado. Podemos ver que usamos la etiqueta *float* junto con los márgenes para situar los elementos.

Recordemos que si no tenemos el *overflow* en el estilo del <header>, este perdería su altura.

```
<header>
    <!-- Cabecera de la página-->
    <div class="logo">
        
    </div>
    <h3>
        Los ositos de la web
    </h3>
</header>
```

Contenido del <header> en el documento HTML.

```
header .logo{
float: left;
margin: 15px;
width: 100px;
}

header .logo img{
width: 100%;
}
```

Estilos para el <header> en el archivo CSS.

Llenando el <section>

Para llenar el <section> colocaremos una imagen y un par de párrafos. Como hemos quitado los estilos heredados tenemos que configurar los márgenes de las etiquetas <p>.

También se han quitado las etiquetas de borde y alto de <section> para que la altura de la página se ajuste al contenido.

```
<section>
    <!-- Contenido principal -->
    <h1>Titulo</h1>
    
    <p> Sed ut perspiciatis unde omnis iste natus error sit voluptatem
accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo
inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo
enim ipsam voluptatem <strong>quia voluptas sit aspernatur aut odit aut
fugit</strong>, sed quia consequuntur magni dolores eos qui ratione voluptatem
sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet,
consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut
labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam,
quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid
ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea
voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum
fugiat quo voluptas nulla pariatur?</p>
```

```
<p> Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium  
doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore  
veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam  
voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia  
consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque  
porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci  
velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore  
magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum  
exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi  
consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit  
esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo  
voluptas nulla pariatur?</p>  
</section>
```

Contenido `<section>` en el documento HTML.

```
section h1{  
    width: 100%;  
    text-align: center;  
    margin: 20px;  
}  
  
section p{  
    text-align: justify;;  
    margin-bottom: 10px;  
}  
  
section img{  
    width: 100%;  
    margin-bottom: 15px;  
}
```

Estilos para el `<section>` en el archivo CSS.

Llenando el `<aside>`

En el caso de `<aside>` pondremos dos imágenes imitando dos *banners*.

```
<aside>  
  <!-- información menos relevante de la página -->  
  <a href="enlace.html"></a>  
  <a href="enlace.html"></a>  
</aside>
```

Contenido del `<aside>` en el documento HTML.

```
aside img{  
    width: 100%;  
    margin-bottom: 20px;  
}
```

Estilos para el `<aside>` en el archivo CSS.

Haciendo nuestra página responsive

Para hacer nuestra página *responsive* introduciremos en nuestro CSS las *media queries*, indicando los tamaños de las pantallas de los dispositivos.

En este caso adaptaremos la página para dispositivos móviles con un ancho máximo de 480px, y para tabletas con un ancho máximo de 700px.

```
@media screen and (max-width: 700px) {
  section {
    width: auto;
    float: none;
  }
  aside {
    width: auto;
    float: none;
  }
}
@media screen and (max-width: 480px) {
  #contenedor{
    width: 100%;
    border:none;
    margin: 0px;
    padding: 0px;
  }
  header {
    margin: 0;
  }
  header .logo{
    margin: auto;
    float: none;
    width: 100px;
  }
  nav{
    display: none;
  }
  header h3 {
    display: none;
  }
  aside {
    display: none;
  }
  section{
    width: 100%;
    margin: 0;
    padding: 5px;
  }
}
```

Gracias a la propiedad de cascada podemos reemplazar los estilos, aplicando nuevas propiedades dependiendo del tamaño del dispositivo.

Despedida

Resumen

Has terminado la lección, veamos los puntos más importantes que hemos tratado.

Una página web puede tener todo el contenido que queramos poner. En esta unidad hemos realizado paso a paso una pequeña muestra de lo que nos permite la tecnologías HTML y CSS.

Hemos hecho un recorrido desde la creación de la estructura básica de una página hasta la inserción de algo de contenido.

Ahora es importante que practiques realizando tus propias creaciones, combinado todas las herramientas comentadas en esta unidad.

Si quieres conocer más etiquetas HTML o reglas CSS, puedes encontrar cientos de manuales y libros por Internet, aunque debes estar atento que estén actualizados a HTML5.

Para conocer más etiquetas HTML y reglas CSS, puedes consultar W3Schools.

<https://www.w3schools.com/>