

# Elaboración Básica de Páginas Web

Introducción a HTML5, CSS y JavaScript



**Fernando Jiménez Ávila**



Material Didáctico de Apoyo para el Curso Básico

Elaboración Básica de Página Web.

© Fernando Jiménez Ávila.

Impreso y hecho en Nezahualcóyotl, Estado de México, Abril 2024.



Contenido

Introducción.....	7
Conceptos básicos de HTML.....	7
Reglas del HTML a tener en cuenta.....	7
Estructuración de un documento.....	8
El DOCTYPE.....	8
La cabecera.....	8
El cuerpo del documento.....	8
Comentarios.....	9
Estructuración del texto.....	10
Párrafos y saltos de línea.....	10
Cabeceras y separadores.....	12
Tratamiento del texto.....	13
Preformatos.....	15
Citas.....	16
Estilos de texto.....	18
Acrónimos y abreviaturas.....	19
Las fuentes.....	20
Los enlaces.....	23
Estructura de los enlaces.....	24
Tipos de enlaces.....	24
Listas.....	27
Listas desordenadas.....	27
Listas ordenadas.....	28
Listas de definición.....	29
Imágenes.....	30
Alineación de imágenes con texto.....	31
Imágenes mapeadas.....	32
Desarrollo de la imagen.....	32
Programación del código.....	32
Tablas.....	33
Estructuración de una tabla.....	33
Alineación y dimensión de las celdas.....	34
Otros atributos.....	36
Reproducción de audio.....	37
Música de fondo.....	37
Los frames.....	38



Visión y Pasión por Emprender y Servir

Los enlaces y el atributo target.....	38
Estructura básica.....	38
Las etiquetas <frame> y <noframe>.....	39
Los formularios.....	40
Declaración del formulario <form>.....	40
Definición de campos de entrada <input>.....	40
Text.....	41
Password.....	41
Hidden.....	41
Checkbox.....	42
Radio.....	42
Definición de campos de selección <select>.....	42
Área de texto <textarea>.....	43
Reset.....	43
Submit.....	43
Introducción a las Hojas de Estilo en Cascada (CSS).....	43
Para qué sirven las Hojas de Estilo en Cascada.....	43
Pros y contras de las Hojas de Estilo en Cascada.....	44
Cascading Style Sheets (CSS).....	44
Primeros pasos.....	45
Estilos de etiqueta.....	46
Definición de estilos mediante clases.....	47
Definición de estilos con id.....	49
La regla @import.....	50
Las etiquetas <div> y <span>.....	50
Comentarios.....	52
Unidades de medida y color.....	52
Propiedades de las Hojas de Estilo en Cascada.....	53
Propiedades de las fuentes.....	53
Propiedades del fondo y del color.....	55
Propiedades de las ventanas.....	55
Propiedades de clasificación.....	57
Ejemplos de aplicación.....	58
Introducción a JavaScript.....	60
JavaScript y HTML.....	60
Introducir comentarios.....	62
Corchetes.....	63

Visión y Pasión por Emprender y Servir

El punto y coma.....	63
Archivos fuente.....	63
Tratamientos de variables.....	64
Las variables.....	64
Tipos de variables.....	65
Variables de cadena.....	65
Variables numéricas.....	66
Enteros.....	67
Coma flotante.....	67
Variables booleanas.....	67
Variables undefined.....	67
Conversión de tipos de datos.....	68
Constantes.....	69
Localización de las variables.....	69
Palabras reservadas.....	70
Los operadores.....	70
Tipos de operadores.....	70
Operadores aritméticos.....	71
Operadores lógicos.....	72
Operadores de comparación.....	73
Operadores condicionales.....	74
Operadores de asignación.....	74
Estructuras condicionales.....	74
Estructuras condicionales.....	74
La sentencia if.....	74
Estructuras if ... else.....	75
Estructuras if anidadas.....	76
Los bucles for.....	78
Bucles for ... in.....	79
Los bucles while.....	81
Los bucles do ... while.....	82
La instrucción break.....	83
La instrucción continue.....	83
La sentencia switch.....	84
Ventanas de confirmación.....	85
Ejemplo de carrito de compras.....	86
El catálogo.....	87



La hoja de pedido.....	91
------------------------	----



Este material didáctico está dirigido a aquellas personas que deseen conocer los fundamentos básicos para diseñar páginas web utilizando HTML, CSS y JavaScript. Este material didáctico se diseñó como un texto práctico que explica mediante ejemplos sencillos los distintos elementos del HTML, de CSS y de JavaScript.

Este material didáctico está enfocado para que, de forma gradual, el lector adquiera los conocimientos básicos de HTML necesarios para poder diseñar una página web, manejar los estilos de CSS e insertar las rutinas de JavaScript.

Para aprovechar este material didáctico al máximo se recomienda seguir los temas por el orden indicado y realizar los ejemplos en computadora.

Probar los ejemplos, modificarlos y ver cómo funciona el programa, es una excelente forma de aprender.

Para la visualización de los diferentes ejemplos se recomienda el uso de un navegador web reciente.

Los ejemplos pueden editarse con cualquier editor de textos, como el Bloc de Notas de Windows, aunque siempre es mejor utilizar un editor especializado como Notepad++, para el desarrollo de los ejemplos.

### Conceptos básicos de HTML

HTML (Hiper Text Markup Language) es el lenguaje de marcado de hipertexto que se creó para el desarrollo de páginas web y que contiene básicamente las funcionalidades necesarias para la visualización de páginas web cumpliendo las especificaciones más estrictas.

El objetivo es lograr páginas web donde la información y la forma de presentarla, o renderizarla, estén claramente definidas y separadas. HTML es, por lo tanto, un lenguaje semántico, que quiere decir que no se define el aspecto de las cosas sino lo que significan.

Esta característica se aprecia claramente en la estructuración semántica, ya que HTML sirve únicamente para transmitir información que contiene un documento web, dejando el aspecto y el diseño para las hojas de estilo en cascada (CSS) y la interactividad y funcionalidad para JavaScript.

Como ya se ha comentado, la diferencia entre los distintos elementos, HTML, CSS y JavaScript, es principalmente un cambio en el concepto y la forma de la estructuración del documento web. Se busca una sintaxis coherente dentro del documento web donde los distintos elementos deben estar correctamente estructurados y anidados, todas las etiquetas en minúsculas, los elementos cerrados correctamente, los valores de los atributos entrecomillados, etc. De esta forma se evita que exista un desorden existente en muchos códigos web.

HTML se basa en el uso de etiquetas, también llamadas marcas, directivas o comandos (tags). Estas etiquetas son textos delimitados por los símbolos menor que (<) y mayor que (>). Básicamente estas etiquetas indican al navegador la forma de representar los elementos (texto, imágenes, etc.) que contiene el documento HTML.

Existe normalmente una etiqueta de inicio <etiqueta> y una etiqueta de final o de cierre </etiqueta>, aunque algunas etiquetas sólo disponen de la de inicio, como es el caso del salto de línea <br>.

La mayoría de las etiquetas se complementan con los llamados atributos, elementos que permiten establecer valores o propiedades de la misma

### Reglas del HTML a tener en cuenta

- 1.- Un documento HTML no puede tener etiquetas abiertas, por ejemplo, un <p> debe cerrarse con un </p>, un <li> debe cerrarse con </li>, y así, lo mismo con la mayoría de las etiquetas,
- 2.- Las etiquetas en un documento HTML deben estar anidados de forma lógica, por lo tanto, deben cerrarse en el orden inverso que fueron abiertos. Por ejemplo, <p><strong> debe cerrarse en el orden </strong></p>.
- 3.- Todas las etiquetas y sus atributos deben escribirse siempre en minúsculas. No ocurre lo mismo con el valor de los atributos que sí pueden contener mayúsculas.
- 4.- Todos los valores de los atributos de las etiquetas deben estar siempre entrecomillados.
- 5.- Se debe recordar siempre que el HTML es un lenguaje semántico, por lo tanto, cuando se desarrolla un documento HTML hay que pensar en la organización y estructuración del mismo más que en su maquetación o formateo.

**Nota:** Las etiquetas mal escritas o desconocidas para un navegador web son ignoradas. Sin embargo, los errores de sintaxis, como, por ejemplo, olvidar delimitar una etiqueta, pueden provocar errores en la validación y visualización del documento web.

A continuación, se analizarán los elementos que componen la estructura básica de un documento web.

#### El DOCTYPE

Para escribir un código HTML válido se debe indicar al navegador el tipo de documento que se va a abrir. La declaración del tipo de documento HTML es obligatoria.

Para eso, se utiliza:

```
<!DOCTYPE html>
```

El resto del documento HTML está definido por una etiqueta de inicio se utiliza la etiqueta de inicio `<html>` y una de terminación `</html>`. Otras dos partes fundamentales de todo documento web son la cabecera, comprendida entre las etiquetas `<head>` y `</head>` y el cuerpo del documento web, entre `<body>` y `</body>`.

La estructura básica de un documento web es como sigue:

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
</body>
</html>
```

#### La cabecera

La cabecera de un documento web viene delimitada por las etiquetas `<head>` y `</head>` respectivamente, y contiene fundamentalmente información sobre el documento HTML como el título, vínculos a hojas de estilo en cascada (CSS), o scripts de JavaScript, instrucciones meta, etc. Todas ellas son opcionales, aunque el uso de la etiqueta `<title>` es más que recomendado.

La etiqueta `<title>` es la que se encarga de especificar el título del documento web en la barra de título del navegador. Éste suele mostrarse generalmente en la parte superior del navegador. También es utilizado como identificador en las listas de favoritos, también llamadas bookmarks o marcadores.

Ejemplo:

```
<title>Este es el título de mi primera página web</title>
```

La siguiente etiqueta relacionada con la cabecera de un documento es `<meta>`, usada principalmente, por ejemplo, para sustituir el documento actual por otro en un determinado número de segundos o bien para incluir elementos informativos como el nombre del autor, la fecha de expiración del documento web o incluso el programa con el que se diseñó la página. En todos los casos, la información se escribe usando pares nombre/valor.

Se emplean atributos del tipo `http-equiv`, y `name`, en ambos casos, el valor a especificar se determina con el campo `content`.

Por ejemplo:

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
```

#### El cuerpo del documento

El cuerpo de un documento HTML contiene el texto de la página web que se va a presentar al usuario. Las etiquetas de inicio y cierre son `<body>` y `</body>` respectivamente, y entre ellas, se debe situar todo el código fuente restante del documento web.



### Visión y Pasión por Emprender y Servir

Los atributos de la etiqueta <body> permiten definir el fondo de la pantalla y los colores del texto, incluso, una imagen de fondo.

Los colores se pueden indicar de dos maneras: en formato hexadecimal o referenciados por su nombre preestablecido.

Por ejemplo, si se quiere indicar colores en formato hexadecimal, se deberá indicar la proporción de rojo, verde y azul correspondiente para formar el color deseado. Así pues, el valor RGB del color vendrá determinado por RR, GG y BB, para los valores de Rojo, Verde y Azul, con valores entre 00 y FF (de 0 a 255), así, por ejemplo, “#000000” correspondería al color negro, “#FF0000” al rojo, “#00FF00” al verde, “#0000FF” al azul, “#FFFFFF” al blanco y así hasta completar todas las posibilidades.

También se puede indicar el nombre del color en inglés, como black, teal, blue, navy, lime, white, purple, yellow, olive, red, marron, gray, fuchsia, green, silver, aqua, etc.

Así, los atributos válidos para la etiqueta <body> son los siguientes:

- bgcolor="#rrggbb o nombre del color", para indicar el color del fondo del documento web.
- text="#rrggbb o nombre del color", para especificar el color general del texto.
- link="#rrggbb o nombre del color", para especificar el color del texto de los enlaces.
- vlink="#rrggbb o nombre del color", para indicar el color del texto de los enlaces ya visitados.
- alink="#rrggbb o nombre del color", para indicar el color en que se pondrá el texto del enlace en el momento de su activación.
- background="nombre del archivo de la imagen", para indicar la imagen de fondo.

### Comentarios

Para incluir comentarios dentro del código fuente HTML, se puede utilizar la etiqueta <!-- comentarios -->. Su utilización es más que recomendable en documentos web extensos o complejos, ya que permite introducir anotaciones para revisiones posteriores del código fuente, por ejemplo:

```
<!--Esto es un comentario, que permitirá documentar el código fuente -->
```

Se muestra a continuación un código fuente de HTML que se puede utilizar como plantilla del documento HTML para los ejercicios:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1>Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

HTML no reconoce más de un espacio en blanco separando cualquier elemento o texto, es decir, los saltos de línea, tabulaciones, más espacios y demás separadores son convertidos a un único espacio en blanco, el navegador web los ignora.

Otro punto importante es que no se debe, aunque se puede, escribir directamente caracteres especiales, tales como comillas, eñes, acentos o símbolos. Para evitar problemas se deben introducir secuencias de escape propias de HTML, conocidas como entidades HTML, algunas de las más comunes:

Sustituir < por &lt;  
Sustituir > por &gt;  
Sustituir & por &amp;  
Sustituir “ por &quot;  
Sustituir ‘ por &apos;  
Sustituir á por &aacute;  
Sustituir é por &eacute;  
Sustituir í por &iacute;  
Sustituir ó por &oacute;  
Sustituir ú por &uacute;  
Sustituir ñ por &ntilde;  
Sustituir Á por &Aacute;  
Sustituir É por &Eacute;  
Sustituir Í por &Iacute;  
Sustituir Ó por &Oacute;  
Sustituir Ú por &Uacute;  
Sustituir Ñ por &Ntilde;

Por ejemplo:

```
<p> Esto está mal escrito</p>  
<p> Esto est&aacute; bien escrito</p>
```

### Párrafos y saltos de línea

Como para HTML todo el texto es continuo, se necesita de algún mecanismo para indicar el principio y el final de un párrafo. La etiqueta <p> y </p> es la encargada de separar párrafos de texto.

Otra etiqueta asociada a la anterior es <br>. Esta etiqueta sólo tiene marca inicial y se usa para cambiar o saltar de línea.

Por ejemplo:

```
<!DOCTYPE html>  
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>  
<meta http-equiv="Content-Language" content="ES-ES"/>  
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">  
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza, Nezahualcóyotl"/>  
<meta name="author" content="Fernando Jiménez Ávila"/>  
<meta name="distribution" content="Global"/>  
<meta name="copyright" content="© Fernando Jiménez Ávila"/>  
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>  
</head>  
<body>  
<header>  
<h1>CCAP México</h1>  
<h1>Bienvenido al Mundo del HTML5</h1>  
<h2>¡No te asustes, no es tan difícil!</h2>  
</header>
```



<!-- Contenido del documento -->

<h1>Uso de párrafo y salto de línea</h1>

Cuando te nombran,  
me roban un poquito de tu nombre;  
parece mentira,  
que media docena de letras digan tanto.

Mi locura seria deshacer las murallas con tu nombre,  
iría pintando todas las paredes,  
no quedaría un pozo  
sin que yo asomara  
para decir tu nombre,  
ni montaña de piedra  
donde yo no gritara  
enseñándole al eco  
tus seis letras distintas.

<p>Cuando te nombran,  
me roban un poquito de tu nombre;  
parece mentira,  
que media docena de letras digan tanto.</p>

<p>Mi locura seria deshacer las murallas con tu nombre,  
iría pintando todas las paredes,  
no quedaría un pozo  
sin que yo asomara  
para decir tu nombre,  
ni montaña de piedra  
donde yo no gritara  
enseñándole al eco  
tus seis letras distintas.</p>

<p>Cuando te nombran,<br>  
me roban un poquito de tu nombre;<br>  
parece mentira,<br>  
que media docena de letras digan tanto.</p>

<p>Mi locura seria deshacer las murallas con tu nombre,<br>  
iría pintando todas las paredes,<br>  
no quedaría un pozo<br>  
sin que yo asomara<br>  
para decir tu nombre,<br>  
ni montaña de piedra<br>  
donde yo no gritara<br>  
enseñándole al eco<br>  
tus seis letras distintas.</p>

<footer>  
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>  
</footer>  
</body>  
</html>

## Cabeceras y separadores

**Visión y Pasión por Emprender y Servir**

Las cabeceras se emplean para dividir los documentos web en secciones, o más concretamente, para marcar los títulos de estas secciones. La etiqueta que define la cabecera es `<h1>` y `</h1>`, donde n varía de 1 a 6. El valor de 1 se correspondería con una letra muy grande y el de 6 con una muy pequeña. La utilización de esta etiqueta genera automáticamente un salto de línea similar al conseguido con la utilización de la etiqueta `<p>`. Por ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza, Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de cabeceras</h1>

<h1>CSS. Hojas de estilo en cascada para webmasters</h1>
<h2>Capítulo X</h2>
<p>Las hojas de estilo en cascada, permiten controlar totalmente el ...</p>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

Para poder separar partes del documento web, la etiqueta `<hr>` dibuja una barra horizontal que divide la pantalla. El atributo `size` permite especificar el grosor de la barra de división, mientras que `width` determina la longitud o el porcentaje relativo con respecto al ancho de la página. La alineación de la barra de división se ajusta con el atributo `align` con los valores `left`, `right` o `center`.

Por ejemplo:

```
<hr align="center" size="10" width="50">
```

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza, Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
```



```
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de cabeceras y de barra horizontal</h1>

<h1>CSS. Hojas de estilo en cascada para webmasters</h1>
<h2>Capítulo X</h2>
<p>Las hojas de estilo en cascada, permiten controlar totalmente el ...</p>

<hr align="center" size="10" width="50">

<h1> Cabeceras y separadores</h1>
<h1>Cabecera de primer nivel</h1>
<h2>Cabecera de segundo nivel</h2>
<h3>Cabecera de tercer nivel</h3>
<hr>
<h4>Cabecera de cuarto nivel</h4>
<h5>Cabecera de quinto nivel</h5>
<h6>Cabecera de sexto nivel</h6>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

### Tratamiento del texto

Para facilitar la estructura visual de un documento HTML, se destacan algunas partes del mismo con respecto al resto. Como símil, se puede pensar en la opción de negritas en un procesador de textos, para destacar una palabra en el documento, pero con la diferencia de que HTML es un lenguaje semántico, es decir, se le indica al navegador qué parte del documento HTML se enfatiza o se destaca, pudiendo, mediante hojas de estilo en cascada (CSS), definir el aspecto de ese énfasis.

### Énfasis

Para dar énfasis a un texto, se dispone de las etiquetas `<em>` y `</em>`, que suele representarse en cursiva en los navegadores y para aplicar más énfasis se utilizan las etiquetas `<strong>` y `</strong>`, representado como negrita.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza, Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
```



```
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de énfasis</h1>

<p> Soy un texto normal que <em>ahora quiero destacar</em>,<br>
Pero si me enfado, <strong>resalto mucho</strong> para destacar</p>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

También se pueden utilizar las etiquetas `<cite>` y `</cite>` para poner una cita en un fragmento de texto. La representación será en cursiva de igual forma que la etiqueta `<em>`.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza, Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de énfasis y cita</h1>

<p> Soy un texto normal que <em>ahora quiero destacar</em>,<br>
Pero si me enfado, <strong>resalto mucho</strong> para destacar</p>
<p>Por ejemplo, la siguiente cita <cite>El respeto al derecho ajeno, es la paz</cite> es de Benito Pablo Juárez García.</p>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```



## Preformatos

HTML ignora los espacios en blanco consecutivos, tabulaciones y saltos de línea, entre otras características. Por lo tanto, para poder mostrar los espacio y tabulaciones y demás del código fuente de un documento web, se usa la etiqueta <pre> y </pre>.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza, Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>
```

<!-- Contenido del documento -->

<h1>Uso de preformato</h1>

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        KeyAscii = 0
        SendKeys "{tab}"
    ElseIf KeyAscii <> 8 Then
        If Not IsNumeric(Chr(KeyAscii)) Then
            Beep
            keyAscii = 0
        End If
    End If
End Sub
```

```
<pre>
Private Sub Text1_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        KeyAscii = 0
        SendKeys "{tab}"
    ElseIf KeyAscii <> 8 Then
        If Not IsNumeric(Chr(KeyAscii)) Then
            Beep
            keyAscii = 0
        End If
    End If
End Sub
</pre>
```

```
<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
```



</body>  
</html>

## Citas

HTML dispone de tres etiquetas para poder definir citas: <blockquote>, <q> y <cite>.

La etiqueta <blockquote> se utiliza para mostrar las citas de párrafos largos. Generalmente los navegadores lo presentan como un sangrado. Las etiquetas que definen la cita son <blockquote> y </blockquote>.

Por ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de citas</h1>

<blockquote>
<p>
Ejemplo de hormigas en hilera<br>
va trepando por el y en sus entrañas<br>
urden sus telas grises las arañas.
</p>
</blockquote>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

Para escribir citas cortas en una línea de un párrafo, se utilizan las etiquetas <q> y </q>, el texto introducido dentro de estas etiquetas se mostrará entrecomillado en la mayoría de los navegadores.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
```





Visión y Pasión por Emprender y Servir

```
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de citas</h1>

<p>Yo soy un texto normal, <q>dijo el párrafo</q></p>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

Para definir una cita refiriéndose al autor de la misma, se utilizan las etiquetas <cite> y </cite>

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de citas</h1>

<p>El libro de HTML, CSS y JavaScript es de la <cite>Editorial CCAP México.</cite></p>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
```

</html>

#### Estilos de texto

Se muestran a continuación otras etiquetas para definir estilos de texto:

- `<b> </b>` se utiliza para mostrar texto en negritas.
- `<i> </i>` para poner texto en cursiva.
- `<tt> </tt>` el texto tendrá un tamaño menor y los caracteres serán similares a los de una máquina de escribir (teletipo).
- `<big> </big>` texto grande, el mayor tamaño de fuente.
- `<small> </small>` texto pequeño, el menor tamaño de fuente.
- `<sup> </sup>` superíndice.
- `<sub> </sub>` subíndice.
- `<address> </address>` se emplea para mostrar un bloque de texto en cursiva, se utiliza generalmente para marcar una dirección o firma.
- `<code> </code>` se utiliza para representar una etiqueta HTML o código de programa.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1>Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de otros estilos de texto</h1>

<p>Los lenguajes de programación de alto nivel más comunes son: <b>Fortran, Algol, Ada, Logo, Progress, Pilot, Forth, <br>
Modula2, Cobol, Lisp, JavaScript, PHP, Perl, Python, Snobol, BASIC, PL/1, APL, APT, C, C++, Pascal, DBase, Clipper, TypeScript,
SQL,</b><br>
entre otros; y los llamados de <i>cuarta generación</i>: <tt>Visual Basic, Visual C, Delphi,</tt> etc.</p>

<p>El Pseudocódigo es <big>un conjunto de oraciones sencillas</big> que se utilizan para representar y escribir un algoritmo, <br>
tal como se aprecia en el ejemplo anterior.

El <small>Pseudocódigo</small> es una forma de representar y escribir algoritmos. <br>
El Pseudocódigo es una <sup>herramienta muy poderosa</sup> para los programadores profesionales y, en particular, <br>
para los que <sub>no tengan experiencia</sub> en el diseño de programas.

<address>
Calle La Cucaracha #2585,<br>
Colonia Benito Juárez,<br>
Código Postal 57000,<br>
Nezahualcóyotl, Estado de México
</address>
```

<br>

<code>

<pre>

Algoritmo OperacionesBasicascon2Numeros

```
Definir Numero1, Numero2 Como Entero;  
Definir Suma, Resta, Producto, Division Como Entero;
```

```
Escribir "Introduce el primer número entero: ";  
Leer Numero1;  
Escribir "Introduce el segundo número entero: ";  
Leer Numero2;
```

```
Escribir "Los números enteros introducidos son: ", Numero1, " y ", Numero2;
```

```
Suma <- Numero1 + Numero2;  
Resta <- Numero1 - Numero2;  
Producto <- Numero1 * Numero2;  
Division <- Numero1 / Numero2;
```

```
Escribir "La suma de ", Numero1, " y ", Numero2, " es: ", Suma;  
Escribir "La resta de ", Numero1, " y ", Numero2, " es: ", Resta;  
Escribir "El producto de ", Numero1, " y ", Numero2, " es: ", Producto;  
Escribir "La división de ", Numero1, " y ", Numero2, " es: ", Division;
```

FinAlgoritmo

</pre>

</code>

<footer>

<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>

</footer>

</body>

</html>

### Acrónimos y abreviaturas

Para definir en un documento web el significado de un acrónimo o de una abreviatura, se utilizan las etiquetas <acronym> y </acronym> y <abbr> y </abbr> respectivamente. Al pasar el ratón por encima de ellas, aparecerá el significado del mismo.

El valor del acrónimo o de la abreviatura se indica mediante el atributo title.

Ejemplo:

<!DOCTYPE html>

<html>

<head>

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

<meta http-equiv="Content-Language" content="ES-ES"/>

<meta name="description" content="Curso de Elaboración Básica de Páginas Web">

<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza, Nezahualcóyotl"/>

<meta name="author" content="Fernando Jiménez Ávila"/>

<meta name="distribution" content="Global"/>

<meta name="copyright" content="© Fernando Jiménez Ávila"/>

<title>Elaboración Básica de Páginas Web ..... CCAP México</title>



```
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1>Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de acrónimos y abreviaturas</h1>

<p>El lenguaje <acronym title="Hyper Text Markup Language">HTML</acronym> es uno de los temas de este material didáctico.</p>
<p>El lenguaje <abbr title="Hyper Text Markup Language">HTML</abbr> es uno de los temas de este material didáctico.</p>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

## Las fuentes

La etiqueta `<font>` permite controlar casi totalmente el estilo de un texto. Se puede establecer el tamaño de la fuente mediante el atributo `size`, la fuente del texto a usar mediante `face` y el color de la misma con el atributo `color`. Debe cerrarse mediante `</font>`.

Los tamaños del texto van del 1 al 17. El tamaño normal es el 3.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza, Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de tamaño de fuente</h1>

<font size=5>Texto en tamaño 5</font>
<font size=1>Texto en tamaño 1</font>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
```



</body>  
</html>

También se puede cambiar el tamaño de la fuente tomando como referencia el tamaño normal de la fuente.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza, Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de tamaño de fuente</h1>

<font size=-1>Texto más pequeño</font>
<font size=+1>Texto más grande</font>
<font size=+2>Texto mucho más grande</font>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

Cambiar el color de un texto se hace indicando los valores como en los ejemplos siguientes:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza, Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
```



Visión y Pasión por Emprender y Servir

```
<h2>¡No te asustes, no es tan difícil!</h2>
</header>
```

```
<!-- Contenido del documento -->
```

```
<h1>Uso de color de fuente</h1>
```

```
<font color="blue">Este texto está en azul</font>
<font color="red">Este texto está en rojo</font>
```

```
<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

Uno de los efectos más interesantes es la posibilidad de poder cambiar el tipo de fuente empleada. Eso sí, para poder visualizarlas se debe disponer de la fuente cargada en la computadora.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>
```

```
<!-- Contenido del documento -->
```

```
<h1>Uso de fuente</h1>
```

```
<font face="cosmic">Este texto se verá con la fuente cosmic</font>
```

```
<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

Si la fuente cargada en el documento web no está disponible en la computadora, la fuente que se visualizará será la predefinida en el navegador.

Si es necesario poner todo el texto con un mismo tamaño, se debe utilizar la etiqueta <basefont>.

Ejemplo:

```
<basefont size=5>
```

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de la etiqueta font</h1>

<font size="7" color="#ff0000">HTML</font>
<font size="6" color="#ff00ff">Disẽo</font>
<font size="5" color="#0000ff">del</font>
<font size="4" color="#00ffff"><i>FUTURO</i></font>

<br>

<font size="3" color="#ff0fef">q</font>
<font size="4" color="#ff0fef">u</font>
<font size="5" color="#ff0fdf">e</font>
<font size="6" color="#ff0fcf"><i> &nbsp;</i></font>
<font size="7" color="#ff0fbf">p</font>
<font size="+4" color="#ff0f9f">a</font>
<font size="+3" color="#ff0f7f">s</font>
<font size="+2" color="#ff00bf">a</font>
<font size="+1" color="#ef00df">d</font>
<font size="+0" color="#df00ff"><i>a</i></font>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

### Los enlaces

Otra función que caracteriza a HTML es el hipertexto o enlace, también conocido como hipervínculo. El hipertexto consiste en moverse a través de un documento o varios, seleccionando texto o imágenes para tal fin.

Cualquier elemento de un documento HTML puede ser definido como enlace. Por lo tanto, una imagen, un texto sencillo, un párrafo o un bloque, pueden ser definidos como enlaces a otros puntos del mismo documento web, a otro documento web o de Internet, inclusive.

## Estructura de los enlaces

La estructura general de un enlace está formada por la etiqueta <a> seguida de determinados atributos. Para comprender rápidamente la filosofía de trabajo de los enlaces, que también se les llaman anclas, se verá a continuación la estructura:

```
<a href="destino del enlace" title="descripción o nombre del enlace">texto indicativo del enlace</a>
```

El atributo href contiene la URL (Localizador Universal de Recursos) de la página web a visitar, por ejemplo, mientras que title mostrará una descripción del sitio, este atributo es opcional.

## Tipos de enlaces

Se analizarán a continuación varios tipos de enlaces.

### 1.- Enlaces a páginas o documentos web externos

Los siguientes ejemplos ilustran distintos modelos de enlaces al exterior:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1>Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de enlaces externos</a>

<!-- Enlace a una URL -->
<a href="http://www.ccapmexico.com" title="Visita CCAP México">Página principal del CCAP México</a>

<!-- Enlace a una dirección de correo electrónico -->
<a href="mailto:informes@ccapmexico.com" title="Envíame un correo electrónico">Correo electrónico de informes del CCAP México</a>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

### 2.- Enlace a un punto de la misma página (anclas)

Cuando se trabaja con documentos web muy extensos o índices, suelen ponerse enlaces desde el principio del documento web o al final, así se agiliza la tarea del usuario al examinar el documento web.





Visión y Pasión por Emprender y Servir

Tomando en cuenta esta situación, el primer paso consiste en marcar las distintas zonas del documento web a las que se desea poder acceder de manera inmediata; en este caso, el inicio y el final del documento.

Así pues, en el principio del documento se añade la línea de código siguiente:

```
<a id="inicio">
```

Se ha marcado como inicio un punto en el comienzo del documento web. Ahora se sitúa al final del documento web y se añade esta otra línea de código:

```
<a id="final">
```

Una vez marcadas las distintas zonas del documento web (ancladas), se van a situar en el mismo, los enlaces que permitirán ir de arriba debajo de la página web con un solo clic y viceversa.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de enlaces internos o anclas</h1>

<a id="inicio">
<a href="#final">Pulsa aquí para ir al final de esta página web</a>
<p>Como se ha mencionado anteriormente, las primeras fases del proceso de programación son la
<p>definición del problema y posteriormente se obtiene la representación adecuada de la descripción
<p>de la solución.</p>
<br><br><br><br><br><br><br><br><br>
<p>Esta descripción puede ser definida por representaciones adecuadas de símbolos a través de ciertos
<p>lenguajes de programación, que funcione como un vehículo descriptor y como un modelo de representación
<p>dada a la solución. Este lenguaje de programación deberá denotar independientemente con respecto a alguna
<p>computadora o lenguaje de programación en específico y debe ser capaz de representar o expresar cualquier idea computacional.</p>
<br><br><br><br><br><br><br><br><br>
<a id="final">
<p>Este lenguaje recibe el nombre de Pseudocódigo o Español Estructurado.</p>
<a href="#inicio">Pulsa aquí para ir al principio de esta página web</a>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
```

</html>

### 3.- Enlaces a otras páginas locales

Cuando se dispone de varias páginas web en un sitio web, es normal que desde una de ellas se pueda acceder a otra y, desde esta otra a una tercera página web, etc.

Cuando el origen de la página web está en el mismo directorio que la página web de destino a la que se quiere enlazar, bastará con escribir su nombre, ejemplo:

```
<a href="noticias.htm" title="Últimas noticias">Noticias de actualidad</a>
```

Si el destino estuviera en un subdirectorio, se debe especificar la ruta completa y correcta separando los directorios con una barra:

```
<a href="imagen/lucas/cuadros.html" title="Reportajes y fotos"> Las fotos de Don Lucas</a>
```

Finalmente, si el documento web de destino estuviera en un directorio superior, se deberá usar dos puntos y la barra, ejemplo:

```
<a href="../bienvenido.htm" title="Página de Inicio">Volver al Inicio</a>
```

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza, Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de enlaces o anclas</h1>

<!--Enlace a una URL -->
<a href="http://www.ccapmexico.com" title="Visita CCAP México" target="_blank">Página principal del CCAP México</a>
<br>
<!--Enlace a una dirección de correo electrónico -->
<a href="mailto:informes@ccapmexico.com" title="Envíame un correo electrónico">Correo electrónico de informes del CCAP México</a>
<br>
<!--Enlaces internos-->
<a id="inicio">
<a href="#final">Pulsa aquí para ir al final de esta página web</a>
<p>Como se ha mencionado anteriormente, las primeras fases del proceso de programación son la
<p>definición del problema y posteriormente se obtiene la representación adecuada de la descripción
<p>de la solución.</p>
<br><br><br><br><br><br><br><br>
<p>Esta descripción puede ser definida por representaciones adecuadas de símbolos a través de ciertos
<p>lenguajes de programación, que funcione como un vehículo descriptor y como un modelo de representación</p>
```

#### Visión y Pasión por Emprender y Servir

```
<p>dada a la solución. Este lenguaje de programación deberá denotar independientemente con respecto a alguna </p>
<p>computadora o lenguaje de programación en específico y debe ser capaz de representar o expresar cualquier idea computacional.</p>
<br><br><br><br><br><br><br>
<a id="final">
<p>Este lenguaje recibe el nombre de Pseudocódigo o Español Estructurado.</p>
<a href="#inicio">Pulsa aquí para ir al principio de esta página web</a>
<br><br>
<!--Enlaces a otras páginas del directorio-->
<a href="C:\Users\FOMENTO\Desktop\PaginaWeb1.htm" title="Página Web 1">Enlace a la Página Web 1</a>
<br>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

#### Listas

Las listas se emplean para presentar de forma ordenada una serie de líneas. Éstas pueden ser:

##### Listas desordenadas

Las listas desordenadas se utilizan para relacionar elementos sin orden específico. Las etiquetas que definen este tipo de listas son `<ul>` y `</ul>` respectivamente, mientras que las etiquetas que se utilizan para indicar las entradas o los elementos de la lista son `<li>` y `</li>`.

El atributo `type` puede tomar el valor de `circle` (círculo sin relleno), `disc` (círculo con relleno) y `square` (cuadrado). Si este atributo se aplica a la etiqueta `<ul>`, toda la lista utilizará el símbolo indicado, si, por el contrario, se aplica a la etiqueta `<li>`, que es la que señala los distintos elementos de la lista, el cambio de símbolo sólo afectará al elemento actual y a los siguientes hasta nueva indicación.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza, Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1>Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de listas desordenadas</h1>

<p> Nombres compuestos</p>
<ul>
<li type="square">Juan Carlos</li>
```

#### Visión y Pasión por Emprender y Servir

```
<li type="circle">Jos&eacute; Luis</li>
<li type="disc">Luis Miguel</li>
<li>Mar&iacute;a de las Nieves</li>
<li>Pepe Nacho</li>
</ul>
```

```
<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

#### Listas ordenadas

Este tipo de lista se utiliza para relacionar elementos con un orden determinado, precedido de un número o de una letra que se incrementa automáticamente. Las etiquetas que definen este tipo de lista son `<ol>` y `</ol>` respectivamente.

Así pues, con dicha etiqueta y con la etiqueta `<li>`, se pueden representar en pantalla distintos elementos de manera organizada.

Las listas ordenadas siempre comienzan por 1 y van aumentando progresivamente; no obstante, estas características pueden controlarse mediante los atributos `type` y `start`.

El atributo `type` determina el formato de la marca de numeración:

- Letras mayúsculas (`type="A"`).
- Letras minúsculas (`type="a"`).
- Números romanos mayúsculas (`type="I"`).
- Números romanos minúsculas (`type="i"`).
- Números (`type=1`).

El atributo `start` especifica el número del primer elemento de la lista para valores distintos a 1, tomado por defecto. El orden se da siempre en números, independientemente del tipo de marca seleccionada.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de listas ordenadas</h1>
```

```
<ol type="i">
<li>Introducción</li>
<li>Breve repaso de HTML</li>
<li>Las Hojas de estilo en cascada</li>
<li>JavaScript</li>
<li>Referencia de HTML</li>
<li>Referencia de CSS</li>
<li>Referencia de JavaScript</li>
</ol>
```

```
<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

#### Listas de definición

Las listas de definición son apropiadas para glosarios, catálogos, etc. Los distintos elementos deben ir englobados entre las etiquetas <dl> y </dl> y están formadas por un primer elemento, al que se le podría denominar término o título, concretado con la etiqueta <dt> y un segundo llamado definición, señalado con la etiqueta <dd>.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza, Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1>Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de lista de definición</h1>

<dl>
<dt>Blue</dt>
<dd>Color azul</dd>
<dt>Yellow</dt>
<dd>Color amarillo</dd>
<dt>Pink</dt>
<dd>Color rosa</dd>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
```

</html>

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza, Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de listas</h1>

<ul>
<li>Introducción</li>
<li>Breve repaso de HTML</li>
</ul>
<ol>
<li>Las Hojas de estilo en Cascada</li>
<li>JavaScript</li>
</ol>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

## Imágenes

Para incluir imágenes en documentos HTML se emplea la etiqueta <img> junto a dos atributos obligatorios: src para definir la imagen a mostrar y alt encargado de mostrar una descripción de la imagen al pasar el cursor sobre ella, el modelo es:

```

```

Es importante conocer los distintos formatos gráficos que se pueden utilizar. Si se especifica un formato no reconocido, el navegador simplemente lo ignorará y no se visualizará la imagen en el documento web, así pues, las imágenes deben estar en formato .gif, .jpg, .jpeg, .png.

Las imágenes GIF ocupan poco espacio, se cargan rápidamente y sólo pueden trabajar con 256 colores, como contrapartida, pueden hacerse fondos transparentes.

El formato JPEG tiene un tamaño aún menor que el GIF y trabaja con color de 24 bits, es decir, 16.7 millones de colores.

El formato PNG es el estándar y se puede trabajar con varias profundidades de paleta de color, pero no todos los navegadores lo implementan correctamente.

### Visión y Pasión por Emprender y Servir

Existen dos atributos de suma importancia para las imágenes: width y height (ancho y alto), con los que se puede redimensionar una imagen para adaptarla a las necesidades propias o simplemente, indicarlos para que el navegador tenga más información en el momento de abrir el documento web.

Ejemplo:

```

```

Si se quiere reducir o aumentar el tamaño de una imagen (a nivel de visualización), se puede indicar el valor en píxeles o porcentaje.

Por ejemplo, width="150" muestra la imagen con 150 píxeles de ancho, mientras que width="50%" reduce la imagen al 50%. En ambos casos, lo único que varía es el tamaño que se muestra, no el tamaño real de la imagen.

También se puede hacer que una imagen sea un enlace a otra página mediante la etiqueta <a>. así pues, para convertir una imagen en un enlace se debe encerrar la definición de la imagen entre las etiquetas del enlace.

Las imágenes que contienen enlaces presentan un borde y el cursor cambia al pasar sobre ellas.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de imágenes</h1>

<p><a href=http://www.ccapmexico.com title="Acceso a la Página Web del CCAP México"></a></p>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

### Alineación de imágenes con texto

El atributo align permite alinear la imagen con respecto al texto en función de los siguientes valores: left, right, top, bottom, middle, mientras que los atributos vspace y hspace, se encargan de definir espacios en blanco arriba y abajo o izquierda y derecha, respectivamente. El valor se especifica en píxeles.

### Imágenes mapeadas

Un componente de las páginas web ciertamente llamativo son las imágenes mapeadas o sensibles. Dicho de otra manera, imágenes en las que se definen diversas zonas que activan distintos enlaces cuando son seleccionadas. Al ir desplazando el cursor sobre la imagen, éste se convierte en el símbolo normal para enlazar con otras páginas (una mano, generalmente). Si se pulsa sobre dicha zona, el navegador realizará la tarea programada, como acceder a un enlace determinado o ejecutar un script, por ejemplo.

### Desarrollo de la imagen

El primer paso a realizar es el diseño de la imagen mediante un programa gráfico, por ejemplo, se puede dibujar un círculo, un cuadrado, un rectángulo y un triángulo. Después, para definir las distintas zonas de la imagen se necesitan conocer las coordenadas que delimitan las áreas de la imagen que se desea sensibilizar o mapear, para ello, se tomarán las coordenadas que delimitan las distintas áreas mediante un programa de dibujo. Finalmente, se asigna a cada zona su correspondiente enlace o acción.

### Programación del código

En primer lugar, se debe definir la imagen a utilizar mediante la etiqueta <img> y asignarle un nombre al mapa resultante mediante el atributo usemap. Al nombre de dicho mapa habrá que anteponerle el símbolo #.

A continuación, se abre el mapa en cuestión mediante la etiqueta <map>.

Ejemplo:

```

```

A continuación, viene el proceso más laborioso. Asignar a cada parte de la imagen la acción deseada. En el ejemplo, se ejecuta una sencilla rutina de JavaScript que facilita información sobre el área seleccionada sin necesidad de pulsar el ratón.

La etiqueta <área> se encarga de gestionar las distintas áreas del mapa con la ayuda de los atributos shape, coords y href.

El atributo shape se encarga de delimitar las distintas zonas del mapa según los siguientes valores:

- rect. Usado para delimitar una superficie rectangular por medio de las coordenadas del vértice superior izquierdo e inferior derecho.
- circle. Delimita una circunferencia mediante la coordenada del centro de la misma y el valor del radio.
- poly. Polígono definido mediante un conjunto de líneas. Se deben definir las coordenadas de los distintos vértices de las polilíneas.

Las distintas áreas del mapa deben ser definidas mediante coordenadas; por lo tanto, se utilizará el atributo coords para introducir dichas coordenadas junto con la URL al que deberá accederse.

Así pues, el código necesario para el mapa sería similar al del siguiente ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>
```



<!-- Contenido del documento -->

<h1>Uso de imagen mapeada o sensible</h1>

```
<map name="formas" id="formas">
<área shape="rect" coords="9,11,33,46" href="" onmouseover="alert('Soy el cuadrado azul')" alt="Cuadrado azul">
<área shape="circle" coords="54,46,10" href="" onmouseover="alert('Soy el círculo rojo')" alt="Círculo rojo">
<área shape="rect" coords="8,65,70,86" href="" onmouseover="alert('Soy el rectángulo verde')" alt="Rectángulo verde">
<área shape="poly" coords="39,11,67,11,66,36" href="" onmouseover="alert('Soy el triángulo negro')" alt="Triángulo negro">
</map>

```

```
<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

**Nota:** Algunos navegadores pueden mostrar en la barra de información amarilla un mensaje indicando que se ha bloqueado el contenido activo, todo lo que hay que hacer es aceptar o desbloquear en Permitir contenido bloqueado....

### Tablas

Las tablas eran el elemento estrella del HTML hasta la aparición de las hojas de estilo en cascada (CSS). Con ellas, el tema de las tabulaciones o espacios de más y las alineaciones se resolvían de forma sencilla. Las tablas están divididas en celdas que pueden contener texto, listas, imágenes, enlaces, formularios, etc.

Como comentario, las hojas de estilo en cascada (CSS) permiten controlar la maquetación o el diseño de un documento web al 100% sin necesidad de ninguna tabla.

#### Estructuración de una tabla

Las etiquetas <table> y </table> son las encargadas de definir el principio y el final de una tabla. El atributo width determina la anchura de la tabla mediante un porcentaje de la dimensión de la pantalla o un determinado número de píxeles.

Ejemplos:

```
<table width="50">
</table>
```

```
<table width="50%">
</table>
```

Se puede añadir un borde en la tabla mediante el atributo border y variar la separación entre celdas mediante el atributo cellpadding y la separación entre el borde de la celda y el contenido de la misma con el atributo cellspacing.

Así pues, se puede configurar una tabla con un grosor de borde de 2 unidades, separación entre celdas de 10 y separación del contenido de las mismas con respecto a los bordes de 20.

Ejemplo:

```
<table border="2" cellspacing="10" cellpadding="20">
```

La etiqueta <caption> se utiliza para poner un título a la tabla, ejemplo:

```
<caption> ésta es a cabecera de la tabla</caption>
```

### Visión y Pasión por Emprender y Servir

A continuación, se deben definir las celdas y las filas de la tabla. Para ello, se utilizarán las etiquetas <td> y </td> para el inicio y el final de cada una de las celdas y <tr> y </tr> para el inicio y el final de cada fila. También se pueden utilizar las etiquetas <th> y </th> para definir el inicio y el final de una celda de cabecera. El contenido de ésta aparecerá en negritas y centrado.

Si se quiere diseñar una tabla de 10 celdas y dos filas para clasificar los números del 0 al 9, el código sería similar al del siguiente ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de tablas</h1>

<table>
<tr>
<td>0</td><td>1</td><td>2</td><td>3</td><td>4</td>
</tr>
<tr>
<td>5</td><td>6</td><td>7</td><td>8</td><td>9</td>
</tr>
</table>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

Como se puede ver en el ejemplo anterior, la etiqueta <tr> se repite tantas veces como filas se quiera que tenga la tabla, mientras que la etiqueta <td> hace lo propio con las distintas celdas.

### Alineación y dimensión de las celdas

Generalmente el contenido de una celda está alineado a la izquierda (left) y centrado verticalmente (middle). No obstante, se dispone de los correspondientes valores de alineación, como se muestra en el siguiente ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
```



Visión y Pasión por Emprender y Servir

```
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de tablas</h1>

<table>
<td align="center">Alineado en el centro</td>
<td align="left">Alineado a la izquierda</td>
<td align="right">Alineado a la derecha</td>
<td align="justify">Justifica el texto</td>
<td align="top">Alineado a la parte superior de la celda</td>
<td align="middle">Alineado en el centro de la celda</td>
<td align="bottom">Alineado a la parte inferior de la celda</td>
<td align="baseline">Alineado a la base de la celda</td>
</table>
<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

También se puede variar la anchura y la altura de una determinada celda con respecto a las otras.

Los atributos colspan y rowspan permiten que una celda ocupe el espacio de varias, expandiéndose hacia la celda de la derecha o hacia la siguiente fila.

Así pues, colspan indica el número de celdas contiguas que debe ocupar hacia la derecha, mientras que rowspan indica cuántas filas hacia abajo se expande la celda.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
```



#### Visión y Pasión por Emprender y Servir

```
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>
```

```
<!-- Contenido del documento -->
```

```
<table>Uso de tablas<h1>
```

```
<table border="2">
<caption>Ejemplo de alineaciones y dimensionados</caption>
<tr>
<td colspan="2" align="left"> Conexión Internet</td>
<td align="center"> Precio</td>
<td align="center"> Pesos</td>
</tr>
<tr>
<td rowspan="2" align="middle"> Módem</td>
<td>Teléfono</td>
<td>Gratis</td>
<td>Gratis</td>
</tr>
<tr>
<td>Proveedores</td>
<td>2,500.00</td>
<td>1,503.00</td>
</tr>
</table>
```

```
<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

#### Otros atributos

El atributo que controla el color de los distintos elementos de la tabla es bgcolor. Para hacer que toda la tabla tenga como color de fondo el verde, el código será como el siguiente:

```
<table bgcolor="green">
```

Si, por el contrario, sólo se quiere que una determinada celda tenga como color de fondo verde, el atributo bgcolor deberá estar en la etiqueta que defina a dicha celda, ejemplo:

```
<td bgcolor="green">
```

Para combinar distintos colores, se utilizará tantas veces sea necesario el atributo bgcolor, recordando que el color que predomine podrá establecerse en la etiqueta <table>.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza, Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
```

Visión y Pasión por Emprender y Servir

```
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de tablas</h1>

<table width="500" border="2" cellspacing="5" cellpadding="5">
<tr>
<td bgcolor="#cc3333" align="left"> CELDA 1</td>
<td>CELDA 2</td>
<td bgcolor="#ffccff" colspan="2" rowspan="2">

</td>
</tr>
<tr>
<td></td>
<td bgcolor="#3366ff" align="right">CELDA 3</td>
</tr>
<tr>
<td colspan="2" rowspan="2">&nbsp;&nbsp;&nbsp;</td>
<td colspan="2">&nbsp;&nbsp;&nbsp;</td>
<td colspan="2">&nbsp;&nbsp;&nbsp;</td>
<td colspan="2">&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr>
<td colspan="2">&nbsp;&nbsp;&nbsp;</td>
<td colspan="2">&nbsp;&nbsp;&nbsp;</td>
<td colspan="2">&nbsp;&nbsp;&nbsp;</td>
<td colspan="2">&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr>
<td colspan="2">&nbsp;&nbsp;&nbsp;</td>
<td colspan="2">&nbsp;&nbsp;&nbsp;</td>
<td colspan="2">&nbsp;&nbsp;&nbsp;</td>
<td colspan="2">&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr>
<td colspan="2">&nbsp;&nbsp;&nbsp;</td>
<td colspan="2">&nbsp;&nbsp;&nbsp;</td>
<td colspan="2">&nbsp;&nbsp;&nbsp;</td>
<td colspan="2">&nbsp;&nbsp;&nbsp;</td>
</tr>
</table>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

Reproducción de audio

La capacidad de reproducir sonido en una página web viene determinada en gran medida por el navegador que se esté utilizando.

Música de fondo

La etiqueta encarga de reproducir un archivo de audio es <bgsound> asociada al atributo src. Los archivos de audio podrán ser del formato .wav, .au, .mid, .avi, .mpg, .mpeg, .ra, .mov, .mp3 y .ogg.

Un atributo fundamental es loop. Será usado para especificar el número de veces que se debe ejecutar o reproducir el archivo de audio.

Loop puede tener un valor numérico que indicaría el número de veces que se reproducirá el archivo de audio en un bucle cerrado.

Ejemplo:

#### Visión y Pasión por Emprender y Servir

<bgsound src="música.mid" loop="infinite">

También se dispone de dos atributos que son: balance y volumen. El atributo balance acepta valores comprendidos entre -10000 y 10000 y el atributo volumen entre -10000 y 0.

#### Los frames

Los frames son marcos que dividen la pantalla en filas y columnas según sean las necesidades.

Cuando se carga un nuevo documento web, el anterior desaparece de la pantalla. Con la utilización de marcos se puede dividir la pantalla en ventanas y visualizar, por ejemplo, un menú en la parte izquierda y los resultados de los enlaces en la parte derecha.

#### Los enlaces y el atributo target

El primer paso antes de utilizar los marcos consiste en conocer las características del atributo target, que determinará la zona de la pantalla en donde se mostrarán los elementos del enlace seleccionado.

Ejemplo:

```
<!--Código para la parte superior -->
<a href="top.htm" target="arriba"> Pulse aquí para ir a la ventana superior</a>
<!--Código para la parte inferior -->
<a href="down.htm" target="abajo"> Pulse aquí para ir a la ventana inferior</a>
```

Como se puede observar, el atributo target determina el lugar de destino del contenido del enlace especificado. Así pues, una vez dividida la pantalla, se deben indicar en los enlaces del código HTML el lugar de la pantalla en donde serán visualizados.

Este atributo tiene una serie de valores:

- target="\_blank". El enlace siempre se carga en una nueva ventana.
- target="\_self". El enlace se carga en la misma ventana en que se encuentra.
- target="\_parent". El enlace se carga en el frameset inmediatamente superior del documento.
- target="\_top". El enlace se carga en la totalidad de la ventana.

#### Estructura básica

Los marcos permiten dividir las páginas HTML en varias ventanas con barras de desplazamiento y características independientes. Por ejemplo, se puede asignarle una URL para que pueda cargar la información solicitada en otras ventanas de la página web. También se puede hacer que el tamaño de la ventana sea modificado por el usuario.

Una de las configuraciones más típicas es la de diseñar ventanas paralelas para realizar consultas desde una de ellas y ver los resultados en la otra. La sintaxis de las vistas es muy similar a la de las tablas y están diseñadas para ser procesadas rápidamente por los visores.

La etiqueta a utilizar es <frameset>, que es la que permite dividir la pantalla en ventana, ya sean verticales u horizontales. Los atributos encargados de determinar el aspecto de esas ventanas son rows y cols, filas y columnas respectivamente.

El atributo rows se encarga de definir el número de divisiones verticales, mientras que el atributo cols hace lo propio, pero con las divisiones horizontales.

Los valores para ambos atributos pueden ser absolutos en píxeles o valores de escala relativos en porcentajes.

La forma de introducir los valores de división de la pantalla se basa en los siguientes formatos:

- n. si se introduce un valor n determinado, se estará indicando altura o anchura de la ventana en píxeles.
- %. Este valor indica que la altura o anchura de la ventana es un porcentaje relativo al tamaño de la ventana que la contiene.
- \*. Un asterisco indica que debe asignarse a la ventana todo el espacio disponible. En el supuesto de que haya varias ventanas, el espacio libre se divide entre ellas y, si hay un valor delante del asterisco, la ventana que lo tenga asignado tomará más espacio relativo, por ejemplo, "2\*,\*", daría 2/3 partes del espacio a la primera ventana y un tercio a la segunda.

Ejemplo:

```
<!--Ejemplo 1-->
<frameset cols="100%" rows="25%,25%,25%,*">
<!--Ejemplo 2-->
<frameset cols="100%" rows="25%,25%,25%,25%">
```

Si se quiere añadir ventanas dentro de otras ventanas, se deben añadir tantos <frameset> como sean necesarios.

### Las etiquetas <frame> y <noframe>

Una vez dividida la pantalla en distintas ventanas se deberá hacer algo para que en esos espacios aparezca la información.

La etiqueta <frame> es la encargada de llamar al documento HTML que se debe cargar en una ventana. Así pues, se deben utilizar tantas ventanas se hayan creado.

Esta etiqueta puede tener hasta seis atributos distintos:

- **src="url"**. El atributo src toma como valor la URL del documento web que se debe mostrar en esa ventana en particular. Si no se incluye, la ventana quedará vacía.
- **name="nombre de la ventana"**. Como se puede deducir, se emplea para dar un nombre a una ventana, de manera que pueda ser el destino de cualquier enlace.
- **marginwidth="valor"**. Se emplea cuando se quiera precisar un número de píxeles entre los bordes izquierdo y derecho de la ventana.
- **marginheight="valor"**. Similar al anterior, pero con respecto a los bordes superior e inferior.
- **scrolling="yes|no|auto"**. Este atributo se utilizará para mostrar las barras de desplazamiento o no, yes mostrará siempre las barras, no nunca pondrá las barras de desplazamiento y auto hará que el navegador decida si son necesarias o no en función del contenido de la misma.
- **onresize**. Atributo que indica al navegador que la ventana no pueda ser redimensionada por el usuario.

Para aquellos navegadores que no pueden gestionar frames, la etiqueta <noframes> se emplea para incluir una página alternativa.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza, Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1>Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de frames o marcos</h1>

<frameset rows="*" cols="280,*">
<frame src="marco1.htm" name="leftmarco" frameborder="yes" scrolling="no" noresize="noresize" id="leftmarco">
```

**Visión y Pasión por Emprender y Servir**

```
<frameset rows="80,*">
<frame src="marco2.htm" name="topmarco" frameborder="yes" scrolling="no" noresize="noresize" id="topmarco">
<frameset rows="80,*,80">
<frame src="marco3.htm" name="topmarco1" frameborder="yes" scrolling="no" noresize="noresize" id="topmarco1">
<frame src="marcos.htm" name="mainmarco" id="mainmarco">
<frame src="marco4.htm" name="bottommarco" frameborder="yes" scrolling="no" noresize="noresize" id="bottommarco">
</frameset>
</frameset>
</frameset>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

**Los formularios**

Todos los elementos del HTML, es decir, las etiquetas vistas para la realización de páginas HTML, carecen de interactividad real con el usuario. Con los formularios se verá cómo el usuario puede pedir y proporcionar información.

Por ejemplo, se puede diseñar un formulario para que los visitantes puedan facilitar sus datos personales y recibir una determinada información.

También puede pensarse en un listado de productos, en los que el usuario escribe el nombre del producto y recibe, a continuación, el precio del mismo, su descripción, etc.

Los formularios pueden introducirse en cualquier parte del documento web, es decir, en tablas, dentro de un texto preformateado, etc. Además, puede contener en su interior listas, imágenes, etc.

La etiqueta que define el inicio de un formulario es `<form>` y el fin del mismo `</form>`.

También se pueden encontrar las etiquetas que definen los campos de entrada `<input>`, las que definen los campos de selección `<select>` y las áreas de texto `<textarea>` y `</textarea>`.

Ejemplo:

```
<form action="archivo de tratamiento" method=post/get">Elementos del formulario</form>
```

**Declaración del formulario <form>**

Como se mencionó anteriormente, los elementos que componen un formulario deben encontrarse entre las etiquetas `<form>` y `</form>`.

A continuación, se situarán todas las etiquetas que generan los botones, entradas de texto, casillas de verificación, etc.

También se pueden utilizar otros elementos del HTML para dar un toque de profesionalidad al formulario.

Los atributos de `<form>` son:

- **action.** Atributo que determina la acción que debe realizarse al pulsar el botón de ejecución.
- **method.** Indica el método de transferencia de la información introducida en el formulario. Pueden emplearse los métodos `get` cuando se trate de scripts (`cgi` o `php`, por ejemplo) y `post` si la información va a ser enviada por correo electrónico.
- **enctype.** Atributo que determina la forma de codificación empleada para el transporte del contenido del formulario.

**Definición de campos de entrada <input>**

La etiqueta `<input>` se emplea para definir todos los elementos del formulario, como botones, cajas de texto, casillas de verificación, etc. El atributo que determina el tipo de elemento es `type`.

Los distintos tipos de entrada son:



#### Visión y Pasión por Emprender y Servir

- text. Como su nombre lo indica, se utiliza para la entrada de texto.
- checkbox. Cuadro vacío de selección. Permite selecciones múltiples.
- hidden. Se utiliza para pasar los datos adquiridos en un formulario a otro sin que se visualice nada en pantalla.
- password. Permite la introducción de texto visualizándose para cada carácter un asterisco.
- radio. Círculo vacío de selección. Solo permite una opción entre varias.
- reset. Borra todos los datos introducidos en el formulario.
- submit. Envía la información del formulario.

#### Text

Los campos básicos de todo formulario son los de texto. Ejemplo:

```
<input type="text" id="nombre" name="nombre" size="30">
```

El valor por defecto del atributo type en los campos input es text, por lo tanto, si se omite en la declaración del input, el navegador lo interpretará como un campo de texto.

Los atributos son:

- type="text", indica que se trata de un campo de texto.
- size, determina el número de caracteres visibles de la casilla.
- id, es el identificador del elemento y su misión es impedir que haya dos elementos con el mismo nombre.
- name, asigna el nombre a la variable del campo.
- maxlength, delimita el número máximo de caracteres que pueden introducirse en el campo. El número de caracteres por defecto es ilimitado.
- value, permite poner un valor inicial para el campo.
- title, muestra una leyenda al situar el cursor sobre el campo

Ejemplo:

```
<input name="Sugerencia" id="Sugerencia" type="text" title="Introduzca su sugerencia" value="Escriba aquí ..." size="20" maxlength="180">
```

Se puede completar la definición del campo asociando un texto al mismo, con la finalidad de informar al usuario. Para ello se dispone de las etiquetas <label> y </label> que se encargan de asociar un texto con un determinado elemento del formulario.

Ejemplo:

```
<label for="Sugerencia">Introduzca una sugerencia</label>
<input name="Sugerencia" id="Sugerencia" type="text" title="Introduzca su sugerencia" value="Escriba aquí ..." size="20" maxlength="180">
```

#### Password

Este tipo de campo se utiliza para introducir claves o contraseñas de acceso. El texto que se va escribiendo en la ventana no es legible para los usuarios, ya que los caracteres son sustituidos por asteriscos a medida que se van introduciendo.

Los atributos son los mismos que los aplicados al campo texto.

Ejemplo:

```
<input name="clave" id="clave" type="password" title="Introduzca la clave de acceso" value="" size="8" maxlength="8">
```

#### Hidden

Los campos hidden son como los campos de texto a excepción de que éstos están ocultos para el usuario, es decir, no se visualizarán en la página web. Son muy útiles para pasar variables en formularios a las que no debe tener acceso el usuario.

Al igual que el tipo anterior, es obligatorio especificar los atributos name, value e id.

### Checkbox

La casilla de verificación o checkbox, se utiliza para seleccionar mediante un cuadrado una serie de elementos de un formulario.

Los elementos de este tipo requieren los atributos name, value e id, que indican el nombre de la variable del campo, el valor del campo y el identificador del campo respectivamente. Por regla general, para el atributo id y name siempre se escribirá el mismo valor.

También se puede incluir el atributo checked cuando se quiera que esa opción aparezca seleccionada por defecto de la siguiente manera:

```
checked="checked"
```

Ejemplo:

```
<p>Selecciones una opción<br>
<input name="Sexo" id="Hombre" type="checkbox" value="HombreOK">
<br>
<input name="Sexo" id="Mujer" type="checkbox" value="MujerOK">
</p>
```

### Radio

Esta opción es similar a checkbox, pero solo permite elegir una opción de entre todas.

Los elementos de este tipo requieren los atributos name, value e id, que indican el nombre de la variable del campo, el valor del campo y el identificador respectivamente. En este caso, es muy importante que el identificador y el nombre no sean idénticos, ya que solo se puede seleccionar una opción de las disponibles.

Ejemplo:

```
<p>
<input type="radio" name="edad" id="18" value="18" checked="checked">
<label for="name">Menos de 18 años</label><br>
<input type="radio" name="edad" id="18-25" value="18-25" checked="checked">
<label for="name">Entre 18 y 25 años</label><br>
<input type="radio" name="edad" id="25-35" value="25-35" checked="checked">
<label for="name">Entre 25 y 35 años</label><br>
<input type="radio" name="edad" id="35-50" value="35-50" checked="checked">
<label for="name">Entre 35 y 50 años</label><br>
<input type="radio" name="edad" id="+50" value="+50" checked="checked">
<label for="name">Mayor de 50 años</label><br>
</p>
```

### Definición de campos de selección <select>

Las etiquetas <select> y </select> permiten al usuario seleccionar una única opción de un conjunto de elementos mostrados como una lista desplegable. Las distintas opciones del campo de selección se introducen mediante los elementos <option> y </option>.

Con el atributo selected se puede visualizar por defecto una determinada opción de la lista, ejemplo:

```
selected="selected"
```

Los demás atributos que reconoce son:

- multiple. Por defecto solo puede ser seleccionada una sola opción. Este atributo permite seleccionar más de una opción, pulsando la tecla CTRL junto con la opción.
- name. Atributo que especifica el nombre de la lista de selección.
- size. Atributo que determina el número de ítems u opciones visibles superior a 1. En este caso, se muestra una barra de desplazamiento lateral.

#### Visión y Pasión por Emprender y Servir

Ejemplo:

```
<select name="aspecto" size="1">
<option selected="selected">Excelente</option>
<option selected="selected">Bueno</option>
<option selected="selected">Regular</option>
<option selected="selected">Malo</option>
</select>
```

#### Área de texto <textarea>

Para definir un área de texto de múltiples líneas, se deben utilizar las etiquetas <textarea> y </textarea>. Los elementos de este tipo requieren los atributos name, title e id, ya comentados anteriormente.

Para definir las dimensiones del área de texto, se disponen de los siguientes atributos:

- cols. Determina el número de caracteres de ancho de la ventana de texto.
- rows. Establece el número de líneas visibles del área de texto, en caracteres.

Ejemplo:

```
<textarea name="quejas" id="quejas" rows="8" cols="60">
Aproveche este espacio para mostrar sus quejas
</textarea>
```

#### Reset

Como su nombre lo indica, se usa para resetear o limpiar el contenido del formulario. Su aspecto es el de un botón y su nombre se especifica con el atributo value.

Ejemplo:

```
<input type="reset" value="Borrar">
```

#### Submit

Es el botón que se encarga de enviar los datos del formulario. Al igual que en el caso anterior, la etiqueta del botón se determina mediante el atributo value.

```
<input type="submit" value="Pulsa aquí para enviar el formulario">
```

### Introducción a las Hojas de Estilo en Cascada (CSS)

Las llamadas hojas de estilo en cascada, cascading style sheets (CSS) abren un abanico de posibilidades para los creadores de páginas web.

Las CSS se convirtieron en una recomendación del W3C (World Wide Web Consortium)

#### Para qué sirven las Hojas de Estilo en Cascada

Usando CSS, los webmasters pueden, por ejemplo, generar un estilo patrón para todo el resto del documento web, con el consiguiente ahorro de tiempo en diseño y mantenimiento.

Las CSS constituyen el complemento ideal para el HTML. Su misión es definir la apariencia y el estilo de sus elementos. Por ejemplo, se puede especificar en una línea que todo el texto enfatizado del documento web use la fuente Verdana en color verde con un tamaño de 12 puntos.

Por ejemplo:

#### Opción HTML

```
<b><font fase="Verdana" size="+2" color="green"> Texto destacado</font></b>
```

#### Opción CSS

```
<style type="text/css">
em {font-size:12; color:green; font:verdana;}
</style>
```

```
<em>Texto destacado</em>
```

Aunque de entrada puede parecer que con CSS es mucho más complicado, se debe pensar que una vez definido el estilo, no se tendrá que volver a hacerlo y bastará con que se coloque la etiqueta `<em>` para que el formato indicado sea utilizado, mientras que con HTML se tendrá que introducir cada vez que se quiera resaltar algo.

Sin duda, el ahorro de líneas de código está garantizado con CSS y el aspecto profesional de la página web asegurado.

Con CSS se podrá definir áreas dentro de un documento web y aplicarles un borde, un color de fondo o una imagen. Se podrán colocar imágenes, tablas, texto; en definitiva, elementos de HTML en un lugar exacto del documento web, sea cual sea el aspecto o tamaño del navegador. Se podrá controlar el sonido en un documento web de una manera concreta con multitud de instrucciones específicas para ajustar volumen, velocidad, etc. También se dispone de filtros para imágenes o transiciones entre pantallas. Resumiendo, un sinfín de posibilidades que, hasta ahora, no estaban al alcance con HTML.

No obstante, nunca se debe olvidar que el usuario es el que tiene siempre la última palabra para decidir cómo será visualizado el documento web en la pantalla de su computadora a través del navegador que utilice.

#### Pros y contras de las Hojas de Estilo en Cascada

CSS tiene ventajas y desventajas en su uso. Las principales ventajas son:

- Se puede modificar la presentación de todos los elementos estándar del documento web sin tener que modificar el código HTML estructural.
- Se dispone de comandos y atributos más potentes y precisos con los que poder maquetar exactamente un documento web.
- Es un lenguaje muy sencillo, ya que se basa en el uso de propiedades muy intuitivas, similares a las de un procesador de textos.
- Se puede generar un estilo externo que contenga todas las definiciones de estilo de un documento web y modificar éste únicamente para efectuar cambios en una o varias páginas web.
- Puede combinarse con JavaScript.
- Su uso estructurado y razonado permite ahorrar muchas líneas de código.

La desventaja de las CSS es la incompatibilidad entre los diferentes navegadores. Aunque todos ellos aseguran la compatibilidad, lo cierto es que, entre versiones de un mismo navegador, ya hay irregularidades.

La solución a este inconveniente pasa por conocer las propiedades implementadas en cada versión del navegador probando los CSS en varios de ellos.

En general, si se basa en los estándares anteriores, la visualización de las CSS estará asegurada y se obtendrán buenos resultados en casi todos los navegadores.

#### Cascading Style Sheets (CSS)

Las CSS tienen una estructura simple, flexible y potente. Dada su fácil estructuración se puede definir la apariencia de cada elemento o grupo de ellos con facilidad cambiando posteriormente su diseño si es necesario, de forma simple y rápida.

#### Sintaxis y reglas

Se dispone de multitud de formas o reglas para definir los estilos dentro de un documento web dependiendo de la finalidad a conseguir. Se puede definir un estilo de manera global, es decir, especificar las propiedades que van a afectar a todo el documento web, aplicar estilos a un elemento HTML en concreto, establecer estilos dependiendo del contexto que rodea al elemento, estar dentro de una tabla, formar parte de una lista, etc.

### Primeros pasos

Una hoja de estilo en cascada (CSS) se define de manera similar a los scripts de JavaScript. La definición global de un estilo estará delimitada por las etiquetas `<style>` y `</style>` respectivamente y en su interior se podrán establecer los estilos que se utilizarán en todo el documento web.

Para especificar el tipo de sintaxis que se empleará se utilizará el atributo `type`. Su valor por defecto es `"text/css"` y selecciona la sintaxis CSS.

Ejemplo:

```
<style type="text/css">
<!--
p {color:green}
-->
</style>
```

Se recomienda definir estilos globales dentro de la cabecera del documento, entre `<head>` y `</head>`.

Una opción más interesante consiste en definir los CSS en un archivo de texto plano aparte para después referenciarlo desde cualquier documento de HTML mediante una sencilla línea de código. Su utilidad es más que evidente, ya que las páginas web que utilizan un mismo estilo, sólo tienen que hacer la correspondiente llamada al estilo mediante la etiqueta `<link>` ubicando en su descripción, la URL del archivo externo.

Ejemplo:

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

La sintaxis para definir estilos en archivos es idéntica a la que se usa para definirlos en el propio documento web, excepto que no es necesario incluir la etiqueta `<style>`.

A continuación, se muestra, a modo de ejemplo, una hoja de estilo en cascada (CSS) que dibuja dos capas en la pantalla con posicionamientos absolutos. El archivo debe guardarse o grabarse en formato de texto plano, con la extensión `.css`.

```
#capa1 {
position:absolute;
top:50px;
left:20px;
background-color:#ffcc00;
width:300px;
height:300px;
}
```

```
#capa2 {
position:absolute;
top:80px;
left:90px;
background-color:#33ff33;
width:200px;
height:200px;
}
```

Así pues, el código necesario para incluir la CSS externa desde un documento web, sería como en el siguiente ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza, Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
```



Visión y Pasión por Emprender y Servir

```
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
<link href="mihoja.css" rel="stylesheet2 type="text/css">
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<div id="capa1">Esta es la CAPA 1</div>
<div id="capa2">Esta es la CAPA 2</div>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

### Estilos de etiqueta

La sintaxis básica es la siguiente:

selector {propiedad:valor}

El selector es un elemento de HTML al que afectará la definición del estilo en función del valor asignado a la propiedad que se desea especificar. Al conjunto propiedad:valor se le llama declaración.

Por ejemplo:

p {color:green}

En el ejemplo anterior, se está indicando que todos los párrafos del documento web serán de color verde.

Si además se quiere que el texto de los párrafos esté justificado, habrá que añadir la propiedad correspondiente, en este caso:

p {text-align:justify}

Con el ejemplo anterior ya se tienen dos líneas de código y sólo se acaba de empezar, la sintaxis de CSS permite añadir todas las propiedades que se quieran para un mismo selector entre sus corchetes separando las mismas con un punto y coma. Así pues, para configurar un párrafo de color verde y con alineación justificada, la sintaxis quedaría de la siguiente forma:

p {color:green; text-align:justify}

Si se quisiera añadir el estilo definido anteriormente a la etiqueta <b>, no sería necesario volver a escribir toda la declaración, bastaría con separar con una coma un selector de otro, por ejemplo:

b, p {color:green; text-align:justify}

Ahora supóngase que, además, la etiqueta <b> debe aparecer sobre un fondo blanco. En este caso, la solución pasa por añadir una línea, ya que es una propiedad que se quiere aplicar sólo al selector <b>, así, el estilo tomaría la siguiente forma:

b, p {color:green; text-align:justify}
b {background-color:white}

Ejemplo:



```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
<style type="text/css">
body {
background-color:#33ff66;
background-position:0px;
cursor:hand;
}
h1 {color:#205596; background-color:#ffcc00;}
a {color:#303030; font-style:italic;}
</style>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<p><a href="enlace">Enlace de texto</a></p>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

Otra posibilidad puede ser la de establecer un estilo dependiendo de una condición estructural. Por ejemplo, si el texto en negritas aparece dentro de la celda de una tabla, entonces el fondo debe ser negro y la letra debe aparecer en cursiva, ejemplo:

```
td b {background-color:black; font-style:italic;}
```

Obsérvese cómo en este caso no se han separado los selectores con una coma. De esta forma, el intérprete de CSS entiende que esta declaración sólo debe cumplirse para las negritas que estén dentro de una celda.

Finalmente, se puede aplicar un estilo concreto un elemento del documento web, bien sea porque de manera puntual se quiere modificar el estilo definido en anteriores declaraciones en el supuesto de que las hubiera o porque sólo se quiere que ese elemento modifique sus propiedades de estilo sin afectar a sus semejantes.

Así pues, volviendo al selector de negritas <b>, se va a indicar en un punto determinado del código un estilo concreto, ejemplo:

```
<p>El texto en negritas en este punto <b style="font-size:12pt; color:#0000ff">es más grande y está en azul.</b>, por ejemplo</p>
```

### Definición de estilos mediante clases

La utilización de clases va a permitir especificar distintos estilos para un mismo elemento o bien generar estilos tipo para poder aplicarlos a cualquier etiqueta HTML según sea necesario. Su sintaxis es:

#### Visión y Pasión por Emprender y Servir

Elemento.nombredeclase {propiedad:valor}

Así pues, a modo de ejemplo se van a definir varios colores de párrafo de la siguiente manera:

```
p.amarillo {color:yellow}
p.verde {color:green}
p.colorextraño {color:#ca36e1}
```

Si se quieren aplicar los estilos de color definidos anteriormente, el código será similar al siguiente:

```
<p>Este párrafo no tiene ningún estilo aplicado</p>
<p class="amarillo">Este saldrá de color amarillo</p>
<p class="verde">Este seguro que verde</p>
<p class="colorextraño">El color extraño se parece al violeta</p>
```

Ahora, si se quisiera determinar un estilo de color para todos los encabezados, no se tendría que repetir la propiedad de color para cada encabezado, bastaría con definir una clase sin asociarla a ningún selector.

Ejemplo:

```
.amarillo {color:yellow}
.verde {color:green}
.colorextraño {color:#ca36e1}
```

Como se puede ver, la sintaxis es la misma, salvo que, en este caso, se está omitiendo el selector, con lo que se está dejando abierta la posibilidad de aplicarlo a cualquier selector.

Ejemplo:

```
<p class="amarillo">Este saldrá de color amarillo</p>
<h1 class="verde">Este encabezado saldrá de color verde</h1>
<b class="colorextraño">Este texto en negritas es de color ...</b>
```

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza, Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
<style type="text/css">
.cuerpo {background-color:#0080ff;}
.estilo1 {
color:#000099;
text-align:center;
font-weight:bold;
font-size:12pt;
font-family:Comic Sans MS;
background-color:#cc99ff;
}
.izquierda {text-align:left; font-weight:bold;}
.derecha {text-align:right}
</style>
```



```
</head>
<body class="cuerpo">
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<p class="estilo1">Texto con propiedades definidas en el estilo 1.</p>
<p class="izquierda">Texto que aparece alineado a la izquierda y en negritas.</p>
<p class="derecha">Texto que aparece alineado a la derecha.</p>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

### Definición de estilos con id

A diferencia del atributo class, los selectores id sólo permiten su utilización en un elemento, es decir, si hay un elemento que tiene asignado el atributo id="centrado" no podrá haber otro con el mismo nombre. La identificación del selector es case sensitive, es decir, es capaz de diferenciar o distinguir las mayúsculas de las minúsculas, de tal forma que, centrado, Centrado y CENTRADO, son identificadores diferentes.

Su sintaxis es la siguiente:

```
#idelemento {propiedad:valor}
```

Volviendo a los estilos de párrafo, supóngase que los párrafos, además de color verde, tienen que estar en negritas, ejemplo:

```
#parrafoespecial {font-weight:bold}
```

Así pues, el código HTML quedaría de la siguiente forma:

```
<p class="verde">Línea de color verde</p>
<p class="verde" id="parrafoespecial">este, además de verde, está en negritas</p>
```

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
<style type="text/css">
#general {margin-left:150px; margin-right:150px; background-color:#ffff99;}
#titulo1 {font-weight:bold; color:#333ccc; font-size:12pt; text-align:justify; width:250px;}
#titulo2 {color:yellow; font-size:12pt; text-align:justify; background:#669966; width:250px;}
</style>
</head>
<body id="general">
```

## Visión y Pasión por Emprender y Servir

```
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>
```

```
<!-- Contenido del documento -->
```

```
<p id="titulo1">
Una tarde de verano Mamá; Pata se hallaba ocupada en la tarea de empollar sus huevos,
cosa que la aburría.
... Al atardecer llegó; a una pequeña cabaña de
labradores, tan ruinosa, que parecía
que se iba a derrumbar de un momento a otro. La tormenta obligó;
al patito agacharse
en el suelo, y entonces advirtió; que la ...</p>
```

```
<h1 id="titulo2">
El patito reconoció; a los maravillosos cisnes, y lleno de una tristeza se dijo:
Me gustaría; dirigirme hacia esas aves tan majestuosas. Pero siendo feo como soy,
si me acerco mucho a ellos me matarán. Quizá; sea mejor
así; prefiero que me maten
ellos a que me picoteen los patos,
me peguen las gallinas y la muchacha de la granja me eche a
puntapiés.
</h1>
```

```
<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

## La regla @import

La regla @import permite importar, CSS desde otras CSS. Es importante destacar que esta regla debe de preceder a cualquier otra especificada en la CSS, ejemplo:

```
@import "estilo.css";
@import url(estilo.css);
```

## Las etiquetas &lt;div&gt; y &lt;span&gt;

Las etiquetas <div> y <span> permiten estructurar los documentos web.

<div> define un bloque de información, mientras que <span> define el contenido de un elemento dentro de un bloque, ya sea un párrafo o una línea. Ambos elementos por sí solos no dotan al contenido de ninguna característica especial, pero junto con las definiciones de estilo permiten crear bloques o elementos personalizados.

Al aplicar esta etiqueta <div> se añade un salto de línea tanto al comienzo como a final de la misma, por ejemplo:

```
<style type="text/css">
div.bloque1 {text-align:center; color:red}
span.sub1 {color:blue; text-decoration:underline;}
</style>
<div class="bloque1">
<h3>Texto centrado y con color rojo</h3>
<p><span class="sub1">Azul; subrayado y azul</span> para ser luego rojo otra vez.</p>
</div>
```



#### Visión y Pasión por Emprender y Servir

El uso de la etiqueta <span> está recomendado en aquellos casos en que se quiera modificar el estilo definido dentro de un bloque <div>.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
<style type="text/css">
body {background:#f5f5dc;}
.posicion1 {font-weight:bold; font-size:70px; line-height:80px; font-family:verdana;}
.posicion2 {margin-top:-60px; font-weight:bold; font-size:70px; line-height:90px; font-family:verdana;}
.posicion3 {margin-top:-70px; font-weight:bold; font-size:70px; line-height:100px; font-family:verdana;}
.posicion4 {margin-top:-80px; font-weight:bold; font-size:70px; line-height:100px; font-family:verdana;}
.rojo {color:#ff3300}
.azul {color:#0099ff}
.verde {color:#66cc33}
.amarillo {color:#ffcc00}
</style>
</head>
<body class="cuerpo">
<header>
<h1>CCAP México</h1>
<h1>Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<div>
<table width="560">
<tr>
<td>
<div class="posicion1">
<span class="rojo">8</span>
<span class="verde">1</span>
<span class="azul">7</span>
</div>
<div class="posicion2">
<span class="verde">5</span>
<span class="azul">6</span>
<span class="amarillo">4</span>
</div>
<div class="posicion3">
<span class="azul">1</span>
<span class="amarillo">4</span>
<span class="rojo">8</span>
</div>
<div class="posicion4">
<span class="amarillo">2</span>
<span class="rojo">7</span>
<span class="verde">1</span>
</div>
</td>
</tr>
</table>
</div>
```

```
</div>
</td>
</tr>
</table>
</div>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

#### Comentarios

Como todo lenguaje de programación, en códigos extensos se hace necesaria la inclusión de comentarios aclaratorios para futuras revisiones.

Para introducir comentarios en las CSS las notas deberán estar encerradas entre `/* y */`.

Ejemplo:

H1 {color:red} `/*cabecera de nivel uno en color rojo*/`

#### Unidades de medida y color

Las unidades de medida en CSS se forman con el signo + o con el signo – (opcional), seguido de un número y de una abreviación de dos letras que indica la unidad. las unidades de longitud pueden ser relativas y absolutas.

Las unidades relativas son:

- em Un em es igual al tamaño de la fuente del elemento actual.
- ex Un ex es x-altura de un conjunto de caracteres.
- px Son píxeles.

Las unidades absolutas son:

- in pulgadas.
- cm centímetros.
- mm milímetros.
- pt puntos (1 pt = 1/72 pulgadas).
- pc picas (1 pc = 12 pt).

También se pueden establecer valores de porcentaje mediante el símbolo %.

Estos valores son relativos a otros según su definición en la propiedad. Normalmente, los porcentajes son relativos al tamaño de fuente del elemento.

Para definir colores, se debe utilizar la correspondiente palabra del color o especificar el color en el modelo RGB numéricamente.

Las palabras básicas disponibles, en inglés, son: black, teal, blue, navy, lime, white, purple, yellow, olive, red, marron, gray, fuchsia, green, silver, aqua, etc.

Los colores RGB se pueden formar de cuatro maneras distintas:

- #rrggbb #11cc45, #55ff0b
- #rgb #0c0, #fcd
- rgb(n,n,n) n es un número de entre 0 y 255 rgb(56,40,210)
- rgb(%,%,% ) porcentajes de color rgb(10%,80%,20%)

#### Visión y Pasión por Emprender y Servir

#### Propiedades de las Hojas de Estilo en Cascada

La definición de los estilos de CSS junto con las características propias que cada navegador implementa, pone a disposición de los webmasters una multitud de propiedades

#### Propiedades de las fuentes

##### font-family

Determina la fuente que se va a utilizar o la familia genérica. Una fuente específica será más difícil de encontrar que una familia de fuente genérica. Así pues, si se hace una asignación de fuente específica, es recomendable acompañarla por un nombre de familia genérica en el caso de que la primera selección de la fuente no se encuentre.

Ejemplo:

```
font-family:Courier
```

```
font-family:New Century Schoolbook, Times, Serif
```

##### font-style

Convierte el texto en cursiva. Las opciones son italic, oblique y normal.

Ejemplo:

```
font-style:italic
```

##### font-variant

Convierte las minúsculas en pequeñas mayúsculas. Las opciones son: small-caps y normal.

Ejemplo:

```
font-variant:small-caps
```

##### font-weight

Indica el grosor de una fuente. Los valores son normal, bold, bolder, lighter o las centenas comprendidas entre 100 y 900.

Ejemplo:

```
font-weight:700
```

```
font-weight:bold
```

##### font-size

Determina el tamaño de la fuente. Los valores pueden ser absolutos (xx-small, xs-small, small, medium, large, x-large, xx-large), relativos (larger o smaller) o porcentaje.

Ejemplo:

```
font-size:100%
```

```
font-size:20pt
```

```
font-size:large
```

##### font

Determina todas las propiedades anteriores en una sola.

Ejemplo:

```
font {font-style:oblique; font-variant:small-caps; font-weight:lighter;}
```

#### Propiedades del texto

##### word-spacing

Determina el espaciado entre palabras. El valor debe estar en formato de longitud.

#### Visión y Pasión por Emprender y Servir

Ejemplo:

word-spacing:4pt

#### letter-spacing

Determina el espaciado entre caracteres. El valor debe estar en formato de longitud un ajuste de 0 evitará justificación.

Ejemplo:

letter-spacing:1px

#### text-decoration

Muestra el texto con distintos efectos. Pueden ser none, underline, overline, line-through y blink.

Ejemplo:

text-decoration:underline

#### vertical-align

Determina la alineación vertical. El valor puede ser un porcentaje, una palabra clave como baseline, middle, sub, super, text-top, text-bottom, top, bottom.

vertical-align:super

vertical-align:50%

#### text-transform

Especifica más efectos de texto. Las opciones son: capitalize, uppercase, lowercase, none.

Ejemplo:

text-transform:uppercase

#### text-align

Establece la alineación del texto. Las opciones son left, right, center y justify.

Ejemplo:

text-align:justify

#### text-indent

Permite establecer tabulaciones, es decir, la cantidad de sangría que debe aplicarse a la primera línea. El valor debe ser una longitud o un porcentaje.

Ejemplo:

text-indent:10px

text-indent:5em

#### line-height

Permite asignar un valor para controlar el espacio entre líneas base de texto. Si el valor es un número, la altura de línea se calcula multiplicando el tamaño de la fuente del elemento por el número.

Los valores en porcentaje son relativos al tamaño de fuente del elemento.

No se permiten valores negativos. Los valores pueden ser normal, una distancia, un porcentaje o un valor numérico.

Ejemplo:

line-height:12pt

line-height:100%

#### Visión y Pasión por Emprender y Servir

##### Propiedades del fondo y del color

###### color

Determina el color de un elemento.

Ejemplo:

color:red

###### background-color

Determina el color del fondo de un elemento.

background-color:white

background-color:#000080

###### background-image

Determina la imagen que va a servir de fondo

Ejemplo:

background-image:textura.gif

background-image:url(/imágenes/foto.jpg)

###### background-repeat

Determina si una imagen de fondo será repetida o no. Los valores disponibles son: repeat, repeat-x, repeat-y y no-repeat.

El valor repeat-x repetirá la imagen horizontalmente, mientras que el valor repeat-y repetirá la imagen verticalmente.

Ejemplo:

background-repeat:no-repeat

background-repeat:repeat-x; background-image:url(formas.jpg)

###### background-attachment

Determina si una imagen de fondo debe moverse a la par que el scroll de la página. Los valores disponibles son: scroll y fixed.

Ejemplo:

background-attachment:scroll

###### background-position

Determina la posición de inicio del background, ya sea color o imagen. Los valores disponibles son porcentaje, longitud, top, center, bottom, left, center, right. También pueden usarse porcentajes y longitudes para asignar la posición de la imagen de fondo.

Los porcentajes son relativos al tamaño del elemento.

Ejemplo:

background-position:20% 65%

###### background

Permite especificar las propiedades del fondo. Los valores disponibles son transparent, color, URL, repeat, scroll y position.

Ejemplo:

background:transparent

##### Propiedades de las ventanas



#### Visión y Pasión por Emprender y Servir

##### margin-top, margin-right, margin-bottom, margin-left

Determina los márgenes superior, derecho, inferior e izquierdo del documento web respectivamente. Los valores pueden ser longitud, porcentaje y auto.

Ejemplo:

```
margin-top:25px
margin-right:50%
margin:1em 2em 3em 4em
margin-top:2.5em
```

##### padding-top, padding-right, padding-bottom, padding-left

Permite especificar el espacio superior, inferior, derecho e izquierdo entre el borde y el contenido de un elemento. Los valores pueden ser longitud o porcentaje.

Ejemplo:

```
padding-top:15px
padding-right:50%
padding-bottom:2in
padding-left:25px
```

##### padding

Permite especificar el espacio entre el borde y el contenido de un elemento. Los valores pueden ser longitud o porcentaje. Se pueden especificar entre uno y cuatro valores. Si se dan cuatro valores, éstos se aplican a los paddings superior, derecho, inferior e izquierdo, respectivamente. Si sólo se da un valor, se aplica a todos los lados si se indican dos o tres valores, los valores de padding restantes se toman del lado opuesto.

Ejemplo:

```
padding:80px
padding:2em 4em 5em
```

##### border-top-width, border-right-width, border-bottom-width, border-left-width, border-width

Determinan el ancho del borde de un elemento. Los valores para el ancho son thin, médium y thick o una longitud.

Ejemplo:

```
border-bottom-width:médium
```

##### border-color

Determina el color del borde de un elemento. Se combina con la propiedad border-style. Se pueden especificar entre uno y cuatro valores. Si se dan cuatro valores, éstos se aplican a los colores de borde superior, derecho, inferior e izquierdo, respectivamente. Si sólo se da un valor, se aplica a todos los bordes y si sólo se indican dos o tres valores, los valores de color de borde restantes se toman del lado opuesto.

Ejemplo:

```
border-color:red; border-style:double
```

##### border-style

Determina el estilo de un borde. Se combina con la propiedad anterior. Los valores del estilo son: none, dotted, dashed, solid, double, groove, ridge, inset y outset.

Ejemplo:

```
border-style:dotted; border-color:blue
```

##### border-top, border-right, border-bottom, border-left, border

Determina el ancho, el estilo y el color de los bordes de un elemento. Los valores para el ancho son: thin, medium y thick. Los valores del estilo son: none, dotted, dashed, solid, double, groove, ridge, inset, outset.





---

Visión y Pasión por Emprender y Servir

Ejemplo:

`border-top:10px solid black`

**width**

Determina el ancho de un documento. Los valores pueden ser longitud, porcentaje o auto.

`width:50%`

`width:10em`

**height**

Determina la altura del elemento. Los valores pueden ser longitud o auto.

`height:100px`

**float**

Permite indicar si una imagen será flotante o no para poder ajustar texto alrededor. Las opciones son: none, left y right.

Ejemplo:

`float:left`

**clear**

Determina los lados en los que no se podrán situar elementos flotantes. Los valores son: none, left, right y both. Un valor left mueve el elemento debajo de cualquier elemento flotante a su izquierda; right actúa en forma similar para elementos flotantes a la derecha. Otros valores son none, que es el valor inicial y both que mueve el elemento debajo de los elementos flotantes ambos lados.

Ejemplo:

`clear:right`

**Propiedades de clasificación**

**display**

Modo de mostrar la información en un bloque. Los valores son: block, inline, list-item y none.

Ejemplo:

`display:none`

**white-space**

Determina cómo se tratan los espacios dentro del elemento. Los valores pueden ser: normal, pre o nowrap.

Ejemplo:

`white-space:normal`

**list-style-type**

Especifica el tipo de marcador de ítem de lista a mostrar. Los valores son: disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha y none.

Ejemplo:

`list-style-type:decimal`

**list-style-image**

Especifica la imagen que se usa como marcador de ítem de lista cuando se activa la carga de imágenes reemplazando al marcador especificado en la propiedad list-style-type.



Ejemplo:

list-style-image:url(/mano.gif)

#### list-style-position

Determina dónde se coloca el marcador con relación al ítem de lista. Los valores disponibles son inside y outside. Si se usa inside, las líneas se ajustarán debajo del marcador en vez de estar con sangría.

Ejemplo:

list-style-position:inside

#### list-style

Especifica de manera rápida las propiedades list-style vistas anteriormente.

Ejemplo:

list-style:url(/flecha.jpg) circle

#### Ejemplos de aplicación

A continuación, se presentan algunos ejemplos con las propiedades vistas anteriormente.

En el primero de ellos se va a utilizar la propiedad hover similar al evento onmouseover de JavaScript.

El objetivo del programa es conseguir que el texto que describe un enlace cambie de color y de formato cuando el cursor del ratón se sitúe sobre él. También se hará que el cursor se transforme en una mano como forma predeterminada.

Ejemplo 1

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>

<style type="text/css">
body {color:#ff9933; cursor:hand; background-color:"ffffff; }
a:hover {color:blue; text-transform:uppercase;}
</style>

</head>
<body class="cuerpo">
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<p>
<a href="www.carlosys.com" style="color:green">Esta es mi web</a><br>
```

**Visión y Pasión por Emprender y Servir**

```
<a href="www.e-oros.com" style="color:#ff8c00;">Esta es m&acute;s interesante</a><br>
<a href="www.ra-ma.es" style="color:#ffcc00;">Editorial Ra-Ma</a><br>
<a href="www.google.com" style="color:#ff8c00;">Soy un buscador</a>
</p>
```

```
<footer>
<p>El contenido de este ejemplo no est protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

**Ejemplo 2**

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboracin Bsica de Pginas Web">
<meta name="keywords" content="curso, web, pgina, html, css, javascript, ccap, Mxico, js, programacin, html5, script, Neza,
Nezahualcyotl"/>
<meta name="author" content="Fernando Jimnez vila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content=" Fernando Jimnez vila"/>
<title>Elaboracin Bsica de Pginas Web ..... CCAP Mxico</title>

<style type="text/css">
ol.roman {list-style-type:lower-roman}
ol.uroman {list-style-type:upper-roman}
ol.ualpha {list-style-type:upper-alpha}
ol.dibu {list-style-image:url(lista.gif)}
</style>

</head>
<body class="cuerpo">
<header>
<h1>CCAP Mxico</h1>
<h1>Bienvenido al Mundo del HTML5</h1>
<h2> No te asustes, no es tan difcil!</h2>
</header>

<!-- Contenido del documento -->

<ol class="lroman">
<li>&Aacute;ste es el primer elemento</li>
<li>&Aacute;ste es el segundo elemento</li>
</ol>
<ol class="uroman">
<li>&Aacute;ste es el primer elemento</li>
<li>&Aacute;ste es el segundo elemento</li>
</ol>
<ol class="ualpha">
<li>&Aacute;ste es el primer elemento</li>
<li>&Aacute;ste es el segundo elemento</li>
</ol>
<ol class="dibu">
<li>&Aacute;ste es el primer elemento</li>
<li>&Aacute;ste es el segundo elemento</li>
</ol>
```

```
<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

### Introducción a JavaScript

JavaScript es un lenguaje de programación creado con el objetivo de integrarse en HTML y facilitar la creación de páginas interactivas sin necesidad de utilizar códigos o scripts de CGI o Java.

No hay que confundir Java con JavaScript. Java es un lenguaje completo que permite crear aplicaciones independientes, mientras que JavaScript es un lenguaje que funciona como extensión del HTML. JavaScript es un lenguaje de programación orientado a objetos, diseñado para el desarrollo de aplicaciones cliente-servidor a través del Internet.

El código de programa de JavaScript, llamado script, se introduce directamente en el documento HTML y no necesita ser compilado, es el propio navegador el que se encarga de traducir o interpretar dicho código.

Gracias a JavaScript se pueden desarrollar programas que se ejecuten directamente en el navegador, que es el cliente, de manera que el navegador o el cliente pueda efectuar determinadas operaciones o tomar decisiones sin necesidad de acceder al servidor. Por ejemplo, al desarrollar un programa que verifique una clave de acceso para poder acceder a una determinada web, JavaScript debe comprobar la información dada por el usuario, verificar que sea correcta y actuar en consecuencia.

Recuérdese que, aunque JavaScript se parece a Java, no es lo mismo; por ejemplo, Java es un lenguaje de programación algo complejo a diferencia de JavaScript que es bastante más sencillo. Otra diferencia notable es que los programas realizados en Java deben ser compilados en un archivo ejecutable, mientras que, con JavaScript, no.

Cabe destacar que no todos los navegadores soportan JavaScript, e incluso ni totalidad de las distintas versiones de JavaScript.

Aunque en la actualidad se podría afirmar que casi la totalidad de los navegadores actuales admiten JavaScript, hay que tener en cuenta que algunos usuarios pueden bloquear la ejecución o inhabilitar el soporte JavaScript en su navegador.

### JavaScript y HTML

Un programa en JavaScript, como ya se comentó, no puede ser compilado, ya que el navegador se encarga de leerlo, o interpretarlo, cuando se carga la página, pero esto no quiere decir que todos los códigos de JavaScript se ejecuten nada más cargarse el documento HTML, algunas funciones pueden permanecer en estado latente, o a la espera de que el usuario realice una acción determinada o se dé una condición establecida, como puede ser el pulsar o dar clic sobre un enlace.

Para definir el inicio de un script de JavaScript se deben utilizar las etiquetas `<script>` y `</script>`, ejemplo:

```
<script language="JavaScript" type="text/javascript">
```

El atributo language indica al navegador el lenguaje script utilizado que, en este caso, debe ser JavaScript. Si se omite la especificación del lenguaje, el navegador asumirá que el script está programado en JavaScript.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza, Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
```



Visión y Pasión por Emprender y Servir

```
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<script language="javascript" type="text/javascript">
document.write("Mi primer programa con JavaScript");
</script>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

El código anterior muestra en pantalla una línea de texto con la frase Mi primer programa con JavaScript.

Los códigos de JavaScript se pueden colocar en la cabecera o en el cuerpo del documento web. A medida que se vaya avanzando, se irá encontrando las ubicaciones idóneas para ellos.

Es muy importante tener siempre presente que JavaScript es un lenguaje de programación case sensitive, es decir, que se distingue las mayúsculas de las minúsculas, por lo consiguiente, habrá que prestar especial atención a la utilización de las variables, así pues, las variables resultado, resultado y RESULTADO serán vistas por JavaScript como variables diferentes.

También tener en cuenta una cuestión sumamente importante, y es que no todos los navegadores pueden soportar JavaScript o, si bien, lo soportan, el usuario ha desactivado o bloqueado su intérprete mediante las opciones de configuración del mismo.

Cuando el usuario utiliza un navegador que no soporta JavaScript, las etiquetas <script> y </script> son ignoradas y se presenta el código script en pantalla al entender que se trata de texto. La solución pasa por añadir dos comentarios al inicio y al final de script, ejemplo:

```
<!--Ocultar para navegadores antiguos
...
...
//fin del camuflaje -->
```

Se disponen también de las etiquetas <noscript> y </noscript>. Entre ellas se especificará una opción alternativa para los navegadores que no soportan scripts. Por ejemplo, un texto diciendo Su navegado no soporta JavaScript o cualquier otra secuencia de comandos alternativos.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
```



```
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<script language="javascript" type="text/javascript">
<!--Aquí camuflamos el código de JavaScript para navegadores antiguos
document.write("Está de suerte, su navegador soporta JavaScript);
//fin del camufale -->
</script>
<noscript>
Su navegador no soporta JavaScript.
</noscript>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

### Introducir comentarios

Como todo lenguaje de programación, en códigos extensos se hace necesaria la inclusión de comentarios aclaratorios para futuras revisiones.

Para introducir comentarios en JavaScript se pueden utilizar dos alternativas diferentes:

- Los comentarios en una sola línea, que irán precedidos de //.
- Los comentarios de varias líneas, que irán encerrados entre /\* y \*/.

Ejemplo:

```
<script>
//Comentario de una sola línea, la siguiente se ejecuta

document.write("Bienvenidos a JavaScript")

//Nuevamente un comentario de una sola línea
//*****
//Comentarios, Comentarios
//*****

/*
Ésta es otra forma de introducir comentarios pero
utilizando muchas líneas, incluso cosas como

document.write("Hola")

son vistas como un comentario
*/
</script>
```

Por muy fácil que parezca un programa, añadir comentarios puede resultar de mucha ayuda cuando transcurrido cierto tiempo se tiene de nuevo la necesidad de revisar el código.

## Corchetes

Los corchetes se utilizan para definir fragmentos de código de manera que éstos no se junten con el resto del programa. Por ejemplo, todo el código que ejecuta una función debe ir bien diferenciado del resto mediante corchetes, ejemplo:

```
<script language="JavaScript" type="text/javascript">
//Definición de la función
for (cuenta = 1; cuenta < 4; cuenta++)
//Inicio del código de la función
{
alert("Ventana de diálogo de prueba\nAparezco " + cuenta)
}
//Fin del código de la función
</script>
```

## El punto y coma

La misión del punto y coma (;) en JavaScript es separar las sentencias que se encuentran en una misma línea. También puede indicar el final de una sentencia que ocupa varias líneas.

Ejemplo:

```
<script language="JavaScript" type="text/javascript">
var texto = "FAQ (Frequently Asked Questions, Documentos de Preguntas Frecuentes). Documentos recopilatorios para usuarios principiantes, que contienen las preguntas más frecuentes sobre un tema determinado. Son la mejor forma de comenzar cosas sobre HTML, CSS y JavaScript";
</script>
```

## Archivos fuente

Hasta ahora se ha visto cómo introducir el código JavaScript dentro de un documento HTML; sin embargo, se puede manejar un script externo mediante el atributo src aplicado a la etiqueta <script>.

La utilización de códigos JavaScript en archivos externos es la mejor opción para trabajar con HTML, siempre que sea posible. El disponer en un archivo externo de todo el código de la página, permite que sea reutilizable en otros documentos.

De esta forma se puede disponer de una librería o biblioteca de scripts desvinculados totalmente del código de un documento HTML y facilitar ampliamente la reutilización de los mismos.

El script debe estar en un archivo de texto sin las etiquetas de comienzo y del final del mismo, <script> y </script>. Además, éste deberá tener la extensión .js.

A continuación, el contenido de un script que obtiene la fecha local, que se vinculará posteriormente desde un documento HTML:

```
//Ejemplo de script externo llamado fecha.js
var hoy = new Date();
dia = hoy.getDate();
mes = hoy.getMonth()+1,
año = hoy.getYear();
año = hoy.getFullYear();
alert("Hoy es " + dia + " del " + mes + " de " + año);
```

El fragmento de código JavaScript anterior será grabado como un archivo de texto plano sin formato con el nombre de fecha.js.

Ahora, se verá el código que se encargará de llamar a ese script:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
```

**Visión y Pasión por Emprender y Servir**

```
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<script language="JavaScript" type="text/javascript" src="fecha.js">
</script>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

Obsérvese cómo en el listado anterior, el código de JavaScript no aparece, sino que, en su defecto, se encuentra una llamada a un archivo llamado fecha.js.

**Tratamientos de variables**

Todos los lenguajes de programación necesitan en algún momento cargar en memoria los datos que se van a procesar. Por ejemplo, solicitar o pedir al usuario su nombre y almacenarlo en una variable nombreusuario o guardar el resultado de algún cálculo en la variable resultado, por ejemplo, guardar el valor obtenido al multiplicar dos números cualesquiera. Como en todo lenguaje de programación, las variables son fundamentales.

**Las variables**

JavaScript admite prácticamente cualquier tipo de nombre para definir a una variable, no obstante, hay una serie de consideraciones que se deben tener presentes:

1. El primer carácter debe ser siempre una letra o el guión bajo o de subrayado (\_). Los restantes caracteres pueden ser letras, número o el guión bajo, teniendo como precaución no dejar espacios en blanco entre ellos.
2. El nombre de la variable no debe coincidir o ser igual a las palabras reservadas de JavaScript.
3. Recordar que JavaScript es case sensitive, es decir, diferencia mayúsculas y minúsculas.

Para declarar variables se utiliza la palabra clave o reservada var seguida del nombre de la variable. Así pues, según lo explicado anteriormente, las siguientes variables serán reconocidas como tales:

```
var nombre;
var Direccion;
var entrada_valor_concreto;
var variable_numero_12;
```

A continuación, se muestran otras variables que no serán reconocidas por JavaScript al no cumplir algunas de las reglas de definición vistas anteriormente:

```
var 1dato;
var entrada datos;
```



```
var while;  
var new;
```

Se recomienda utilizar siempre la misma pauta o notación para definir los nombres de las variables. Puede escribirse todo el nombre en minúsculas o bien la primera letra en mayúscula y las siguientes en minúsculas, conocida como notación de camello, ejemplos:

```
var Nombre;  
var ApellidoMaterno;  
var ImpuestoSobrelaRenta
```

Es muy importante este punto, ya que en scripts complejos o extensos pueden producirse errores difíciles de localizar.

Aunque las siguientes variables parezcan iguales, JavaScript las interpretará como diferentes variables:

```
var resultadosuma;  
var Resultadosuma  
var resultadoSuma  
var ResultadoSuma;  
var RESULTADOSUMA  
var ResultadosumA
```

A pesar de que las variables declaradas anteriormente pueden representar el mismo concepto o la misma idea y en consecuencia un mismo valor, para JavaScript son seis variables diferentes para almacenar distintos valores.

#### Tipos de variables

JavaScript puede manejar varios tipos de datos distintos decidiendo el tipo de variable que deberá empelar durante la ejecución del script, es decir, define qué tipo de valor almacenará la variable.

Los cuatro tipos de variables que puede almacenar JavaScript son:

- Variables de cadena, alfanumérica o de texto.
- Variables numéricas.
- Variables lógicas o booleanas.
- Variables indefinidas (undefined).

Un quinto tipo de variable podrían ser los datos nulos (null). Los valores nulos se utilizan para comprobar si a una variable se le ha asignado un valor o no. Así, null representa un valor nulo para cualquier tipo de variable; por el contrario, una variable que no ha sido inicializada tiene un valor indefinido o NaN en un contexto numérico.

#### Variables de cadena

Una variable de cadena, alfanumérica o de texto, es aquella que contiene texto. Las cadenas de texto en JavaScript se delimitan con comillas dobles o simples y pueden contener cualquier tipo de carácter. Por supuesto, también pueden tener un valor nulo.

Ejemplos:

```
var País = "México"  
var EntradaCodigo;  
EntradaCodigo = "57000";  
var Valor = " ";  
var Dato = "descuento";
```

Como norma general, las comillas simples serán utilizadas dentro de fragmentos de código delimitados por comillas dobles o viceversa, ejemplo:

```
document.write("Qué quiere decir la palabra 'nuncupatorio'");  
alert('Pulsa la tecla "amigo"');  
alert("No sé qué significa");
```

**Visión y Pasión por Emprender y Servir**

Hay una serie de caracteres que permiten representar caracteres especiales, como podrían ser las comillas, un salto de línea en un texto, a la cual se le conoce como secuencia de escape:

- |            |  |
|------------|--|
| 1. \b      | carácter anterior  |
| 2. \f      | salto de página  |
| 3. \n      | salto de línea   |
| 4. \r      | retorno de carro   |
| 5. \t      | tabulador  |
| 6. \\      | carácter \   |
| 7. \'      | comilla simple   |
| 8. \"      | comilla doble  |
| 9. \XXX    | caracteres con codificación latín-1 representados por tres dígitos en octal      |
| 10. \xXX   | caracteres con codificación latín-1 representados por dos dígitos en hexadecimal |
| 11. \uXXXX | caracteres Unicode especificados por cuatro dígitos en hexadecimal               |

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1>Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<script language="JavaScript" type="text/javascript">
var Cadena1 = "Esto es una cadena de texto que no utiliza ningún carácter especial";
var Cadena2 = "\n\nEsto es una cadena \nde texto que sí utiliza\ncaracteres especiales";
var Cadena3 = "\n\n251 Fernando Jiménez Ávila\xA9 CCAP México";
var Cadena4 = "\n\u00AE por \rFer", 1997-2024";
alert(Cadena1 + Cadena2 + Cadena3 + Cadena4)
</script>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

**Variables numéricas**

Las variables numéricas son aquellas variables que contienen números enteros o decimales, fraccionarios o de coma o punto flotante.



#### Enteros

JavaScript puede utilizar la base decimal (base 10), la hexadecimal (base 16) y la octal (base 8). Por defecto, el sistema es el decimal.

```
var numero;  
numero = 100;  
numero = -1000;
```

Para especificar un número en base octal, se deberá anteponer un cero al número, cabe hacer mención que los números en base octal sólo pueden contener los dígitos del 0 al 7, ejemplos:

```
var NumeroOctal;  
NumeroOctal = 03254;  
NumeroOctal = 02;
```

Para especificar un valor en base hexadecimal, se debe anteponer al número el prefijo 0x, cabe hacer mención que los números en hexadecimal incluyen los dígitos del 0 al 9 y las letras comprendidas entre la A y la F, ejemplos:

```
var NumeroHexadecimal;  
NumeroHexadecimal = 0xff;  
NumeroHexadecimal = 0x12f;
```

#### Coma flotante

Un valor de coma flotante, que también es conocido como de punto flotante, se compone de un valor entero seguido de un punto y de decimales o de una fracción decimal. El exponente se indica mediante una e o E seguida por un entero positivo o negativo, ejemplos:

```
var NumeroComaFlotante;  
NumeroComaFlotante = 10.236;  
NumeroComaFlotante = 43.433e+2;  
NumeroComaFlotante = -56.25;
```

#### Variables booleanas

Las variables booleanas o lógicas se especifican mediante los valores verdadero (true) o falso (false), ejemplos:

```
var EstoyContento;  
EstoyContento = false;  
EstoyContento = true;
```

La utilización de los tipos booleanos es de suma importancia cuando se tienen que comparar datos o tomar decisiones.

#### Variables undefined

Las variables undefined corresponden a variables que han sido creadas pero que no tienen ningún valor asignado.

Mediante este sencillo ejemplo de script, se podrá conocer si una variable tiene o no asignada un valor:

```
<!DOCTYPE html>  
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>  
<meta http-equiv="Content-Language" content="ES-ES"/>  
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">  
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza, Nezahualcóyotl"/>  
<meta name="author" content="Fernando Jiménez Ávila"/>  
<meta name="distribution" content="Global"/>  
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
```



```
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<script language="JavaScript" type="text/javascript">
var CURP;
if (CURP == undefined)
{
document.write("La variable 'CURP' no está definida");
}
</script>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

### Conversión de tipos de datos

JavaScript no necesita que se declare especialmente el tipo de variable, ya que en función de las operaciones que debe realizar el tipo irá cambiando automáticamente.

La conversión se realiza de izquierda a derecha. Es decir, el operando de la derecha será convertido al tipo del operando de la izquierda.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<script language="JavaScript" type="text/javascript">
var Variable1 = "50"; //valor de cadena
var Variable2 = 10; //valor numérico entero
ResultadoUno = Variable1 + Variable2;
```

**Visión y Pasión por Emprender y Servir**

```
ResultadoDos = Variable1 + Variable2;
alert("Conversión variable numérica a cadena. \nEl resultado es " + ResultadoUno);
alert("Conversión variable cadena a numérica. \nEl resultado es " + ResultadoDos);
</script>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

En el script anterior, ResultadoUno tendrá un valor de "5010", porque con el operador +, JavaScript convierte los valores numéricos en cadenas realizando la conversión de izquierda a derecha. Es decir, el operando de la derecha, el valor numérico entero, será convertido al tipo del operando de la izquierda, el valor de cadena de texto.

Por el contrario, ResultadoDos tendrá un valor de 40, ya que, con otros operadores, por ejemplo, el de la sustracción o resta, los valores numéricos no pueden convertirse en cadenas de texto.

**Constantes**

Las constantes son variables que no cambian su valor, es decir, que siempre tienen el mismo valor, por ejemplo:

```
const Prefijo = "977";
const UMA = 107.57;
const x = 90;
const IVA = 0.16;
```

Para indicar a JavaScript que se trata de una constante y no de una variable, debe anteponerse la palabra clave o reservada const al nombre de la variable en la declaración; en caso contrario, JavaScript lo tratará como una variable.

Una constante no cambia nunca de valor y no puede ser redefinida en un mismo script.

**Localización de las variables**

Las variables pueden ser globales y locales según el lugar en que se hayan definido. Las primeras pueden usarse en cualquier parte del código, mientras que las segundas sólo pueden hacerlo dentro de la función que las define o declara.

Aunque no es obligatorio, se deben declarar las variables antes de utilizarlas. Cuando se utiliza var se está haciendo que la variable sea local al ámbito donde se ha declarado. Por el contrario, si no se utiliza var para declarar una variable, ésta será global a toda la página web, sea cual sea el ámbito en el que haya sido declarada.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza, Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
```

#### Visión y Pasión por Emprender y Servir

```
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>
```

```
<!-- Contenido del documento -->
```

```
<script language="JavaScript" type="text/javascript">
//Definición de variable global
var VariableGlobal = "Esta cadena de texto se ha definido FUERA de una función. Es global.";
//Definición de una variable local
function ejemplo()
{
var VariableLocal = "Eta cadena de texto se ha definido DENTRO de una función. Es local.";
document.write(VariableLocal);
}
//Impresión en pantalla de la variable global
document.write(VariableGlobal + "<br>");
//Impresión en pantalla de la variable local
ejemplo();
</script>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

Si se intenta utilizar una variable local en un ámbito global, JavaScript mostrará un mensaje de error diciendo que la variable no está definida.

#### Palabras reservadas

Al igual que sucede con todos los lenguajes de programación, JavaScript tiene una serie de términos reservados, a los cuales se les conoce como palabras clave o palabras reservadas, que no pueden ser empleados para nombrar variables, funciones, objetos y métodos.

Algunas de las palabras reservadas de JavaScript son:

abstract	else	instanceof	super
boolean	extends	interface	switch
break	false	long	synchronized
byte	final	native	this
case	finally	new	throw
catch	float	null	throws
char	for	package	transient
class	function	private	true
const	goto	protected	try
continue	int	public	var
default	implements	return	val
do	input	short	while
double	in	static	with

#### Los operadores

Los operadores permiten combinar diferentes valores en una expresión., anteriormente, se analizó como trabajan las variables con los datos, ahora se verá cómo manipular dichas variables asignando, si es necesario, los resultados a nuevas variables.

#### Tipos de operadores

Los operadores de JavaScript pueden dividirse en varios grupos. Su clasificación es meramente funcional y se basa en la operación que realiza cada operador, así pues, los grupos son:

#### Visión y Pasión por Emprender y Servir

- Operadores aritméticos.
- Operadores lógicos.
- Operadores de comparación.
- Operadores condicionales.
- Operadores de asignación.
- Operadores especiales.

#### Operadores aritméticos

Los operadores aritméticos son los encargados de realizar las operaciones básicas como sumar, restar, multiplicar y dividir:

- Suma, +.
- Resta, -.
- Multiplicación, \*.
- División, /.
- Módulo, resto de división, %.
- Incremento, preincremento, postincremento, ++.
- Decremento, predecremento, postdecremento, --.
- Negación, -.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1>Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Ejemplo de operaciones aritméticas</h1>

<script language="JavaScript" type="text/javascript">
var Valor1, Valor2, Valor3;
var Suma, SumaNegada, Resta, Multiplicacion, Division, RestoDivision, Varios;
Valor1 = 80;
Valor2 = 60;
Valor3 = 15;
Suma = Valor1 + Valor2;
SumaNegada = -Suma;
Resta = Valor1 - Valor2;
Multiplicacion = Valor1 * Valor2;
Division = Valor1 / Valor2;
RestoDivision = Valor1 % valor2;
Varios = (Valor1 + Valor3) * Valor2;
```

### Visión y Pasión por Emprender y Servir

```
document.write("El resultado de la suma es: " + Suma + "<br>");
document.write("Ahora se niega la suma, es: " + SumaNegada + "<br>");
document.write("El resultado de la resta es: " + Resta + "<br>");
document.write("El resultado de la multiplicación es: " + Multiplicacion + "<br>");
document.write("El resultado de la división es: " + Division + "<br>");
document.write("El resultado del resto de división es: " + RestoDivision + "<br>");
document.write("El resultado de la variable Varios es: " + Varios + "<br>");
</script>
```

```
<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

En el script anterior se han definido las variables Valor1, Valor2 y Valor3 que son las encargadas de contener números con los que se van a realizar las operaciones aritméticas básicas para el ejemplo. Los resultados de dichas operaciones se guardan en una variable a la que se le ha dado el mismo nombre que el operador utilizado. Así pues, la variable que contiene el resultado de sumar dos números se llama Suma, y el resultado de la sustracción o resta se almacena en Resta y así sucesivamente.

Otro tipo de operadores aritméticos muy utilizados son los incrementales o decrementales que se aplican antes o después del operando. Su utilización es algo diferente a la de los operadores clásicos.

Ejemplo:

```
variable++ //Devuelve el valor de la variable y después incrementa su valor en uno
++variable //Aumenta el valor de la variable en uno y después devuelve su valor
variable-- //Devuelve el valor de la variable y después disminuye su valor en uno
--variable //Disminuye el valor de la variable en uno y después devuelve su valor
```

A continuación, un ejemplo de uso de estos operadores:

```
Pesos = Pesos + 1; //Aumenta el valor de Pesos en uno;
Pesos++; //Aumenta el valor de Pesos en 1
```

Como se puede observar, para obtener un resultado idéntico en la primera línea se ha tenido que escribir más.

El otro operador aritmético es la negación. Ejemplo:

```
Numero = 100;
Numero = -Numero;
```

En este ejemplo, el valor inicial de la variable Numero es 100 positivo, y después de aplicar o usar el operador de negación pasa a ser -100.

### Operadores lógicos

Los operadores lógicos o booleanos se emplean para que un programa tome una decisión en función de un cálculo lógico. Los valores que se obtienen son true (verdadero) o false (falso).

Los operadores son los siguientes:

- Suma lógica (&&). Se corresponde con y. Este operador devuelve un valor true cuando las dos condiciones comparadas se cumplen. En el supuesto de que una ellas sea falsa, el resultado será siempre falso.
- Producto lógico (||) se corresponde con o. Este operador devuelve true siempre que una de las dos condiciones sea verdadera. En caso contrario, devuelve falso.
- Negación (!). se corresponde con no. Este operador devuelve siempre el valor contrario, es decir, si la condición o variables es true, devuelve false y viceversa.

La verdadera utilidad de estos operadores se ve al trabajar con estructuras condicionales, de momento se toma como referencia el siguiente ejemplo:





#### Visión y Pasión por Emprender y Servir

//Ejemplos de sumas lógicas

```
Suma1 = true && true           //true
Suma2 = true && false          //false
Suma3 = false && true           //false
Suma4 = false && false          //false
Suma5 = false && (2==3)         //false
Suma6 = "negro" && "blanco"     //false → blanco
Suma7 = false && "negro"        //false
Suma8 = "cat" && false           //false
```

//Ejemplos de productos lógicos

```
Producto1 = true || true       //true
Producto2 = true || false      //true
Producto3 = false || true      //true
Producto4 = false || false     //false
Producto5 = false || (2==3)    //false
Producto6 = "negro" || "blanco" //true → negro
Producto7 = false || "blanco"  //true → blanco
Producto8 = "negro" || false    //true → negro
```

//Ejemplos de negación

```
Negacion1 = !true              //false
Negacion2 = !false             //true
Negacion3 = !"negro"           //false
```

#### Operadores de comparación

Los operadores de comparación son similares a los lógicos, con la excepción de que éstos no tienen por qué ser booleanos..

- Igualdad (==). Devuelve true si los dos operandos son iguales. Si alguno de ellos es de diferente tipo, JavaScript lo convierte para la comparación.
- No igualdad o diferente (!=). Devuelve true si los dos operandos son diferentes. Si alguno de ellos es de diferente tipo, JavaScript lo convierte para la comparación.
- Igualdad estricta (===). Devuelve true si los dos operandos son iguales y del mismo tipo.
- No igualdad estricta (!==). Devuelve true si los dos operandos no son iguales o del mismo tipo.
- Mayor que (>). Devuelve true si el operando de la izquierda es mayor que el de la derecha.
- Menor que (<). Devuelve true si el operando de la izquierda es menor que el de la derecha.
- Mayor o igual que (>=). Devuelve true si el operando de la izquierda es mayor o igual que el de la derecha.
- Menor o igual que (<=). Devuelve true si el operando de la izquierda es menor o igual que el de la derecha.

Ejemplo:

//Variables de ejemplo

```
Variable1 = 5;
Variable2 = 10;
```

//Comparaciones

```
5 == Variable1           //true
"5" == Variable1         //true
5 == "5"                  //true
Variable1 != 10           //true
Variable2 != "5"          //true
5 === Variable1           //true
Variable1 !== "5"         //true
5 !== "5"                 //true
```

#### Visión y Pasión por Emprender y Servir

```
Variable2 > Variable1 //true
Variable2 >= Variable1 //true
Variable1 >= 5 //true
Variable1 < Variable2 //true
Variable1 <= variable2 //true
Variable2 <= 10 //true
```

#### Operadores condicionales

Un operador condicional devuelve un valor determinado en función del resultado obtenido al evaluar una condición.

(condición) ? valor1: valor2;

Si la condición es verdadera se devuelve valor1 (verdadero). En caso contrario, si la condición es falsa se devuelve valor2 (falso). Ejemplo:

(nombre = "Fer") ? "Me llamo Fernando" : "Tú no eres Fernando";

Según el fragmento de script anterior, si el valor de la variable nombre es Fer, el resultado que se obtienen es Me llamo Fernando. En caso contrario, el resultado será Tú no eres Fernando.

#### Operadores de asignación

Un operador de asignación se utiliza para asignar un valor a una variable.

- Igualdad (=), coloca el resultado del operando de la derecha en la variable de la izquierda del símbolo: valor1 = valor2.
- Adición (+). valor1 += valor2.
- Sustracción (-). valor1 -= valor2.
- Multiplicación (\*). valor1 \*= valor2.
- División (/). valor1 /= valor2.
- Módulo (%). valor1 %= valor2.

#### Estructuras condicionales

Hasta este momento ya se sabe trabajar con variables, manipularlas y operar con ellas. Otra de las acciones que JavaScript puede hacer es tomar decisiones comparando variables a través de estructuras condicionales.

#### Estructuras condicionales

Al igual que sucede con la mayoría de los lenguajes de programación, en determinado momento es necesario controlar el desarrollo o el flujo del programa que éste tome decisiones.

#### La sentencia if

El comando if se emplea cuando es necesario ejecutar un fragmento de código en función de los valores de determinadas variables.

La estructura if tiene la siguiente sintaxis:

```
if (condición)
{
Código de programa en caso de que la condición sea verdadera
}
```

Como se aprecia en el modelo anterior, la condición siempre se indica entre paréntesis.

Es importante destacar que los operadores que se deberán utilizar son los de comparación en vez de los de asignación.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1>Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Ejemplo de if</h1>

<script language="JavaScript" type="text/javascript">
var Valor1, Valor2;
Valor1 = 20;
Valor2 = 10;
if (Valor1 >= Valor2)
{
document.write("El Valor 1 es más grande que el Valor 2")
}

</script>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</body>
</html>
```

### Estructuras if ... else

Las estructuras if ... else tienen la siguiente sintaxis:

```
if (condición)
{
Código de programa en caso de que la condición sea verdadera
}
else
{
Código de programa en caso de que no se cumpla la condición
}
```

Así pues, se puede diferenciar dos bloques: el bloque if que ejecuta un código si la condición es verdadera o se cumple; y el bloque else que hace lo propio, pero en caso de que la condición no sea cierta o no se cumpla, es decir, es falsa.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
```



Visión y Pasión por Emprender y Servir

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>uso de if ... else</h1>

<script language="JavaScript" type="text/javascript">

function hola()
{
if (document.formulario.nombre.value == "Fernando")
{
alert("Buenos días Fernando. Hace rato que te esperaba.");
}
else
{
alert("No te conozco, pero de todos modos, buenos días");
}
}
</script>

<form action="" id="formulario" name="formulario">
<p>Seleccione un nombre de la lista:
<select name="nombre" size="1" onchange="hola();">
<option value="Manolo">Manolo</option>
<option value="Lucas">Lucas</option>
<option value="Fernando">Fernando</option>
<option value="José">Jos&eacute;</option>
<option value="Javier">Javier</option>
</select>
</p>
</form>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

Estructuras if anidadas

Los bloques if pueden estar formados por muchas estructuras if, es decir, estructuras if que están colocadas unas dentro de otras, en otras palabras, anidadas.

**Visión y Pasión por Emprender y Servir**

Según el ejemplo anterior, un bloque if sólo dispone de dos opciones, la verdadera y la falsa. No obstante, se puede hacer que en un caso de que la condición sea falsa, el programa pueda presentar dos opciones más y así hasta infinitas opciones.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza, Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>uso de if anidadas</h1>

<script language="JavaScript" type="text/javascript">
var nombre;
nombre = prompt("Hola, ¿quién eres, Fernando o Nancy?", "");
if (nombre == "Fernando")
{
alert("Buenos días Fernando, hace rato que te esperaba.");
}
else
{
if (nombre == "Nancy")
{
alert("Buenos días Nancy, qué tal tus vacaciones.");
}
else
{
alert("No te conozco, pero de todos modos, buenos días.");
}
}
}
</script>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

En este ejemplo se ha introducido un elemento nuevo, el método prompt(). Este método abre una ventana de diálogo en la que se puede introducir una cadena que será almacenada en la variable nombre. Después, el valor introducido es comparado en todas las sentencias if.

Sin embargo, el número de líneas se puede reducir de la siguiente manera:

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1>Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de if con condiciones</h1>

<script language="JavaScript" type="text/javascript">
var nombre;
nombre = prompt("Hola, ¿quién eres, Fernando Nancy?","");
if ((nombre == "Fernando") || (nombre == "Nancy"))
{
alert("Buenos días " + nombre + ", hace rato que te esperaba");
}
else
{
alert("No te conozco " + nombre + ", pero de todos modos, buenos días.");
}
</script>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

Como se puede observar en el ejemplo anterior, se utiliza una única sentencia if ... else para evaluar las dos posibles respuestas con un operador lógico o.

### Los bucles for

Las estructuras if permiten que un programa tome decisiones, ahora se verá cómo se puede hacer que una determinada tarea se repita hasta que se cumpla con una condición.

Los bucles for se utilizarán como contadores. Gracias a ellos se puede hacer que un determinado proceso se repita un número de veces determinado.

Para configurar un bucle for se tienen que establecer cuatro parámetros o valores:

- Valor inicial. Determina el valor inicial del contador.
- Condición. Comprueba si la condición es verdadera o falsa.
- La ejecución. Se ejecutan las instrucciones deseadas.
- Actualización. Se actualiza el valor del Contador para Volver a ejecutar el paso 2, condición.

El modelo del bucle for es:

```
for (valor inicial; condición; actualización)
{
  Instrucciones a ejecutar por el bucle
}
```

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso del bucle for</h1>

<script language="JavaScript" type="text/javascript">
var contador = 0;
for (contador; contador < 11; contador++)
{
  document.write("este es el número " + contador + "<br>");
}
</script>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

### Bucles for ... in

Los bucles for ... in repiten una variable sobre todas las propiedades de un objeto. Así, para cada valor de un objeto que se le pasa, se ejecutarán las sentencias del bucle.

Para configurar un bucle for ... in hay que tener en cuenta los siguientes elementos:

- Valor inicial. Determina el valor de la primera propiedad del objeto.
- La ejecución. Se ejecutan las instrucciones deseadas.
- Actualización. Se asigna a una variable el valor de la siguiente propiedad del objeto. Si no hay más propiedades, se termina el bucle.

La estructura básica de un bucle for ... in es:



```
for (variable in objeto)
{
  Sentencias
}
```

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza, Nezhualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de for .. in</h1>
<script language="JavaScript" type="text/javascript">
function recorrer(obj, nombreobjeto)
{
  var propiedad;
  propiedad = window.open("", "Ventana Propiedad", "scrollbars=yes, width=300, height=300");
  propiedad.document.write("<head><title>PROPIEDADES</title></head>");
  propiedad.document.write("<b>Propiedades Input radio</b><br><br><br>");
  for (m in obj)
  {
    propiedad.document.write("<p style='color:blue; font-size=10pt'>" + nombreobjeto + "." + m + " = " + obj[m] + "</p>");
  }
}
</script>

<form action="" id="formulario">
<p>
<input type="radio" name="radio">
Opci&oacute;n 1
<br>
<input type="button" value="Pincha" onclick="recorrer(document.formulario.radio,'radio')">
</p>
</form>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

En este ejemplo, se recorren todas las propiedades del objeto radio hasta que llega a la última. Solo entonces se termina la ejecución del bucle.



## Los bucles while

La instrucción while ejecuta de manera constante una determinada secuencia de Código siempre que se cumpla una condición.

Los bucles while pueden generar secuencia infinitas, por lo que se deberá tener especial cuidado cuando se definan sus parámetros. Puede quedar bloqueado el sistema si un script se encuentra en un bucle sin fin.

La estructura básica de un bucle while es:

```
while (condición)
{
  Instrucciones a ejecutar con el bucle
}
```

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza, Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1>Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

</h1>Uso de while</h1>

<script language="JavaScript" type="text/javascript">
var contador = 0;
while (contador < 11)
{
  document.write("Este es el número " + contador + "<br>");
  contador++;
}
alert("Fin del bucle");
</script>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

En este bucle, mientras la condición (contador < 11) es verdadera, se ejecuta el Código delimitado entre las llaves. Cuando la condición es falsa, se deja de ejecutar el bucle y se pasa a la siguiente instrucción o línea de Código.

## Los bucles do ... while

Los bucles do ... while son muy conocidos en muchos lenguajes de programación, como, por ejemplo, Visual Basic.

La misión de un bucle de este tipo es la de colocar una expresión condicional después de las instrucciones que deben ejecutarse, de esta manera, es seguro que las instrucciones dentro del bucle son ejecutadas por lo menos una vez.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza, Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1>Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de do ... while</h1>

<script language="JavaScript" type="text/javascript">
var password;
do
{
parssword = prompt("Introduzca la clave de acceso[papepipopu]");
}
while (password != "papepipopu");
alert("Código de acceso correcto.\nFin del bucle.");
</script>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

Al ejecutar el script anterior aparece una ventana de diálogo que pide al usuario la entrada de una clave de acceso, entre corchetes aparece la correcta. Si la condición (password != "papepipopu") se cumple, es decir, es verdadera, el bucle deja de ejecutarse saltando el programa a la línea siguiente.

Si se intenta cambiar la clave de acceso especificada introduciendo otra, se verá cómo el script vuelve al inicio del bucle solicitando nuevamente la clave de acceso.

## La instrucción break

**Visión y Pasión por Emprender y Servir**

La instrucción break permite interrumpir la ejecución de un bucle, ya sea for o while, aunque éste no haya sido ejecutado totalmente. Así pues, se puede interrumpir o cancelar la ejecución de un bucle cuando se cumpla una determinada condición.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de break</h1>

<script language="JavaScript" type="text/javascript">
var nombre = "";
while (nombre != "fin")
{
nombre = prompt("Introduce un nombre: ", "");
if (nombre == "fin")
{
alert("Se ha escrito la palabra mágica");
break;
}
}
</script>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

**La instrucción continue**

La instrucción continue no termina la ejecución del bucle, sino que puede dejar de ejecutar una repetición determinada y pasar a la siguiente.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
```

**Visión y Pasión por Emprender y Servir**

```
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de continue</h1>

<script language="JavaScript" type="text/javascript">
var numero = 0;
for (numero = 0; numero < 100; numero++)
{
if (numero/2 == parseInt(numero / 2))
{
continue;
}
document.write("El número " + numero + " es impar<br>!");
}
</script>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

En este ejemplo, se muestra en pantalla todos los números impares. Con la función `parseInt()`, se va a localizar los números pares para eliminar su impresión de la pantalla mediante la instrucción `continue`, visualizando así únicamente los números impares.

**La sentencia switch**

La sentencia `switch` evalúa una expresión y la compara con cada una de las etiquetas asociadas. Cuando encuentra una igualdad, el programa evalúa la expresión asociada a dicha etiqueta.

La sintaxis de `switch` es la siguiente:

```
switch (expresión)

{
case etiqueta1:
sentencias;
break;
case etiqueta2:
sentencias;
break;
...
default:
sentencias;
}
```

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de switch</h1>

<script language="JavaScript" type="text/javascript">
var actriz = prompt("¿Cuál de estas actrices es la mejor?\n[Catherine Zeta Jones, Demi Moore, Sandra Bullock]");
switch (actriz)
{
case "Catherine Zeta Jones":
alert("Tu actriz favorita es Catherine Zeta Jones");
break;
case "Demi Moore":
alert("Tu actriz favorita es Demi Moore");
break;
case "Sandra Bullock":
alert("Tu actriz favorita es Sandra Bullock");
break;
default:
alert("Ningún valor coincide con las etiquetas case definidas.\nPruebe otra vez");
}
</script>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

### Ventanas de confirmación

Hasta ahora, las ventanas de alerta que se han visto se limitan a mostrar una ventana de diálogo con un texto definido y un botón de Aceptar. Si en una estructura condicional if ... else se especifica la instrucción confirm con sus correspondientes opciones condicionales, se puede hacer que el usuario tome decisiones de manera directa.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
```

**Visión y Pasión por Emprender y Servir**

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>
</head>
<body>
<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Uso de ventana de confirmación</h1>

<script language="JavaScript" type="text/javascript">
function confirma()
{
if (confirm("Mensaje de confirmación.\n¿Estás de acuerdo?"));
{
alert("Muy bien, has pulsado ACEPTAR");
}
else
{
alert("OK. Has pulsado CANCELAR");
}
}
}
</script>

<form action="" id="formulario">
<p>Ventana de alerta con confirmación.<br>
<input type="button" value="Pulsa aquí;" onclick="confirma();">
</p>
</form>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

**Ejemplo de carrito de compras**

Hasta hace no mucho tiempo, el comprar productos por Internet era algo inusual ya que los sistemas de transacciones disponibles eran precarios e inseguros. Actualmente, existe un método que está adquiriendo popularidad en las distintas webs de ventas: el carrito de compras.

En el carrito de compras se añaden a una lista virtual los productos que interesan de una serie de catálogos disponibles sin necesidad de tener que hacerlo de forma personal, y se envía de una sola vez toda la solicitud de compra.

Por lo regular, los carritos de compras están implementados o desarrollados en PHP u otro lenguaje de programación mucho más seguro que JavaScript. Sin embargo, se mostrará cómo implementar un carrito de compras.

### Visión y Pasión por Emprender y Servir

En general, el diseño de un carrito de compras se compone de tres elementos, el primero es el catálogo de productos o servicios con los precios, unidades, descripción del producto o servicio, fotos, etc.

Se puede tener una o varias páginas dependiendo del número de productos o servicios a ofrecer. Generalmente, cada producto o servicio tiene al lado un botón que permite añadirlo al carrito de compras y otro que se encarga de mostrar dicho carrito de compras, es decir, los productos o servicios que se van añadiendo.

El siguiente elemento es la hoja de pedido de los productos o servicios. Si se pulsa el enlace correspondiente, se encontrará con una tabla o listado que es, propiamente, el carrito de compras. En ella se pueden ver los productos, los precios parciales y totales, descuentos, etc., de los elementos que se van añadiendo. Estos listados suelen tener opciones para vaciar todo el carrito o para eliminar productos o servicios uno a uno.

Finalmente, aparecen los elementos necesarios para formalizar la transacción comercial. Una vez realizada dicha transacción, suele recibirse un correo electrónico de confirmación con la lista de la compra, el número del pedido, fecha aproximada de entrega, etc.

A continuación, se diseñará un carrito de compras ficticio para una tienda.

Este carrito de compras utiliza cookies para gestionar los pedidos. Si el navegador tiene desactivada la recepción de cookies, el ejemplo no funcionará. Por otro lado, la seguridad que ofrece es muy baja y no debe usarse nunca para transmitir números de cuenta o de tarjetas vía Internet, es sólo un ejemplo.

### El catálogo

Para el diseño del catálogo que contendrá todos los productos disponibles en la tienda virtual de ejemplo, se han empleado la mayoría de las funciones para trabajar con las cookies, junto con una interesante hoja de estilos que configura en tonos azulados la pantalla, también se ha cambiado el característico aspecto de los botones y cuadros de texto de los formularios.

La función `meterencarro()`, se basa en el comando `setcookie` que se encargará de añadir al carrito de compras los distintos productos que se van seleccionando. La función numera secuencialmente las distintas entradas a la cookie y verifica, además, el número máximo de ellas.

Los datos que pasa `setcookie` a la cookie se obtiene de una serie de inputs de texto ocultos del formulario de cada producto. En este caso, para cada entrada se almacena el número de elementos pedidos, el producto, la referencia y el precio.

Finalmente, para dar un poco de control al script, se ha añadido una sencilla sentencia condicional que permite abortar la selección de un producto. Una vez seleccionado y validado un producto, una ventana de información avisa de la selección efectuada.

Se ha limitado el número máximo de unidades de compra para cada producto a 99 y el total de pedidos a 5.

### Ejemplo de catálogo.htm

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza, Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>

<style type="text/css">
    body {
        background:rgb(102,153,204);
        font: 10pt Verdana, Geneva, Arial, Helvetica, sans-serif;
    }
    #inputs {
        border:1px inset rgb(153,204,255);
        width: 40px;
```

**Visión y Pasión por Emprender y Servir**

```
background:rgb(148,204,252);
text-align : center;
}
#boton {
background: rgb(102,153,204) border:1px outset rgb(153,204,255);
width:80px;
font : 8pt Verdana, Geneva, Arial, Helvetica, sans-serif;
background-color : #FFFACD;
color : #4B0082;
}
</style>

<script language="JavaScript" type="text/javascript">

function meterencarro(formu)
{
unidades=formu.numpedido.value;
descripcion=formu.producto.value;

if (confirm("El siguiente producto se va a añadir al carro de la compra.\n\n " + descripcion + " \n\n¿Está usted de acuerdo?"))
{
numeroregistro = 0;
numeroregistro = getcookie("numerodeorden");
numeroregistro++;

if ( numeroregistro > 5 )
alert("ATENCIÓN\nSu carrito está lleno.\nPor favor, acceda a la hoja de pedido.\nGracias.");
else
{
actualizarbase = formu.numpedido.value + "|" + formu.precio.value + "|" + formu.referencia.value + "|" + formu.producto.value;
nuevopedido = "Order." + numeroregistro;
setcookie (nuevopedido, actualizarbase, null, "");
setcookie ("numerodeorden", numeroregistro, null, "");

aviso = "PRODUCTO SELECCIONADO\n\n" + "Cantidad: " + unidades + " unidad/es.\n"
+ "Producto: \n" + descripcion + ".\n\nPulse sobre Ver Carro para acceder\na su lista de compra.\nGracias";
alert(aviso);
}
}
}

function getcookieval (offset)
{
var endstr = document.cookie.indexOf(";", offset);

if (endstr == -1)
endstr = document.cookie.length;
return unescape(document.cookie.substring(offset, endstr));
}

function getcookie (name)
{
var arg = name + "=";
var alen = arg.length;
var clen = document.cookie.length;
var i = 0;

while (i < clen)
{
var j = i + alen;
```



## Visión y Pasión por Emprender y Servir

```

if (document.cookie.substring(i, j) == arg) return getcookieval (j);
i = document.cookie.indexOf(" ", i) + 1;
if (i == 0) break;
}
return null;
}

function setcookie (name,value,expires,path,domain,secure)
{
    document.cookie = name + "=" + escape (value) +
        ((expires) ? "; expires=" + expires.toGMTString() : "") +
        ((path) ? "; path=" + path : "") +
        ((domain) ? "; domain=" + domain : "") +
        ((secure) ? "; secure" : "");
}

</script>
</head>

<body>

<header>
<h1>CCAP México</h1>
<h1> Bienvenido al Mundo del HTML5</h1>
<h2>¡No te asustes, no es tan difícil!</h2>
</header>

<!-- Contenido del documento -->

<h1>Carrito de la compra DEMO</h1>

<div align="center">

<font face="Verdana" size="+1" color="Navy">&nbsp;   CARRITO DE LA COMPRA DEMO</font>
</div>
<br>

<table width="700" border="1" cellspacing="1" cellpadding="5" align="center" style="font: 10pt Verdana, Geneva, Arial, Helvetica, sans-serif;">
<tbody>
<tr style="font: bold">
<td width="20" >Foto </td>
<td width="150" >Producto </td>
<td width="80" >Pesos </td>
<td width="80" >Control</td></tr>
<tr>
<td></td>
<td style="font: 9pt; text-align: justify;"><b>Diseño de páginas web interactivas con JavaScript y CSS.</b> Este libro constituye la herramienta perfecta para realizar páginas Web interactivas, ya que se explican de manera clara y con multitud de ejemplos, las características fundamentales de JavaScript </td>
<td>$ 180.30</td>
<td>
<table>
<tr>
<td colspan="3" align="center" style="font: 7pt; color: #191970;">Indique cantidad</td>
</tr>
<tr>
<td><form name="order">
<input onclick="meterencarro(this.form)" type="button" id="boton" value="Adquirir"></td>
<td><input maxlength=2 id="inputs" size="2" value="1" name="numpedido"></td>

```

#### Visión y Pasión por Emprender y Servir

```

<td><a href="pedido.html"></a>
</td><input type="hidden" value="18.03" name="precio">
<input type="hidden" value="Diseño de páginas web interactivas con JavaScript" name="producto">
<input type="hidden" value="ID373-7" name="referencia">
</form>

</tr>
</table>

</td>
</tr>
<tr>
<td></td>
<td style="font: 9pt; text-align: justify;"><b>Macromedia FLASH 5. Curso práctico.</b> En el libro encontrará completas y detalladas
explicaciones asequibles a todos los niveles para aprender
a trabajar con Flash 5.</td>
<td>$ 270.50</td>
<td>

<table>
<tr>
<td colspan="3" align="center" style="font: 7pt; color: #191970;">Indique cantidad</td>
</tr>
<tr>
<td>
<form name="order">
<input onclick="meterencarro(this.form)" type="button" id="boton" value="Adquirir">
</td>
<td><input maxlength=2 id="inputs" size="2" value="1" name="numpedido"></td>
<td><a href="pedido.html"></a>
</td>
<input type="hidden" value="27.05" name="precio">
<input type="hidden" value="Macromedia FLASH 5. Curso práctico." name="producto">
<input type="hidden" value="ID471-7" name="referencia">
</form>
</tr>
</table>

</td>
</tr>
<tr>
<td></td>
<td style="font: 9pt; text-align: justify;"><b>MIDI y MOD fácil con el PC.</b> En este libro encontrará un método adecuado a sus necesidades para
componer música con su PC, ya sea a través del Midi o de los archivos de muestras (Módulos). </td>
<td>$ 285.50</td>
<td>

<table>
<tr>
<td colspan="3" align="center" style="font: 7pt; color: #191970;">Indique cantidad</td>
</tr>
<tr>
<td><form name="order">
<input onclick="meterencarro(this.form)" type="button" id="boton" value="Adquirir"></td>
<td><input maxlength=2 id="inputs" size="2" value="1" name="numpedido"></td>
<td><a href="pedido.html"></a>
</td>
<input type="hidden" value="28.55" name="precio">
<input type="hidden" value="Midi y MOD fácil con el PC." name="producto">
<input type="hidden" value="ID214-5" name="referencia">
</form>

```



```
</tr>
</table>
</td>
</tr>

<tr>
<td></td>
<td style="font: 9pt; text-align: justify;"><b>Enciclopedia de CorelDRAW 9.</b> El libro contiene todas las aplicaciones del paquete con
explicaciones exhaustivas a todos los niveles</td>
<td>$ 510.90</td>
<td>

<table>
<tr>
<td colspan="3" align="center" style="font: 7pt; color: #191970;">Indique cantidad</td>
</tr>
<tr>
<td><form name="order">
<input onclick="meterencarro(this.form)" type="button" id="boton" value="Adquirir"></td>
<td><input maxlength=2 id="inputs" size="2" value="1" name="numpedido"></td>
<td><a href="pedido.html"></a>
</td><input type="hidden" value="51.09" name="precio">
<input type="hidden" value="Enciclopedia de CorelDRAW 9." name="producto">
<input type="hidden" value="ID411-3" name="referencia">
</form>
</tr>
</table>
</td>
</tr>
</tbody>
</table>
<br>
<div align="center"><font face="Verdana" size="1" color="Navy"><a
href="mailto:ferjimenez@ccapmexico.com.com">ferjimenez@ccapmexico.com</a><br>
Optimizado para una resolución de 800*600<br>
Copyright &#169; Fernando Jiménez Ávila. Nezahualcóyotl. México,2024
</div>

<footer>
<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>
</footer>
</body>
</html>
```

### La hoja de pedido

En la hoja de pedido del carrito de compras, aunque pueda parecer todo lo contrario, está el verdadero trabajo de programación. Por un lado, se tiene que visualizar el contenido de los productos que se han añadido al carrito de compras mediante una estructura más o menos clara, y por el otro, se tiene que facilitar al usuario todas las herramientas necesarias para poder gestionar libremente los productos del carrito de compras.

En primer lugar, se verá cómo se muestran los productos que el usuario ha ido añadiendo al carrito de compras. En este caso, todo el trabajo lo realiza la función `visualizacarrito()`, que se ejecuta en un lugar estratégico del documento, en concreto, justo después de la definición de las columnas de la tabla de la hoja de pedido.

La función `getcookie()` es la encargada de leer el contenido de las cookies del carrito de compras para ir añadiendo en una matriz los diferentes campos de datos. También calcula el valor total del carrito de compras, multiplicando los parciales o subtotales de cada producto para sumar al final el total de cada fila. El valor total es procesado por la función `formatopesos()`, que muestra con dos decimales, cuyo valor se ha redondeado, el precio total de la compra.

Finalmente, el script se encarga de generar el resto de celdas de la tabla, así como los botones eliminar y catálogo, mediante JavaScript.

El resto de las funciones son `deletcookie()`, que establece la fecha actual en la cookie seleccionada, con lo que ésta desaparece del carrito de compras.

Un formulario al final de la hoja de pedido permite enviar a una cuenta de correo electrónico todo el contenido del carrito de compras.

No hay que olvidar que no es prudente solicitar por este medio un número de cuenta o de tarjeta, ya que la seguridad de la transacción es mínima.

### **Ejemplo de pedido.htm**

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<meta http-equiv="Content-Language" content="ES-ES"/>
<meta name="description" content="Curso de Elaboración Básica de Páginas Web">
<meta name="keywords" content="curso, web, página, html, css, javascript, ccap, México, js, programación, html5, script, Neza,
Nezahualcóyotl"/>
<meta name="author" content="Fernando Jiménez Ávila"/>
<meta name="distribution" content="Global"/>
<meta name="copyright" content="© Fernando Jiménez Ávila"/>
<title>Elaboración Básica de Páginas Web ..... CCAP México</title>

<style type="text/css">
    body
    {
        background:rgb(102,153,204);
        font:10pt Verdana, Geneva, Arial, Helvetica, sans-serif;
        text-align:left;
    }
    #inputcarro
    {
        border:1px inset rgb(153,204,255);
        width:40px;
        background:rgb(102,153,204);
        text-align:center;
        color:white;
    }
    #inputs
    {
        border:1px inset rgb(153,204,255);
        width: 280px;
        background:rgb(102,153,204);
        text-align:center;
        color:white;
    }
    #boton
    {
        background: rgb(102,153,204) border:1px outset rgb(153,204,255);
        width:80px;
        font:8pt Verdana, Geneva, Arial, Helvetica, sans-serif;
        background-color:#FFFACD;
        color:#4B0082;
    }
    div.marco {border:2px groove rgb(153,204,255);width:600px; padding:15px}
    p.texto {letter-spacing: 3px;color:rgb(0,51,102);font-weight: bold}
    span.label { width:150px;vertical-align:top;color:white; float:left}
    div {margin-top:15px}
</style>
```

```
<script language="javascript" type="text/javascript">

function getcookieval (offset)
{
    var endstr = document.cookie.indexOf(";", offset);

    if (endstr == -1)
        endstr = document.cookie.length;
    return unescape(document.cookie.substring(offset, endstr));
}

function getcookie (name)
{
    var arg = name + "=";
    var alen = arg.length;
    var clen = document.cookie.length;
    var i = 0;

    while (i < clen)
    {
        var j = i + alen;
        if (document.cookie.substring(i, j) == arg) return getcookieval (j);
        i = document.cookie.indexOf(" ", i) + 1;
        if (i == 0)
            break;
    }
    return null;
}

function setcookie (name,value,expires,path,domain,secure)
{
    document.cookie = name + "=" + escape (value) +
        ((expires) ? "; expires=" + expires.toGMTString() : "") +
        ((path) ? "; path=" + path : "") +
        ((domain) ? "; domain=" + domain : "") +
        ((secure) ? "; secure" : "");
}

function formatpesos(input)
{
    var pesos = Math.floor(input)
    var tmp = new String(input)

    for (var decimalAt = 0; decimalAt < tmp.length; decimalAt++)
    {
        if (tmp.charAt(decimalAt)==".")
            break;
    }
    var centimos = "" + Math.round(input * 100)
    centimos = centimos.substring(centimos.length-2, centimos.length)
    pesos += ((tmp.charAt(decimalAt+2)=="9")&&(centimos=="00"))? 1 : 0;

    return pesos + "." + centimos
}

function quitardelcarrito(RemOrder)
{
    if (confirm("El producto seleccionado va a ser eliminado de su carro de la compra.\n¿Está usted de acuerdo?"))
    {

```

```
numerodeorden = getcookie("numerodeorden");

for(i=RemOrder; i < numerodeorden; i++)
{
    nuevopedido1 = "Order." + (i+1);
    nuevopedido2 = "Order." + (i);
    datos = getcookie(nuevopedido1);
    setcookie (nuevopedido2, datos, null, "/");
}
nuevopedido = "Order." + numerodeorden;
setcookie ("numerodeorden", numerodeorden-1, null, "/");
deletecookie(nuevopedido,"/");
location.href=location.href;
}
}

function visualizarcarrito()
{
    numerodeorden = 0;
    subtotal=0;
    preciototal=0;
    numerodeorden = getcookie("numerodeorden");
    tablas = "";

    for (i = 1; i <= numerodeorden; i++)
    {
        nuevopedido = "Order." + i;
        datos = "";
        datos = getcookie(nuevopedido);
        ficha0 = datos.indexOf("|", 0);
        ficha1 = datos.indexOf("|", ficha0+1);
        ficha2 = datos.indexOf("|", ficha1+1);
        campos = new Array;
        campos[0] = datos.substring( 0, ficha0 );
        campos[1] = datos.substring( ficha0+1, ficha1 );
        campos[2] = datos.substring( ficha1+1, ficha2 );
        campos[3] = datos.substring( ficha2+1, datos.length );
        subtotal = subtotal + (campos[1] * campos[0]);
        preciototal = formatopesos(subtotal);
        tablas += "<tr style='font: 9pt; text-align: justify; color=white'><td>" + campos[2] + "</td><td style='color=navy'>"
            + campos[3] + "</td><td>" + campos[1]
            + " €</td><td><input type=text id='inputcarro' size=2 name='Cantidad'+ i +'\" value='\""
            + campos[0] + "\"></td>"
            + "<td><input type=button id=boton value='\" Eliminar \"' onClick='\"quitardelcarrito(\"+i+\")\">"
            + "&nbsp;<input type=button id=boton value='\" Catálogo \"' onClick='\"parent.history.back()\"></td>"
            + "<input type=hidden name='\"Referencia\"+ i +'\" value='\"' + campos[2] + '\">"
            + "<input type=hidden name='\"Producto\"+ i +'\" value='\"' + campos[3] + '\">"
            + "<input type=hidden name='\"Pesos\"+ i +'\" value='\"' + campos[1] + '\">";
    }
    document.write(tablas);
    document.write("</td></tr><tr><td colspan=2 style='font: 9pt'>TOTAL CARRITO IVA INCLUIDO</td><td style='color: navy'>");
    document.write(preciototal);
    document.write(" $</td><td colspan=2 style='font: 7pt; text-align: center; color=white'>Páginas web interactivas con Javascript y CSS. Editorial
    CCAP México. 2024</td>");
}

function deletecookie(name, path, domain)
{
    if (getcookie(name))
```





Visión y Pasión por Emprender y Servir

<p style='font: 9pt; text-align: center; color=white'>El pedido será colocado en la bandeja de salida de su programa de correo para ser enviado a un correo electrónico inexistente llamado carritonulo@carrito.com</p>

</div>

</div>

<br>

<div align="center"><font face="Verdana" size="1" color="Navy" style="font:normal"><a href="mailto:ferjimenez@ccapmexico.com">ferjimenez@ccapmexico.com</a><br>

Optimizado para una resolución de 800\*600<br>

Copyright &#169; Fernando Jiménez Ávila. Nezahualcóyotl. México,2024

</div>

</form>

<footer>

<p>El contenido de este ejemplo no está protegido por derechos de autor.</p>

</footer>

</body>

</html>