MANTENIMIENTO DE PORTALES DE INFORMACIÓN

Pedro Muñoz Rodríguez

Vision Libros

MANTENIMIENTO DE PORTALES DE INFORMACIÓN

PEDRO MUÑOZ RODRÍGUEZ

MANTENIMIENTO DE PORTALES DE INFORMACIÓN

C.F.G.M Explotación de Sistemas Informáticos

Vision Libros

© Obra: Mantenimiento de Portales de Información

Primera edición: 2009

© Autor: Pedro Muñoz Rodríguez

ISBN: 978-84-9886-667-4

Diseño de cubierta: Marian González

© Editado por VISION LIBROS www.visionlibros.com

Gestión, promoción y distribución: Grupo Editor Vision Net S.L.
C./ San Ildefonso 17, local, 28012 Madrid. España.
Tlf: 0034 91 3117696 // Email: pedidos@visionnet.es
www.visionnet-libros.com

Disponible en librerías físicas y online.

Las opiniones expresadas en este trabajo son exclusivas del autor. No reflejan necesariamente las opiniones del editor, que queda eximido de cualquier responsabilidad derivada de las mismas.

Queda prohibida, salvo excepción prevista por la ley cualquier forma de reproducción, distribución, comunicación pública o transformación de esta obra sin contar con la autorización de los titulares de la propiedad intelectual. Diríjase a CEDRO (Centro Español de Derechos Reprográficos, www.cedro.es o por teléfono 917021970) si necesita fotocopiar, escanear o utilizar algún fragmento de esta obra. Gracias por comprar una edición autorizada de esta obra y por respetar las leyes del *copyright*.

ÍNDICE

1. INTRODUCCIÓN 13
2. DHTML 15
2.1. INTRODUCCIÓN 15
2.1.1. Introducción a las hojas de estilo 16
2.1.2. Introducción al posicionamiento de contenidos 16
2.1.3. Introducción a las fuentes descargables. 17
2.2. HOJAS DE ESTILO 17
2.2.1. Definición de estilos mediante hojas de estilo en cascada 17
2.2.2. Definición de Estilos mediante el Modelo de Objetos Documento de JavaScript 19
2.2.3. Ejemplos 20
2.2.4. Herencia de estilos 22
2.3. CREACIÓN Y USO DE HOJAS DE ESTILOS 24
2.3.1. Definición de Hojas de Estilo con la etiqueta <STYLE> 24
2.3.2. Definición de Hojas de Estilo en Ficheros Externos 26
2.3.3. Definición de clases de estilo 27
2.3.4. Definición de estilos individuales con nombre 30
2.3.5. Uso de criterios de selección contextual 31
2.3.6. Especificación de Estilos para Elementos Individuales 34
2.3.7. Combinando hojas de estilo 35
2.4. PROPIEDADES DE FORMATO DE ELMENTOS DE BLOQUE 37
2.4.1. Formato de bloques: Introducción y ejemplos 37
2.4.2. Márgenes 40
2.4.3. Bordes 41
2.4.4. Separadores 42
2.4.5. Herencia de Propiedades de Formato de Bloque 42
2.5. FUENTES DESCARGABLES 43
2.5.1. Cómo Usar Fuentes Descargables 44
2.5.2. Herramientas de Diseño 45
2.6. MANUAL DE REFERENCIA DE HOJAS DE ESTILO 45
2.6.1. Comentarios en las Hojas de estilo 45
2.6.2. Nuevas etiquetas HTML 45
<STYLE> 46
<LINK> 46
<SPAN> 47
2.6.3. Nuevos atributos para las etiquetas HTML 48
STYLE 48

CLASS 48
ID 49
2.6.4. Nuevas propiedades de objetos JavaScript *51*
TAGS 51
2.6.5. Propiedades de las Hojas de Estilo *52*
Fuentes 52
Propiedades del Texto 53
Propiedades de Formato de Elementos de Bloque 55
Propiedades de Color y Fondo 58
Propiedades de Clasificación 60
Unidades 61
3. COOKIES 63
3.1. FUNCIONAMIENTO DE LAS COOKIES 64
3.2. GUARDANDO COOKIES 66
3.3. USOS BIEN VISTOS DE LAS COOKIES 67
3.3.1. Personalización de las páginas *67*
3.3.2. Idioma *67*
3.3.3. Tiendas electrónicas *68*
3.3.4. Carritos de la compra *68*
3.3.5. Almacenamiento de login y password. *68*
3.4. USOS NO TAN BIEN VISTOS DE LAS COOKIES 68
3.4.1. Seguimiento de las visitas a un Web *68*
3.4.2. Carteles publicitarios *69*
3.4.3. Marketing personalizado *69*
3.4.4. Almacenamiento de login y password *69*
3.4.5. Almacenamiento de información sensible *70*
3.5. ¿CÓMO VER LAS COOKIES? 70
3.6. ¿CÓMO CONFIGURAR LAS COOKIES EN TU NAVEGADOR? 71
3.7. NAVEGACIÓN ANÓNIMA 72
3.8. RIESGOS 72
3.8.1. Riesgos reales *72*
3.8.2. Riesgos imaginarios *74*
3.9. ALTERNATIVAS A LAS COOKIES 75
4. SEGUIMIENTO DE SESIONES 77
4.1. INTRODUCCIÓN 77
4.2. CARACTERÍSTICAS DEL SEGUIMIENTO DE SESIONES EN ASP 78
4.2.1. Sintaxis *79*
4.2.2. Métodos más importantes *81*
4.2.3. Propiedades más importantes *82*
4.3. CARACTERÍSTICAS DEL SEGUIMIENTO DE SESIONES EN PHP 82
4.3.1. session_start() *83*

4.3.2. session_id() ... *83*
4.3.3. session_register() ... *83*
4.3.4. session_name() ... *84*
4.4. CARACTERÍSTICAS DEL SEGUIMIENTO DE SESIONES EN ... 85
4.4.1. Obtener una Sesión ... *86*
4.4.2. Almacenar y Obtener Datos desde la Sesión ... *87*
4.4.3. Invalidar la Sesión ... *89*
4.4.4. Manejar todos los Navegadores ... *90*
4.4.5. EJEMPLO COMPLETO ... *92*
5. JAVASCRIPT ... 97
5.1. INTRODUCCIÓN ... 97
5.2. COMENZANDO A PROGRAMAR CON JAVASCRIPT ... 98
5.2.1. Definición de variables ... *98*
5.2.2. Tipos de operadores ... *99*
5.2.3. Estructuras de Flujos De Control ... *100*
5.2.4. Funciones ... *101*
5.3. EVENTOS ... 102
5.3.1. Manejador de eventos: ... *103*
Evento onclick: ... 104
Evento onblur: ... 105
5.4. OBJETOS ... 106
5.4.1. Objeto document ... *106*
5.4.2. Objeto string ... *108*
5.4.3. Objeto Math ... *109*
5.4.4. Objeto Date ... *111*
5.4.5. Objeto Navigator ... *112*
5.4.6. Objeto History ... *113*
5.4.7. Objeto form ... *114*
5.4.8. Objeto window ... *116*
5.4.9. Objeto location ... *118*
5.4.10. Objeto link ... *119*
5.5. COOKIES ... 119
5.6. CAMBIO DINÁMICO DE IMAGENES ... 120
6. PHP 123
6.1. INTRODUCCIÓN ... 123
6.1.1. ¿Qué es PHP?. ... *123*
6.1.2. ¿Qué ventajas tiene PHP?. ... *124*
6.1.3. ¿Por qué PHP?. ... *125*
6.1.4. ¿Qué necesitamos? ... *126*
6.2. LA SINTAXIS ... 126
6.2.1. Variables. ... *127*

6.2.2. Arrays. ... *129*
6.2.3. Clases. ... *129*
6.2.4. Conversión de tipos. ... *130*
6.2.5. Cadenas de caracteres y expansión de comillas. ... *130*
6.2.6. Sentencias. ... *132*
6.2.7. Funciones. ... *132*
6.2.8. Variables globales. ... *133*
6.2.9. Estructuras de control. ... *133*
6.2.10. Inclusión de ficheros. ... *135*
6.3. TRABAJANDO CON PHP ... 136
6.3.1. ¿Cómo mostrar información? ... *136*
6.3.2. Formato de salida simple. ... *140*
6.3.3. Cómo leer datos: Formularios. ... *141*
6.3.4. Cómo pasar información entre programas. ... *141*
6.4. TRABAJANDO CON VARIABLES ... 142
6.4.1. Variables ... *142*
6.4.2. Asignación por valor o por referencia ... *143*
6.4.3. Información sobre el contenido de una variable ... *144*
6.4.4. Información sobre el tipo de variable ... *145*
6.4.5. Modificar variables ... *147*
6.4.6. Separar contenidos de variables ... *149*
6.4.7. Variables y constantes ... *150*
6.5. TRABAJANDO CON CADENAS DE CARACTERES ... 150
6.5.1. Adaptar cadenas al contexto ... *150*
6.5.2. Limpiar cadenas ... *153*
6.5.3. Mayúsculas y minúsculas ... *154*
6.5.4. Funciones criptográficas ... *155*
6.5.5. Información y modificación del contenido ... *156*
6.5.6. Búsqueda en cadenas ... *157*
6.5.7. Comparaciones ... *158*
6.5.8. Grupos de variables. ... *159*
6.6. TRABAJANDO CON ARRAYS ... 160
6.6.1. Arrays y cadenas de caracteres ... *160*
6.6.2. Array como pila ... *161*
6.6.3. Ordenación de arrays ... *162*
6.6.4. Modificación de arrays ... *164*
6.6.5. Extraer información de un array ... *166*
6.7. TRABAJANDO CON FECHAS ... 168
6.7.1. Funciones para fechas ... *168*
6.7.2. Leer una fecha ... *173*

6.7.3. Obtener la fecha 173
6.8. SISTEMAS DE FICHEROS (DIRECTORIOS Y FICHEROS). 174
6.9. PERMISOS Y PROPIETARIOS 174
6.9.1. Funciones informativas 176
6.9.2. Operaciones con ficheros 179
6.9.3. Operaciones en ficheros 181
6.9.4. Operaciones con directorios 183
6.10. CONTROL DE EJECUCIÓN Y USUARIOS 185
6.10.1. Ejecutando otros programas 185
6.10.2. Validando usuraos 186
6.10.3. Propiedades de los documentos HTTP,HTML, URL 187
6.11. FUNCIONES ÚTILES 190
6.11.1. Variables de entorno y funciones informativas 190
6.11.2. Funciones matemáticas 192
6.11.3. Expresiones regulares en PHP 195
6.12. OPERACIONES AVANZADAS 196
6.12.1. Transferencias de ficheros y FTP 196
6.12.2. Correo electrónico 199
6.12.3. Generación de contenidos dinámicos 201
6.12.4. Operaciones de red y socket 203
6.13. ACCESO A BASES DE DATOS 204
6.14. COMO … 205
6.14.1. Variables 205
6.14.2. Información y errores 207
6.14.3. Gráficos 208
6.14.4. Ejecución 209
6.14.5. Formularios 209
6.14.6. Solucionar errores frecuentes 211

1. INTRODUCCIÓN

La asignatura de mantenimiento de portales de información se imparte en el segundo curso del ciclo formativo de grado medio de explotación de sistemas informáticos (E.S.I.). En esta asignatura se dan las bases para que el futuro técnico pueda gestionar un sistema de información basado en internet. Entre las labores que deberá poder desempeñar están:

- Mantener la explotación de un sitio Web dentro de los parámetros de rendimiento y seguridad prefijados.

- Actualizar contenidos de servidores Web locales o remotos, mediante el uso de clientes FTP o herramientas de gestión de contenidos.

- Elaborar páginas Web con lenguajes de marcas, mediante herramientas editoras de textos o específicas de desarrollo Web, incluyendo scripts de navegador, y realizando la verificación de su funcionamiento.

- Elaborar páginas Web mediante lenguajes script de servidor.

2. DHTML

2.1. INTRODUCCIÓN

Con HTML los documentos que se generan son estáticos e inanimados, es decir una vez que el navegador ha cargado la página esta no sufre ningún tipo de modificación, es decir permanece fija, no pudiendo ser modificado ni en forma ni en contenido.

DHTML ofrece a los creadores de páginas web la posibilidad de modificar, cambiar la apariencia, ocultar, mostrar y animar el contenido dinámicamente.

Los tres componentes del HTML Dinámico son:

- Hojas de Estilo
- Posicionamiento de Contenidos
- Fuentes Descargables

Usando conjuntamente, estos componentes ofrecen un mayor control sobre la apariencia, manejo y comportamiento de las páginas web. Las hojas de estilo permiten especificar atributos estilísticos de los elementos tipográficos de las páginas.

Con el posicionamiento de contenidos, estás seguro de que cada componente será mostrado dentro de la página, exactamente donde deseas que aparezca, pudiendo modificar dinámicamente (después de cargada la página y mientras se visualiza) tanto la apariencia como la localización.

Usando fuentes descargables, podemos estar seguros de que la página se vera con el aspecto (en cuanto a fuentes) con que se diseñó, ya que se especifica y se incluye la fuente de letra en el documento HTML. Así, el navegador que utilice el usuario final, utilizará la fuente con la que nosotros diseñamos la página, y no intentará hacerlo con la más aproximada que tenga definida

2.1.1. Introducción a las hojas de estilo

Antes de la introducción de las hojas de estilo, los creadores de páginas web sólo tenían un control parcial sobre el aspecto final de sus páginas. Por ejemplo, se podía especificar que cierto texto debía verse como una cabecera, pero no se podían colocar márgenes a una página ni escoger un borde decorado para un texto.

Las hojas de estilo nos permiten un mayor control sobre el aspecto de nuestros documentos. Con ellas podemos especificar muchos atributos tales como colores, márgenes, alineación de elementos, tipos y tamaños de letras, y muchos más. Podemos utilizar bordes para hacer que ciertos elementos resalten del resto de un documento. Podemos especificar que se utilicen diferentes fuentes para diferentes elementos tales como párrafos o cabeceras.
Además podemos emplear hojas de estilo como patrones o páginas maestras de forma que múltiples páginas puedan tener el mismo aspecto.
Las hojas de estilo pueden crearse empleando dos tipos de sintaxis, CSS (Cascade Style Sheets) y JavaScript.

2.1.2. Introducción al posicionamiento de contenidos

Ya no se está restringido al posicionamiento secuencial de los elementos sobre la página. Especificando la posición de los bloques podemos decidir donde se mostrara cada elemento en vez de dejar esta tarea al arbitrio del navegador. Por ejemplo, podemos colocar un bloque en la esquina superior izquierda, y otro en la inferior derecha. También podemos hacer que varios bloques compartan espacio, de forma que se puedan solapar. Ahora se puede decidir la posición exacta de los elementos.

Utilizando JavaScript podemos cambiar el aspecto de la página dinámicamente. Podemos hacer que los elementos aparezcan o se desvanezcan, podemos cambiar su color y su posición, etc. Podemos añadir animaciones

dentro de nuestras páginas moviendo y modificando ciertas partes de la misma.

El uso conjunto las hojas de estilo y el posicionamiento de contenidos nos permite crear páginas web que utilicen diferentes estilos en diferentes partes de la página.

Para el posicionamiento de contenidos también podemos utilizar dos tipos de sintaxis: CSS y JavaScript.

2.1.3. Introducción a las fuentes descargables.

Utilizando fuentes descargables podemos adjuntar fuentes con nuestras páginas web. De esta forma se garantiza que las página siempre serán mostradas con la fuente que deseemos. Ya no es necesario emplear las fuentes genéricas para conseguir que nuestras páginas tengan aproximadamente el mismo aspecto en todas las plataformas. Tampoco estaremos restringidos por las características propias de las fuentes en cada plataforma.

Para proteger los derechos de autor de los diseñadores de fuentes, estas están protegidas de forma que es imposible que el usuario las copie y las pueda usar de nuevo. Así podemos incluir fuentes en nuestras páginas sin tener que preocuparnos de que los usuarios las puedan copiar.

2.2. HOJAS DE ESTILO

En esta sección se introduce el uso de las hojas de estilo. Se da un breve resumen de los diferentes tipos de sintaxis que se pueden emplear para definir estilos, se muestran algunos ejemplos y se explica la herencia de propiedades.

2.2.1. Definición de estilos mediante hojas de estilo en cascada

Una hoja de estilo consiste en una o más definiciones de estilo. En sintaxis CSS, los nombres de las propiedades y los valores se encierran entre llaves {}.
El criterio de selección determina a que elementos se aplica, o es aplicable, el estilo. Si el criterio de selección es un elemento HTML, el estilo es aplicado a todos las instancias de dicho elemento. El criterio de selección también puede ser una clase, un ID o contextual. Cada una de estos criterios de selección se verá a continuación.

En una definición de estilo cada propiedad es seguida por dos puntos y el valor de dicha propiedad. Cada par propiedad/valor está separado del siguiente por un punto y coma (;).

Por ejemplo, la siguiente hoja de estilo en cascada contiene dos definiciones de estilos. El primero especifica que todos los párrafos, <p>, se verán en negrita y en color blanco. El segundo hará que todas las cabeceras, <h1>, aparezcan centradas.

```
<style type="text/css">
<!--
    p {font-weight: bold; color: white;}
    h1 {align: center;}
-->
</style>
```

La definición de estilos se puede encerrar entre comentarios (<!-- ... !-->), de esta forma los navegadores que no reconozcan la etiqueta <style> la ignoraran.

Es importante no incluir dobles comillas en la especificación de valores de las CSS.

CSS exige un estricto cumplimiento de sus normas de sintaxis. Asegúrate de no omitir ningún punto y coma entre los pares nombre/valor. Si lo haces se ignorara por completo la definición de estilo. De igual forma, si accidentalmente se añade un símbolo extraño la definición será ignorada.

2.2.2. Definición de Estilos mediante el Modelo de Objetos Documento de JavaScript

Mediante JavaScript también es posible definir hojas de estilo utilizando el modelo de objetos documento. Este modelo ve una página web como un objeto con propiedades que pueden ser consultadas y modificadas. En el siguiente ejemplo se cambia el valor de color de la etiqueta h1 de la propiedad tags del objeto document:

```
document.tags.h1.color = "red"
```

La propiedad tags siempre se aplica al objeto document del documento actual, por eso podemos omitir document en la expresión documentaos.

El siguiente ejemplo usa JavaScript para crear una hoja de estilo que contiene dos definiciones de estilo. El primero especifica que todos los párrafos <p> se verán en negrita y en color blanco. El segundo hará que todas las cabeceras <h1> aparezcan centradas

```
<style type="text/javascript">
    tags.p.weight = "bold";
    tags.p.color = "white";
    tags.h1.align = "center";
</style>
```

No se pueden encerrar entre comentarios (<!-- ... -->) los contenidos de una hoja de estilo creada mediante JavaScript.

También se puede utilizar with para abreviar la definición de estilos para elementos sobre los que hay que modificar varias propiedades. Este ejemplo especifica que todas las etiquetas <p> serán verdes, negritas, cursivas y usaran la fuente helvética.

```
with (tags.p)
{
    color = "green";
    fontWeight = "bold";
    fontStyle = "italic";
    fontFamily = "helvetica";
}
```

2.2.3. Ejemplos

Usando hojas de estilo se pueden especificar muchos atributos de estilo. Algunas de estas características son el alineamiento, indentación y color del texto, la familia, peso y estilo de las fuentes,... Podemos seleccionar una imagen o un color de fondo para cualquier elemento. También podemos escoger el color y estilo de las marcas de lista.

Se pueden poner márgenes y especificar bordes para elementos de bloque, escoger el relleno de los elementos que tienen bordes, indicar la distancia entre el borde y el contenido del elemento.

El siguiente ejemplo muestra una hoja de estilo sencilla descrita mediante sintaxis CSS y JavaScript. En ella se especifica que todos los elementos <p> tendrán márgenes derecho e izquierdo y que su texto estará centrado. Todos los elementos <h4> estarán subrayados y en verde. Todos los elementos <h5> se mostraran en mayúsculas, tendrán un borde con aspecto 3D sobresaliente de 4 puntos de ancho y color rojo. Su texto será rojo y el fondo amarillo. Finalmente, los <blockquote> serán azules y en cursiva, su altura de línea (interlinea) será un 150% mayor que por defecto y su primera línea estará indentado un 10%.

Sintaxis CSS

```
<style type="text/css">
    p {text-align: center; margin-left: 20%; margin-
right: 20%;}
    h4 {text-decoration: underline; color: green;}
    h5
    {
        text-transform: uppercase;color:red;
        border-width: 4pt; border-style: outset;
        background-color: yellow; padding: 4pt;
        border-color: red;}
    }
    blockquote
    {
        color: blue; font-style: italic;
        line-height: 1.5; text-indent: 10%
    }
</style>
```

Sintaxis JavaScript

```
<style type="text/javascript">
    with (tags.p)
    {
        textAlign = "center"; marginLeft = "20%";
marginRight = "20%";
    }
    with (tags.h4) {textDecoration = "underline";
color = "green";}
    with (tags.h5)
    {
        textTransform = "uppercase"; color = "red";
        borderWidth = "4pt"; borderStyle = "outset";
        backgroundColor = "yellow"; paddings("4pt");
        borderColor = "red";
    }
    with (tags.blockquote)
    {
        color = "blue"; fontStyle = "italic";
        lineHeight = "1.5"; textIndent = "20pt";
    }
</style>
```