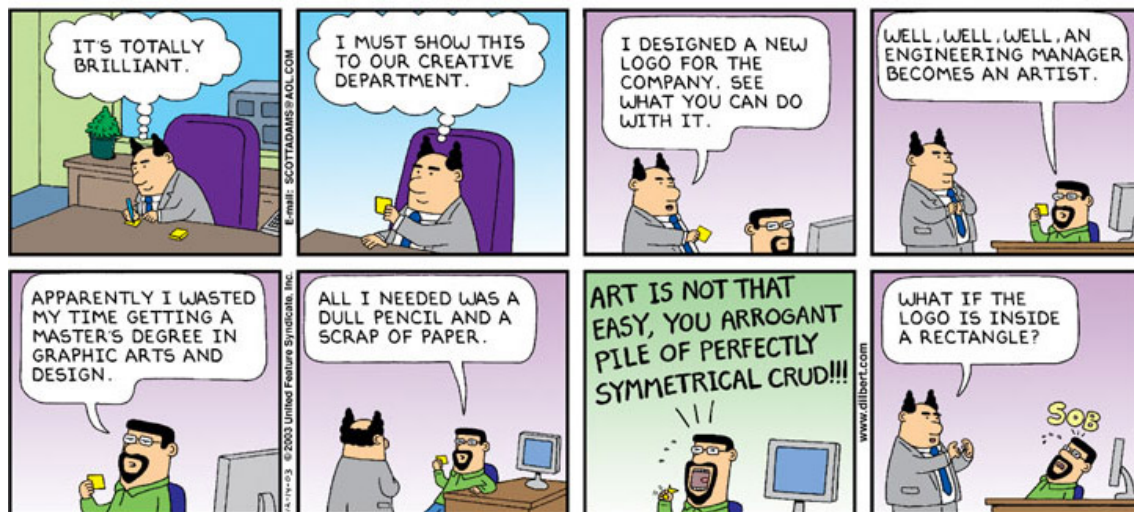


Comercio Electrónico

Práctica 4: Edición de contenidos web



Antonio Sanz
ansanz@unizar.es

Rafael del Hoyo
rdelhoyo@ita.es

Objetivo de la práctica

En esta práctica se pretende realizar una introducción a las tecnologías existentes para la creación de contenidos web tanto estáticos como dinámicos. Se proporcionará al alumno unos conocimientos básicos de HTML y CSS para que pueda aplicarlos al proyecto de comercio electrónico.

¿Qué hay que preparar de forma previa a la práctica?

Dado que en esta práctica se va a empezar a trabajar con el servidor web y con el servidor de FTP instalados en la práctica anterior, es necesario repasar dicha práctica y asegurarse de que ambos servicios se están ejecutando correctamente.

Será de especial importancia el tener bien configurado tanto la directiva *Document Root* del Apache como el usuario y contraseña (junto con sus permisos de acceso) del servidor de FTP para que el acceso a la web por FTP sea correcto.

¿Cuál es el resultado de la práctica?

El resultado es un boceto de diseño de la página principal del proyecto de comercio electrónico, empleado HTML y CSS.

¿Qué se aprende con esta práctica?

Se aprenden unos principios de diseño y construcción de páginas web empleando las tecnologías más usadas en la actualidad y empleando un entorno de desarrollo totalmente libre y configurable.

Introducción

Una vez tenida la idea y habiendo pensado y organizado toda la información que vamos a ofrecer en nuestro proyecto de comercio electrónico, llega el momento de plasmar todos esos contenidos en la creación de la propia página web. Para ello se va a emplear el lenguaje HTML (*HyperText Markup Language*, o *Lenguaje de Marcas de HyperTexto*), que en su versión 4.0 ya empieza a tener una acepción casi total por todos los programas de diseño de páginas web (y mucho más importante, por los navegadores).

Después de ver los rudimentos de HTML pasaremos a las hojas de estilo (también llamadas CSS o *Cascading Style Sheets*), que permiten definir propiedades a nivel de un sitio web completo, ahorrando una importante cantidad de trabajo a la hora de definir el aspecto global de un sitio web.

HTML (HyperText Markup Language)

HTML es un lenguaje interpretado, no compilado. Cuando un cliente se conecta a nuestro servidor y pide que se le entregue una página web mediante el protocolo HTTP, el servidor busca la página y la entrega al navegador. Es tarea entonces del navegador empleado el interpretar dicha página y mostrarla por pantalla al cliente (es por eso que una página web se puede ver de diferentes formas en función de los navegadores y las características de los ordenadores empleados).

El lenguaje HTML es lo que se denomina un “lenguaje de marcas”. Todos los comandos de HTML están regidos por lo que se llaman *tags* o etiquetas, de forma que un ejemplo de *tag* HTML podría ser:

< COMANDO> Texto a tratar < / COMANDO >

Los comandos van entre los caracteres “<” y “>”, y se distingue el inicio del comando del fin del mismo mediante el carácter “/”. A continuación vamos a ver los *Tags* HTML más empleados en el diseño de páginas web.

Editor Web - PHP DESIGNER

Aunque es posible diseñar páginas completas con un editor de textos totalmente básico (como el “Bloc de Notas” de Windows o el “vi” de Linux), siempre es recomendable el tener un entorno de desarrollo lo más amigable posible para que nos facilite el desarrollo de la página web.

Dentro del desarrollo de páginas web es muy interesante el encontrar editores que tengan al menos las siguientes características:

- Ser WYSIWYG (“What you see is what you get” o “Lo que ves es lo que hay”): Estos editores permiten ver el aspecto de la web que estamos construyendo en el propio programa sin tener que subir el contenido a la página web.
- Tener la capacidad de soportar visores externos: Al poder ver el aspecto de nuestra web en nuestro navegador local, podemos estudiar la compatibilidad con los navegadores (así como ver el aspecto definitivo de la web).
- Tener clientes integrados de FTP/SSH: De esta forma desde el propio programa podemos conectarnos al servidor y actualizar nuestra web.

A lo largo de la práctica se va a emplear el editor PHP Designer (que usaremos tanto para el HTML como para el PHP), que tiene las características indicadas anteriormente.

La instalación del programa no tiene misterio alguno (aunque sí que será necesario instalarlo en un directorio diferente para cada grupo de prácticas). Una vez instalado el PHP Designer, lo primero que se tendrá que hacer es crear un sitio web dentro del programa (un directorio bajo el cual estén guardadas todas las páginas que se vayan creando).

En primer lugar, tendremos que asegurarnos de que tanto el servidor web como el servidor de FTP están arrancados correctamente (son necesarios para el desarrollo de la práctica). A continuación arrancaremos el PHP Designer y accedemos al botón de “FTP”, dentro de la pestaña de “Task Panels”. Una vez allí pulsaremos sobre “Connect to Server”, indicando los parámetros necesarios para la conexión:

- Site Name: El nombre que le queramos dar a nuestro sitio web.
- Hostname: El nombre o dirección IP de nuestro servidor FTP.
- UserName: Usuario definido en el servidor web
- Password: Contraseña de dicho usuario
- Port: El puerto en el que está escuchando el servidor FTP (21 por defecto).

Una vez introducidos los datos, se pulsará sobre “New Site” para que lo cree. Si todo ha ido bien en la pestaña de la izquierda aparecerá nuestro sitio web definido con la estructura de carpetas del XAMPP (que deberemos borrar para dejar nuestro sitio web totalmente limpio y listo para usar).

Estructura básica de una web

Los *tags* HTML básicos (el esqueleto mínimo que tiene que tener toda página web) son los siguientes:

```
<html> → Indica el principio del código HTML
<head> → Indica la cabecera
<title> → Indica el título que tiene la propia web en el navegador
<body> → El inicio del texto de la propia web
```

Un “Hola Mundo!!” en HTML sería algo tan sencillo como :

```
<html>
<head>
  <title> Mi primera web</title>
</head>
<body>
  Hola , mundo !!
</body>
</html>
```

Una vez hayamos depurado este código, se tendrá que guardar dicha página en local (crear un directorio en D:\ que tenga vuestro nombre y grupo de práctica) antes de poder subirlo al servidor. Para poder publicar (o “subir” en la jerga) vuestra web tan solo será necesario pulsar en la pestaña de “FTP” de los “Task Panels” del PHP Designer, seleccionando la opción de “Upload Files” (es necesario haber configurado y probado la conexión por FTP con anterioridad).

Nota: En este caso concreto tanto el programa con el que se crea la web como el propio servidor están en el mismo equipo, siendo los ficheros que tengamos en el servidor web y en nuestra carpeta local exactamente los mismos. Se podría plantear la solución de “copiar de un directorio a otro”, pero dicha solución solo valdría para este caso, ya que lo normal es que el servidor web esté en otro equipo remoto, siendo necesario el uso de FTP para la publicación de contenidos.

Tratamiento de texto en HTML

Como se ha podido comprobar en el ejemplo anterior HTML deja escribir texto tal cual, sin ponerle pega alguna (pero también sin ponerle formato). Los siguientes tags HTML permiten actuar sobre el texto:

<P> </P> → Nuevo párrafo. Muy útil para separar texto. (paragraph)

 → Salto de línea (es uno de los pocos tags que no tiene un </ >) (break)
 → Texto en negrita (bold)
<i> </i> → Texto en cursiva (italic)
<CENTER> </CENTER> → Texto centrado

Es interesante el indicar que los tags de HTML no son sensibles a las mayúsculas (y producirán el mismo resultado).

Caracteres especiales

Como ya veremos en la parte de formularios, hay ciertos caracteres especiales que se muestren correctamente tienen que ser codificados de forma previa. Algunos de estos caracteres pueden ser los siguientes:

& → &
" → "
 → (espacio en blanco, se usa mucho como "separador" en tablas)
á → á
é → é
í → í
ó → ó
ú → ú
ñ → ñ

Afortunadamente PHP Designer nos realiza la codificación de forma automática para este tipo de caracteres.

Comentarios

Es recomendable el introducir de cuando en cuando comentarios dentro del código HTML para hacerlo más legible (y facilitar las posibles modificaciones). Se considera como comentario todo lo que esté entre estas cadenas:

```
<!-- Esto es un comentario -->
```

Estos comentarios obviamente no saldrán reflejados en la pantalla. Se recomienda encarecidamente realizar un comentado inteligente del código generado (redundará de forma positiva en el desarrollo y mantenimiento del sitio web).

Enlaces

La forma principal de conducir a los usuarios por nuestra página web se realiza a través de los enlaces. Un enlace puede tener la forma siguiente:

```
<A HREF="http://www.unizar.es/"> Universidad de Zaragoza </A>
```

donde el texto que sigue a HREF es la URL y el texto entre el ">" y el es la representación de ese enlace en la web (se pueden poner ambos textos iguales sin mucho problema).

Un enlace puede apuntar a otra página web dentro de nuestro sitio web. En este caso, es muy importante el establecer la ruta o *path* relativo de forma correcta.

```
<A HREF="login.html"> Inicio de Sesión </A>  
<A HREF="catalogo/inicio.html"> Nuestros productos </A>
```

También es posible que un enlace apunte a una web externa a la nuestra :

```
<A HREF="http://www.google.es"> Si lo encuentra, busque aquí </A>  
<A HREF="ftp://ftp.rediris.es"> Rediris FTP Server </A>
```

Y en algunos casos, que apunte a una dirección de correo :

```
<A HREF="mailto:info@miproyecto.com"> Contacte con nosotros </A>
```

Listas de elementos

Una forma de presentar información ordenada en HTML puede conseguirse mediante el uso de listas ordenadas (con o sin numeración). La sintaxis es la siguiente:

- Lista sin numerar:

```
<ul>  
  <li>Uno</li>  
  <li>Dos</li>  
  <li>Tres</li>  
</ul>
```

- Lista numerada

```
<ol>  
  <li>Uno</li>  
  <li>Dos</li>  
  <li>Tres</li>  
</ol>
```

Imágenes

El insertar imágenes dentro de una página web es algo extremadamente sencillo. El tag HTML es el siguiente:

```
<IMG SRC="imagen.gif" ALT="descripcion" ALIGN=TOP WIDTH=200  
HEIGHT=115>
```

con los siguientes parámetros:

- SRC = Imagen a incluir (con una ruta relativa, recordad)
- ALT = En caso de que la imagen no exista o no se pueda mostrar, texto que se muestra como alternativa a la imagen. Recomendable ponerlo para mejorar la usabilidad y accesibilidad a la web.
- ALIGN = Alineación de la foto con respecto al texto
- WIDTH & HEIGHT = Altura y anchura de la imagen. Si la imagen es más grande que esos parámetros se verá empujada, y si es más pequeña se verá estirada. Se recomienda el emplear programas de retoque fotográfico para dejar la imagen al tamaño exacto necesitado.

Es muy fácil hacer que una imagen tenga insertado un enlace. Para ello unimos ambos tags HTML de la forma siguiente:

```
<A HREF="ejemplo.html"><IMG SRC="imagen.gif" BORDER=0></A>
```

(el BORDER=0 se pone para que la imagen no tenga un borde azul, verifícalo)

Tablas

Una de las formas más comunes de presentar información en una web es a través del uso de tablas (ya veremos en la práctica siguiente su fácil integración con PHP). La sintaxis básica de una tabla es la siguiente:

```
<TABLE BORDER>  
<TR>  
    <TD>Fila 1 - Celda 1</TD>  
    <TD>Fila 1 - Celda 2</TD>  
</TR>  
  
<TR>  
    <TD>Fila 2 - Celda 1</TD>  
    <TD>Fila 2 - Celda 2</TD>  
</TR>  
  
<TR>  
    <TD>Fila 3 - Celda 1</TD>  
    <TD>Fila 3 - Celda 2</TD>  
</TR>  
</TABLE>
```

donde

- TABLE = Indicador de principio y fin de tabla

- TR = Nueva fila
- TD = Nueva columna

Se recomienda encarecidamente el ser muy cuidadosos con la sintaxis de las tablas, así como indexar el código correctamente (es muy fácil tener tablas anidadas, y en esos casos seguir la legibilidad sin un código indexado es altamente complicado).

Capas

En los principios del diseño de páginas web se empleaba para la creación de zonas de contenido lo que se denominaban *frames* o marcos. Aun cuando eran sencillos de crear, presentaban inconvenientes de cara al usuario (algunos navegadores no los entendían bien, otros los separaban en ficheros diferentes, etc.).

La creación de zonas de contenido se realiza en la actualidad mediante lo que se denominan capas o bloques de información mediante el tag <DIV>. Lo fundamental de las capas es que pueden tener estilos (que permiten definir perfectamente todas las características del HTML generando un código muy limpio).

CSS – Hojas de Estilo

Cuando se están diseñando sitios web complejos con una gran cantidad de páginas, es fácil ver que un cambio en el aspecto del sitio (por ejemplo, cambiar de color el fondo) obliga al diseñador a cambiar todas las páginas del sitio, una tarea laboriosa y repetitiva.

El tener que realizar esta labor a mano está causado por la unidad entre la estructura de la página y el aspecto de la misma, por lo que sería interesante el disponer de una herramienta que permitiera realmente separar el estilo de la estructura. Esta herramienta son las hojas de estilo o *Cascading Style Sheets*.

Las características principales de una CSS son las siguientes:

- Definen COMO mostrar elementos HTML.
- Se guardan en un fichero individual (no tienen porqué formar parte del código).
- Son parte del estándar HTML 4.0.
- Son “apilables”.

La sintaxis base de una CSS es la siguiente:

```
selector {propiedad: valor}
Ej: body {color: black}
```

Si el valor tiene más de una palabra, hay que ponerlo entre comillas, y si queremos poner más de una propiedad, tenemos que usar punto y coma. Es también razonable para estilos complejos el reescribir cada propiedad en una línea:

- Ej1: p {font-family: "sans serif"}
- Ej2: p {text-align:center;color:red}

- Ej3:


```
p
{
text-align: center;
color: black;
font-family: arial
}
```

Si queremos tratar varios elementos de un mismo tipo de forma diferente (por ejemplo, un párrafo alineado a la derecha y otro a la izquierda) podemos emplear el selector *class* de la forma siguiente:

- Ej4:


```
p.right {text-align: right}
p.center {text-align: center}
```

Y añadir en nuestro HTML la etiqueta *class* para saber dónde aplicar cada estilo de la CSS:

```
<p class="right">
Este párrafo ira alineado a la derecha.
</p>
<p class="left">
Este párrafo ira alineado a la izquierda .
</p>
```

Otra opción muy útil es la de crear estilos que no estén atados a elementos HTML (para poder emplearlos donde nosotros queramos). Estos estilos se crean empleando solamente el selector *class* de la forma siguiente:

- Ej5:


```
En la CSS:
.centrado {text-align: center}

En el HTML:
<p class="centrado">
Este párrafo ira centrado
</p>
```

También es posible el introducir comentarios dentro de la propia CSS (algo que se recomienda para la legibilidad de la misma).

- Ej6:


```
/* Esto es un comentario en una CSS */
```

La práctica más común en la actualidad para manejar hojas de estilo o CSS es tener todos los estilos en un fichero dentro de nuestro sitio web y hacer referencia al mismo dentro de nuestro código HTML. Esta llamada se puede realizar de la forma siguiente:

- Ej7:


```
<head>
<link rel="stylesheet" type="text/css" href="mihoja.css" />
</head>
```

Las hojas de estilo son anidables, es decir, podemos tener varias CSS que se aplican sobre un mismo documento (de ahí viene la C de “Cascading” o “en

cascada"). El orden de prioridades será el siguiente (de menos importante a más importante):

- Las indicadas por el navegador
- Hoja de estilos externa al HTML
- Hoja de estilos interna al HTML
- Estilo definido dentro de la propia etiqueta HTML

Otro concepto muy importante es el de validación. Si nuestra CSS cumple con los estándares establecidos por el W3C será un paso más para que nuestra web sea visible en todos los navegadores y sistemas operativos. Podemos validar una CSS si usamos el validador del W3C, disponible en:

<http://jigsaw.w3.org/css-validator/>

Creación de páginas web usando CSS

Una de las ventajas principales de la creación de páginas web empleando hojas de estilo es la posibilidad de separar en gran medida el texto de la presentación del mismo.

Dejando en el código HTML una estructura de capas mediante la aplicación de las etiquetas <DIV> y aplicando clases a dichas etiquetas podemos conseguir páginas web muy limpias y con una facilidad de modificación inmejorable.

La estructura de una página de inicio empleando este sistema sería la siguiente:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd" >
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en" >

<head>
  <title>Web de prueba</title>
  <link rel="stylesheet" href="style.css" type="text/css" media="screen">
</head>

<body>
<div class="header">cabecera</div>
<div class="container">
  <div class="menu">menu</div>
  <div class="contenido">
    Aqui va el contenido
  </div>
  <div class="cleaner">&nbsp;</div>
</div>
<div class="footer">datos personales</div>
</body>
</html>
```

Como se puede ver, se han creado varios bloques de contenido, asignando a cada uno de ellos una clase que definiremos a posteriori en la hoja de estilos. Podemos hacer dos comentarios adicionales:

- La definición del DOCTYPE es una etiqueta necesaria para la correcta interpretación del texto por parte de algunos navegadores.
- Es posible insertar bloques de contenido dentro de otros bloques de contenido (como podemos ver, la capa con la clase "container" tiene dentro dos capas con clases "menú" y "contenido").

Como puede verse, en este código HTML no se hace referencia alguna a la posición de las capas, ni a los diferentes tipos de letra que puedan tener. Tan solo será necesario especificar algunas características peculiares (algún texto en negrita, por ejemplo) porque el resto de características irán especificadas en la CSS, que podemos ver a continuación:

Contenidos del fichero “style.css”

```
/* Body: Cuerpo del texto, lo que se aplica si no hay ningún otro estilo que lo modifique */

body {
    background-color: #ddd;                /* Color de fondo */
    font-family: Vernada, Arial, Times;    /* Tipo de letra a usar */
    font-size: 11px;                      /* Tamaño de la letra */
    color: #666;                          /* Color de la letra */
    margin: 5px;                          /* Margen con respecto a la capa */
}

/* Header : Zona de contenido que se emplea como cabecera */
.header {
    margin: auto;                          /* Se emplea para que la capa se redimensione de forma automática */
    width: 740px;                          /* Tamaño de la capa → Escogido para 800x600 */
    background-color: #ccc;
    padding: 8px;                          /* Zona de margen interior de la capa, se aplica en todos los bordes */
    border: 2px solid #999;                /* Zona de margen externo de la capa, con un color aplicado */
    font-size: 15pt;
    font-weight: bold;
    text-align: right;                     /* Justificado del texto a la derecha */
                                          /* 740 + 8x2 + 2x2 = 760 = espacio dejado para el navegador */
}

/* Container : Contenedor que engloba al menú y a la zona de contenido central */
.container {
    margin: auto;
    width: 760px;
    background-color: #FFF;
    margin-top: 2px;                      /* el margen se puede aplicar solo a un borde */
}

/* Menu: Zona de contenido a la izquierda que contiene el menú de navegación */
.menu {
    background-color: #ccc;
    width: 150px;                         /* Un tamaño recomendable para un menu */
    float: left;                          /* Para que se apile a la izquierda */
}

/* Contenido : Zona de contenido principal */
.contenido {
    width: 590px;                         /* 590 + 150 + 10 + 10 = 760 .... Voilà */
    background-color: #eee;
    float: left;                          /* Se apila a la izda ... pero después de la OTRA */
    padding: 10px 10px 10px 10px;
    text-align: justify;
    color: #000;
}

/* Añadido especial para que el Firefox coopere */
.cleaner {
    clear: both;                          /* Esto impide que ninguna capa se acople a izda o dcha → pone orden */
    font-size: 1px;                      /* Firefox descarta las capas sin contenido, por eso ponemos un pixel */
}

/* Pie de pagina : Para informacion de contacto u otras cosas */
.footer {
    clear: both;
    width: 760px;
    margin: auto;
    background-color: #FFF;
    margin-top: 2px;
    text-align: center;
}
```

Como puede verse, la capacidad de generar y modificar el código es muy potente a la par que sencilla (puede consultarse toda la sintaxis completa de las CSS en <http://www.w3.org/TR/REC-CSS2>).

La elección de colores puede realizarse de una forma rudimentaria (lo profesional sería desde la paleta del Gimp, por ejemplo) escogiendo el código HTML para cada color desde la siguiente URL:

<http://www.visibone.com/colorlab/>

Trabajo a realizar durante la práctica

Se tiene que entregar como resultado de la práctica un boceto en HTML de lo que podría ser la “página de inicio” del proyecto de comercio electrónico. No tiene que ser en caso alguno una página web totalmente hecha y depurada, sino una aplicación práctica de todas las tecnologías vistas a lo largo de la práctica, en concreto:

- Ejemplos de todas las etiquetas HTML vistas en la práctica.
- Uso de una hoja de estilos.

Enlaces de interés

Tutoriales sobre HTML – CSS – JavaScript – PHP

<http://www.webestilo.com/>

<http://www.w3schools.com/default.asp>

Tutoriales sobre CSS

<http://www.w3schools.com/css/default.asp>

http://www.w3schools.com/css/css_examples.asp

Especificación de CSS2 del W3C

<http://www.w3.org/TR/REC-CSS2>

Herramienta de validación online de CSS del W3C

<http://jigsaw.w3.org/css-validator/>

W3Schools - Formularios

http://www.w3schools.com/html/html_examples.asp

Web Developer's Handbook

<http://www.alvit.de/handbook/>