

**CSS**

**paso a paso**

**Edgar D'Andrea**

# Contenido

<b>Introducción.....</b>	<b>15</b>
<b>1. CSS: Conceptos básicos.....</b>	<b>21</b>
<b>2. Definición de estilos.....</b>	<b>35</b>
<b>3. Valores y unidades.....</b>	<b>63</b>
<b>4. Herencia y cascada.....</b>	<b>73</b>
<b>5. Propiedades de Fuentes.....</b>	<b>81</b>
<b>6. Formateo del texto.....</b>	<b>101</b>
<b>7. Formateo de listas .....</b>	<b>135</b>
<b>8. Bordes, márgenes y rellenos.....</b>	<b>147</b>
<b>9. Colores y fondos.....</b>	<b>163</b>
<b>10. Posicionamiento y diseño.....</b>	<b>179</b>
<b>11. Contenido autogenerado .....</b>	<b>223</b>
<b>12. HTML dinámico y JavaScript.....</b>	<b>235</b>
<b>13. CSS, JavaScript y pseudo clases.....</b>	<b>285</b>
<b>14. Estilo en la impresión.....</b>	<b>301</b>
<b>15. Estilos de interfaz de usuario.....</b>	<b>317</b>
<b>A. Codificación CSS.....</b>	<b>323</b>
<b>B. Guía rápida de CSS.....</b>	<b>329</b>
<b>C. Práctica completa.....</b>	<b>363</b>
<b>Índice.....</b>	<b>401</b>

# Índice general

<b>Introducción.....</b>	<b>15</b>
¿Qué es CSS?.....	15
Enfoque del libro.....	16
¿Qué necesito saber para seguir el aprendizaje?.....	16
¿Qué necesito tener para seguir las prácticas?.....	16
Esquema general del libro.....	17
<b>Capítulo 1:</b>	
<b>CSS: Conceptos básicos.....</b>	<b>21</b>
¿Para qué se necesitan los estilos?.....	22
Asignación de estilo por el autor del documento.....	22
<b>Interrelación HTML y estilos.....</b>	<b>23</b>
Primera definición CSS.....	25
Modos de definir los estilos.....	27
Hoja de estilo interna.....	27
Hoja de estilo externa (<LINK>).....	28
Definición en línea.....	29
¿Por qué se denomina hoja de estilo en cascada?.....	30
La cascada llega al estilo en línea.....	32
Notación !important.....	32
Decisión de la precedencia.....	32
<b>Capítulo 2:</b>	
<b>Definición de estilos.....</b>	<b>35</b>
<b>Legibilidad.....</b>	<b>36</b>
<b>Etiqueta &lt;STYLE&gt;.....</b>	<b>36</b>
Atributos en la etiqueta <STYLE>.....	38
Definiciones dentro del bloque <STYLE>.....	38
Aplicación de estilo en selectores.....	40
Referencia a una clase.....	40
Definición de una clase.....	42
Definición de estilo con identificador.....	43
Selector universal.....	44
Selector limitado.....	44

- Limitación a un atributo de un selector.....45
- Estructura de un documento HTML.....46
  - Selección por jerarquía.....49
- Herencia del estilo.....50
- Pseudo clases.....51
  - Pseudo clases para la etiqueta <A>.....51
    - Orden de las definiciones.....52
  - Pseudo clase :first-child .....53
  - Pseudo clase :lang .....53
- Pseudo elementos.....53
- Hojas de estilo en funcionamiento.....56
- Comentarios en CSS.....59
- Modos para asignar estilo.....60
  - Modo 1: etiqueta <STYLE>.....60
  - Modo 2: uso de la etiqueta LINK.....61
    - ¿Qué contiene una hoja de estilo externa?.....61
  - Modo 3: importando el estilo mediante la sentencia import.....62
  - Modo 4: mediante el atributo style (inline).....62

### Capítulo 3:

- Valores y unidades.....63
- Valores de las propiedades en las hojas de estilo.....63
- Unidades de medida.....64
  - Unidades de medida absolutas.....64
  - Unidades de medida relativas .....65
  - Unidades por porcentajes.....65
- Valores de color .....67
  - Colores RGBA.....68
  - Colores HSL .....68
  - Colores HSLA .....69
- Valores de URL.....69
- Valores de texto y palabras clave.....70
- Valores de control de audición.....70

### Capítulo 4:

- Herencia y cascada.....73
- Herencia.....74

<b>Cascada.....</b>	<b>75</b>
Estilos definidos por el autor.....	75
Estilos de usuario.....	76
Estilos del agente de usuario.....	76
<b>Prioridad en etiquetas LINK.....</b>	<b>77</b>
Determinación de la prioridad de aplicación.....	78
<b>Especificidad.....</b>	<b>78</b>
Cálculo de la especificidad.....	78

## Capítulo 5:

### **Propiedades de Fuentes.....81**

<b>Fuentes.....</b>	<b>81</b>
Propiedades asociadas a la fuente del texto.....	82
Espaciados.....	83
Tipos de fuentes.....	83
Tamaño de la fuente.....	86
Ventajas y desventajas de cada modo.....	86
Unidades de medida.....	86
El problema del redimensionamiento del texto.....	88
Estilo de la fuente.....	89
Espaciado de líneas.....	91
Ejemplos con fuentes de texto.....	92
Propiedad simplificadora: font.....	94
Uso de fuentes del sistema.....	95
<b>No todo es CSS ni todo son párrafos &lt;P&gt;.....</b>	<b>96</b>

## Capítulo 6:

### **Formateo del texto.....101**

<b>Alineación horizontal del texto (text-align).....</b>	<b>101</b>
<b>Alineación vertical del texto (vertical-align).....</b>	<b>103</b>
<b>Sangría del texto (text-indent).....</b>	<b>106</b>
<b>Elementos flotantes.....</b>	<b>109</b>
Propiedad clear.....	111
<b>Espaciado de letras y palabras.....</b>	<b>113</b>
Ejemplos con espaciado de letras y de palabras.....	113
Espacios en blanco.....	115
<b>Mayúsculas y minúsculas (text-transform).....</b>	<b>117</b>
<b>Decoración de texto (subrayar, tachar, etc.).....</b>	<b>119</b>

- Ejemplos con decoración y transformación de texto.....121
- Propiedades CSS vs. atributos en tablas HTML.....123**
  - Equivalencia de atributos.....125
  - Bordes de tablas.....125
  - Espaciado al borde de la tabla.....126
  - Contracción del borde.....127
  - Celdas vacías.....127
  - Table-layout.....131
  - Títulos de la tabla: alineación y posicionamiento.....132

## **Capítulo 7:**

- Formateo de listas .....135**
- Cómo funcionan las listas.....135**
  - Creación de un menú horizontal .....137
- Listas CSS.....138**
  - Creación de listas (display:list-item).....138
- Tipos de estilo de listas CSS (list-style-type).....139**
  - Posicionamiento de las marcas de los elementos .....143
  - Imágenes como marcadores.....144

## **Capítulo 8:**

- Bordes, márgenes y rellenos.....147**
- Concepto de caja de formato.....147**
- Campo de acción de márgenes, bordes y rellenos.....149**
- Elemento de relleno.....150**
  - Definición de valores.....150
  - Efecto del relleno.....151
- Elemento de bordes .....152**
  - Ancho del borde.....153
  - Estilo del borde.....154
  - Color del borde.....156
    - Codificación del color.....156
  - Conjunto de propiedades border.....156
  - Espaciado de bordes.....157
- Elemento de márgenes.....158**
  - Ejemplos con bordes, padding y márgenes.....159
- Contornos dinámicos.....161**

## Capítulo 9:

<b>Colores y fondos.....</b>	<b>163</b>
<b>Propiedades asociadas al color y al fondo.....</b>	<b>163</b>
<b>Color en los elementos.....</b>	<b>164</b>
Colores de primer plano (foreground).....	164
Distintas formas de expresar colores.....	164
Color de fondo (background).....	166
¿Hasta dónde se extiende el color de fondo? .....	169
Ejemplos con fondos de color .....	169
<b>Imágenes como fondo.....</b>	<b>170</b>
Repetición de la imagen.....	172
Desplazamiento de imágenes.....	173
Posicionamiento de la imagen de fondo.....	174
Ejemplo del uso de imágenes como fondo.....	175
<b>Propiedad background.....</b>	<b>177</b>

## Capítulo 10:

<b>Posicionamiento y diseño.....</b>	<b>179</b>
<b>Propiedades relacionadas con el posicionamiento.....</b>	<b>180</b>
<b>Posicionamiento de los elementos.....</b>	<b>181</b>
Posicionamiento estático (static).....	182
Posicionamiento relativo (relative).....	184
Ejemplo de posicionamiento relativo.....	186
Posicionamiento absoluto (absolute).....	187
Posicionamiento fijo (fixed).....	189
Ejemplo de posicionamiento fijo.....	190
Posicionamiento mediante porcentajes.....	191
Porcentajes con posicionamiento relativo.....	191
Porcentaje con posicionamiento absoluto.....	193
Cambio del tamaño con el posicionamiento.....	195
Flotación de los elementos.....	196
Evitar la flotación de otro elemento.....	197
Ejemplo de propiedad float para crear contenido en columnas.....	199
Control del tamaño del elemento.....	201
Especificación de límites de tamaños.....	201
Contenido que excede el contenedor .....	205
<b>Superposición de elementos.....</b>	<b>208</b>
Colocación de elementos por debajo del nivel 0.....	211
Ejemplo de superposición de elementos.....	214

**Control de la visibilidad del elemento.....215**  
    Diferencia entre visibility: hidden y display: none.....215

**El diseño mediante la propiedad display.....216**  
    Creación de tablas con la propiedad display .....220  
    Organización de una tabla con la propiedad display.....221

## **Capítulo 11:**

**Contenido autogenerado .....223**

**Propiedad Content.....223**

**Pseudo elementos.....225**  
    first-line y first-letter.....226  
    :before y :after.....228  
        Objetos open-quote y close-quote.....229  
    Numeración automática de los elementos.....231  
        Incremento del contador.....231  
        Reinicialización del contador.....231  
        Ejemplo del uso del objeto counter.....232  
        Listas numeradas.....233

## **Capítulo 12:**

**HTML dinámico y JavaScript.....235**

**Tareas propias del lenguaje script .....235**

**CSS y HTML dinámico.....236**

**JavaScript, un lenguaje cliente.....239**  
    JavaScript no es Java .....240  
    Lo que JavaScript no hace.....240

**La compatibilidad puesta en duda.....242**  
    Una decisión estratégica .....243

**¿Cómo codificar con JavaScript? .....244**  
    Complementos en los navegadores.....244  
    Extensión para desarrollador Web.....245  
    Herramientas de desarrollo de Internet Explorer.....247  
    Transferencia de servidor local a remoto.....247  
    Recomendaciones para el entorno de pruebas.....248

**Ejecución de una aplicación JavaScript.....249**  
    Conexión HTML/JavaScript mediante eventos.....253

**Conceptos básicos de Javascript.....254**

**Estructura del código .....255**



<b>Comentarios en el código.....</b>	<b>256</b>
<b>Nomenclatura .....</b>	<b>257</b>
Nomenclatura de variables.....	258
Nomenclatura de funciones.....	258
Longitud de los nombres.....	259
Palabras reservadas.....	259
<b>¿Está activado JavaScript?.....</b>	<b>259</b>
<b>Scripts de JavaScript.....</b>	<b>260</b>
Modo indirecto.....	261
Uso de archivos externos.....	263
<b>Bloque de instrucciones.....</b>	<b>264</b>
<b>Variables.....</b>	<b>265</b>
Nomenclatura de las variables.....	266
Tipo de una variable.....	266
<b>Sentencias condicionales.....</b>	<b>268</b>
Sentencia if.....	268
Operadores de comparación de valores.....	268
Combinación de condiciones.....	269
Sentencia if/else.....	269
Sentencia switch.....	270
Operador ternario.....	271
<b>Control de bucles de ejecución.....</b>	<b>271</b>
Sentencia for.....	271
Incremento/decremento por unidades.....	273
Sentencia while.....	274
Sentencia do/while.....	274
<b>Funciones.....</b>	<b>275</b>
Parámetros de la función.....	276
Llamada de función.....	276
Cómo saber cuál fue la función llamante.....	276
Una función que se llama a sí misma.....	278
<b>Ámbito de variables: locales y globales.....</b>	<b>279</b>
<b>Evaluación de una cadena de código JavaScript.....</b>	<b>281</b>
<b>JavaScript y los motores de búsqueda .....</b>	<b>282</b>
 <b>Capítulo 13:</b>	
<b>CSS, JavaScript y pseudo clases.....</b>	<b>285</b>
<b>Uso de CSS desde JavaScript.....</b>	<b>289</b>

Efecto de ampliación de texto.....289  
Ocultar y mostrar texto.....292  
Uso práctico como zoom de imágenes.....295  
Pseudo clases.....298

## Capítulo 14:

### Estilo en la impresión.....301

Tipos de medios .....302  
    Especificación del tipo de medio.....303  
        Regla @media.....303  
    Atributo media en etiqueta STYLE.....303  
    Atributo media en etiqueta LINK.....304  
    Importación de una hoja de estilo.....304  
Configuración de un documento para impresión.....305  
    Regla @page.....305  
    Propiedad size.....306  
    Propiedad margin.....306  
    Marcas de corte.....306  
    Control de los saltos de página.....307  
    Líneas viudas y de huérfanas.....308  
    Control de páginas pares e impares.....309  
Ejemplo de estilo de impresión.....309  
    Vista de impresión.....314

## Capítulo 15:

### Estilos de interfaz de usuario.....317

Cursores.....317  
Colores de la interfaz.....319  
Fuentes.....320

## Apéndice A:

### Codificación CSS.....323

Edición de código CSS.....324  
Validación de código CSS .....324  
Complementos para edición CSS en Firefox.....326

## Apéndice B:

**Guía rápida de CSS.....329**  
**Esquema de la sintaxis CSS.....329**  
**Tipos de medios.....329**  
**Unidades.....330**  
**Tipos de selectores.....330**  
**Propiedades agrupadas por funcionalidad.....331**  
    **Bordes.....331**  
    **Colores y fondo.....332**  
    **Contenido generado y listas .....332**  
    **Fuentes.....333**  
    **Márgenes.....333**  
    **Modelo visual (formato general).....333**  
    **Modelo visual (detalles de formato) .....334**  
    **Modelo visual (efectos).....334**

**Apéndice C:**  
**Práctica completa.....363**

**Índice.....401**

# Introducción

## ¿Qué es CSS?

CSS son las siglas de Cascade Style Sheets, es decir, hojas de estilo en cascada, y es una codificación que permite asignar estilo a los documentos HTML.

Los documentos HTML, en su origen, carecían de grandes posibilidades estéticas y en la versión 4.0 de HTML se añadió la posibilidad de crear estilos a partir de CSS. En realidad, antes de HTML 4.0 las capacidades de formateo de un documento HTML eran poco menos que espartanas y además, lo que se podía realizar requería un gran esfuerzo de codificación. Para resolver este problema, W3C creó el estándar CSS y actualmente la versión recomendada es CSS 2.1.

A partir de HTML 4, toda la codificación de estilo pudo separarse de la codificación HTML y almacenarse en archivos independientes. Esto hace que la tarea de creación de estilos pueda distribuirse mejor entre el equipo de desarrollo y además permite un mejor reaprovechamiento del código.

---

*Gracias a CSS un mismo documento HTML se puede visualizar aplicando diferentes estilos.*

---

## Enfoque del libro

Es un libro teórico práctico y lo normal es que se siga del modo ordenado por capítulo haciendo las prácticas propuestas para fijar los conocimientos teóricos explicados en cada uno de los temas.

Al final se incluye un trabajo práctico completo en el que se deben utilizar las diferentes técnicas estudiadas a lo largo de los capítulos.

CSS, como también sucede con el lenguaje HTML, es un tema que no exige demasiado desde el punto de vista teórico y todo parece muy lógico y simple. Pero no hay que dejarse engañar por esto y pensar que no hace falta realizar las prácticas. Es en las prácticas en donde se produce el verdadero aprendizaje y la recomendación es que se realicen todas las prácticas propuestas en el libro y que el lector además realice variaciones para comprobar los efectos de los cambios en el código.

## ¿Qué necesito saber para poder seguir el aprendizaje?

El lector debe conocer HTML; se da por supuesto que el lector comprende la codificación HTML y que quiere comenzar a dominar la codificación CSS.

---

*En el caso que el lector no domine HTML se recomienda que primero estudie y practique sus conceptos básicos antes de comenzar este libro.*

---

El tema CSS (hojas de estilo en cascada) se trata de modo detallado y no se da por supuesto que el lector tenga conocimientos previos y aunque cada tema se trata desde los detalles más básicos también se llega hasta el nivel avanzado.

## ¿Qué necesito tener para realizar las prácticas?

No se precisa ningún software en especial. Sirve cualquier ordenador que tenga instalado un navegador (al menos alguno de estos: Mozilla Firefox, Chrome, Opera, Safari o Internet Explorer).

Lo ideal es tener instalado más de un navegador porque, tal como se explica en el libro, no todos los navegadores y versiones se comportan de igual manera ante ciertas codificaciones CSS.

No se precisa conexión a Internet ya que los ejemplos se trabajan en modo local.

## Esquema general del libro

Este libro está compuesto de 15 capítulos, con teoría y práctica, y tres apéndices: la guía rápida, otras diversas tablas auxiliares y la práctica completa.

- **Introducción:** Es el presente capítulo en donde se indica a quién va dirigido este libro, con qué elementos se debe contar y cómo están organizados los temas. Se realiza un resumen de los temas tratados en cada uno de los capítulos.
- **Capítulo 1, Conceptos básicos:** Se enseñan los conceptos básicos de CSS y los diferentes modos de uso de los estilos. La etiqueta <LINK>, la etiqueta <STYLE> y la definición en línea o inline. Se trata el tema de la precedencia de las definiciones.
- **Capítulo 2, Definición de estilos:** Se explica la sintaxis básica de una declaración CSS, el concepto de selector y las distintas variantes para seleccionar los elementos del documento HTML. Se explica el uso de clases, identificadores y los selectores limitados por condiciones de jerarquía o por contenido de propiedades con valores determinados. Se analiza la estructura de un documento HTML para enseñar el mecanismo de herencia de propiedades entre padres e hijos y para el uso de selectores con condiciones de jerarquía padre-hijo y hermano. Finalmente se estudian las pseudo clases y los pseudo elementos que nos facilitan la definición de estilos.
- **Capítulo 3, Valores y unidades:** Antes de comenzar a estudiar las propiedades se realiza un estudio de los distintos tipos de valores que se pueden asignar a las diferentes propiedades. En este capítulo el lector se podrá familiarizar con las distintas unidades de medida, las diferentes maneras de definir un color, el modo de informar direcciones url, y las unidades especiales utilizadas por las funciones de audición.
- **Capítulo 4, Herencia y cascada:** En este capítulo se estudia la diferencia entre dos conceptos que suelen confundirse y que son diferentes: la herencia y el mecanismo de cascada. Se analiza cómo se definen las prioridades en la determinación del estilo que se aplica y el cálculo de la especificidad.
- **Capítulo 5, Propiedades de las fuentes:** En este capítulo se comienza realmente con la codificación CSS estudiando las propiedades de las fuentes. Después de una breve descripción de las características de las fuentes tipográficas se analizan las propiedades para definir la fuente, su

tamaño y su estilo, el espaciado entre letras y entre líneas y las fuentes del sistema que se pueden utilizar en nuestro documento.

- **Capítulo 6, Formateo de texto:** El texto es el contenido principal de los sitio web, por lo tanto CSS suministra una gran cantidad de propiedades para dar estilo al texto que complementan a las propiedades que actúan sobre las fuentes vistas en el capítulo anterior. En este capítulo se estudian las propiedades que actúan sobre la alineación horizontal y vertical, la sangría del texto, es espaciado de letras y de palabras, la transformación de texto sobre mayúsculas y minúsculas, los elementos de decoración del texto (subrayado, tachado, etc.) y los elementos flotantes que rodean al texto.
- **Capítulo 7, Formateo de listas con CSS:** En este capítulo se estudiarán las propiedades CSS que se pueden utilizar para dar estilo a las listas. Las listas pueden ser ordenadas y desordenadas y dentro de esta clasificación se pueden organizar de modo horizontal o vertical. Las listas se suelen complementar con numeración o con imágenes de viñetas, las que pueden ser estándar o personalizadas mediante el uso de imágenes.
- **Capítulo 8, Bordes, márgenes y rellenos:** La presentación de una página depende en gran medida de un buen uso de estos tres elementos: bordes, márgenes y rellenos. Estos elementos contribuyen en la delicada tarea de llevar la atención del visitante a las partes principales de la página. En este capítulo, después de explicar el concepto de caja de los elementos, estudiaremos todas las propiedades disponibles desde CSS para aplicar el estilo buscado.
- **Capítulo 9, Colores y fondos:** En este capítulo aprenderemos todo respecto a la gestión del color y las propiedades que nos permiten asignar el color a los elementos y al fondo. Los fondos de los elementos pueden tener un color uniforme o puede utilizar una imagen, esto también lo estudiaremos dentro de este tema. El uso de imágenes de fondo implica conocer el problema del desplazamiento y el posicionamiento. En las prácticas del capítulo se utilizan también, a modo de repaso, propiedades de estilo aprendidas en los capítulos anteriores.
- **Capítulo 10, Posicionamiento y diseño:** El posicionamiento de los elementos en una página es una de las necesidades esenciales en el diseño de un documento. En este capítulo estudiaremos los distintos tipos de posicionamiento (estático, relativo, absoluto, fijo), las propiedades y valores utilizados en CSS y las posibilidades para cambiar

y limitar los tamaños de los elementos con propiedades que no son las típicas de ancho y altura. Veremos también la capacidad para determinar la flotación de los elementos. La colocación de los elementos en capas permite crear diseños dinámicos, esto también lo veremos en este capítulo. Finalmente estudiaremos una de las propiedades más importantes para el diseño CSS: la propiedad display.

- **Capítulo 11, Contenido autogenerado:** Aunque CSS se utiliza para dar estilo también tiene ciertas capacidades para producir contenido. En este capítulo veremos la propiedad content y los pseudo elementos y objetos que la complementan para realizar su función. Estudiaremos la numeración automática y la utilización de contadores.
- **Capítulo 12, HTML dinámico y JavaScript:** Éste es un libro de CSS pero es necesario que un desarrollador de páginas web tenga también conocimiento de los lenguajes de script. En este capítulo se enseñan sólo los conceptos básicos de Javascript, es decir, de modo resumido lo mínimo necesario para comprender los fundamentos del lenguaje, que, sin ninguna duda, nos encontraremos en casi todas las páginas web.
- **Capítulo 13, CSS, JavaScript y pseudo clases:** Con la base de conocimientos que nos da el capítulo anterior sobre el lenguaje de programación JavaScript trataremos de sacarle provecho para que nos ayude a trabajar de modo dinámico con las propiedades de estilo CSS. Éste es un capítulo más práctico que teórico porque se trata de conocer la conexión entre CSS y JavaScript y después la cantidad de trucos que se pueden implementar dependen de la imaginación. Las pseudos clases no tiene relación con JavaScript sino que es una posibilidad de CSS para conseguir efectos dinámicos similares a los que se logran desde script.
- **Capítulo 14, Estilos en la impresión:** Hasta el momento hemos trabajado siempre sobre estilos para páginas web, es decir, para el medio screen; pero CSS también nos permite crear estilos para otros tipos de medios (handheld, impresoras braille, tv, teletipos, etc.). En este capítulo nos centraremos en el tipo más importante, después de screen: el estilo en la impresión y veremos como podemos controlar márgenes, saltos de página, etc, y todos los detalles de la impresión en papel.
- **Capítulo 15, Estilos de la interfaz de usuario:** En las interfaces de usuario se utiliza una serie de elementos, como iconos, colores y fuentes, que son propios del sistema y que poseen características que se definen a nivel de la plataforma al configurar el sistema, por ejemplo, los colores asignados a cada parte de la interfaz de usuario y el tipo de letra que se usa en cada área.



- **Apéndice A, Codificación CSS:** En este capítulo se mencionan algunas de las técnicas y programas utilizados para la edición de código CSS.
- **Apéndice B, Guía rápida de CSS:** En este apéndice se incluye un resumen de acceso rápido: tipos de unidades, tipos de medios, propiedades ordenadas alfabéticamente y ordenadas por funcionalidad.
- **Apéndice C, Práctica completa:** Al final del libro o a medida que se van estudiando los capítulos se podrá ir completando la práctica incluida en este apéndice hasta llegar a un caso completo del uso de CSS utilizando sus principales funcionalidades.

La práctica utiliza como base un archivo html creado de modo completo (practica.html) al que le faltan las indicaciones de estilo; la tarea será crear un archivo de instrucciones de estilo y modificar el código html para que se adapte a las instrucciones CSS que definiremos.

---

*El código que acompaña a este libro se puede descargar desde el sitio web [www.edgardandrea.com/libros](http://www.edgardandrea.com/libros).*

---

**Vistazo parcial del libro**  
**Salto de páginas...**

# 1. CSS: Conceptos básicos

La Web nace fundamentándose en HTML y en los documentos de texto pero a lo largo de su continua evolución fue necesitando mecanismos que mejorasen la presentación de los contenidos, que, por otra parte, cada vez son de origen más diverso, lo que ha permitido convertir a la Web en una plataforma de publicación. Una parte importante de esta evolución se consigue a partir de la implementación de las hojas de estilo en cascada (CSS; Cascade Style Sheets).

CSS actualmente es una herramienta infaltable en todos los sitios web ya que permite que los autores de contenido puedan ajustar la presentación para todo tipo de medio de difusión.

En este capítulo revisaremos los aspectos básicos de CSS y en los capítulos siguientes iremos conociendo todos los elementos de este estándar conjuntamente con las prácticas necesarias para dominar el tema.

## ¿Para qué se necesitan los estilos?

Antes que nada cabe señalar que los estilos existen aunque no los utilicemos mediante un estándar como CSS.

Cuando creamos el título de un artículo en un documento Word y a la letra le asignamos un tamaño un poco más grande que al resto y, si es necesario, le asignamos letra negrita o cursiva estamos aplicando un estilo.

---

*El concepto de estilo utilizado en los procesadores de texto, como Word o Libre Office, es muy similar al estilo aplicado mediante CSS, por lo que si se entienden los fundamentos de uno no costará mucho aprender el otro.*

---

En la publicación electrónica los estilos se aplican mediante una codificación dinámica del texto que requiera algún tipo de formateo. Por ejemplo, supongamos que tenemos un texto de un artículo, como en el ejemplo mencionado anteriormente, y queremos que el título se destaque del resto, a esa parte del texto se le asigna un estilo "Encabezamiento 1", que podremos utilizar también toda vez que haga falta dentro del documento.

---

*El estilo denominado "Encabezamiento 1" lo habremos definido una única vez y lo podremos utilizar tantas veces como sea necesario.*

---

La alternativa a este enfoque es asignar el estilo elemento por elemento dentro del documento HTML; en este caso, un cambio en el formato del estilo requerirá que recodifiquemos múltiples definiciones.

---

*Cuando se utiliza CSS, una etiqueta HTML cualquiera, por ejemplo, <h1> o <p>, no debería incluir información sobre cómo será visualizado su contenido, solamente debería servir para marcar la estructura del documento. La información de estilo debe incluirse en una hoja de estilo.*

---

## Asignación de estilo por el autor del documento

Cuando un autor crea un documento tiene dos opciones:

1. Codificar cada detalle del estilo en cada elemento HTML
2. Marcar cada elemento simplemente con un identificador que haga referencia al estilo

Esta segunda opción es la mejor, no sólo porque se requiere menos código y el mantenimiento es más sencillo sino porque de esta manera el código se divide en dos partes bien diferenciadas y se puede dejar la tarea de presentación a una persona más especializada en el aspecto estético.

Aunque no es la única opción disponible, los estilos se pueden definir en un documento específico e independiente que se denomina hoja de estilo. Esto permite una clara separación entre contenido y presentación. Una de las ventajas de este enfoque es que podemos tener un mismo documento (contenido) preparado para ser presentado en diferentes medios con sólo

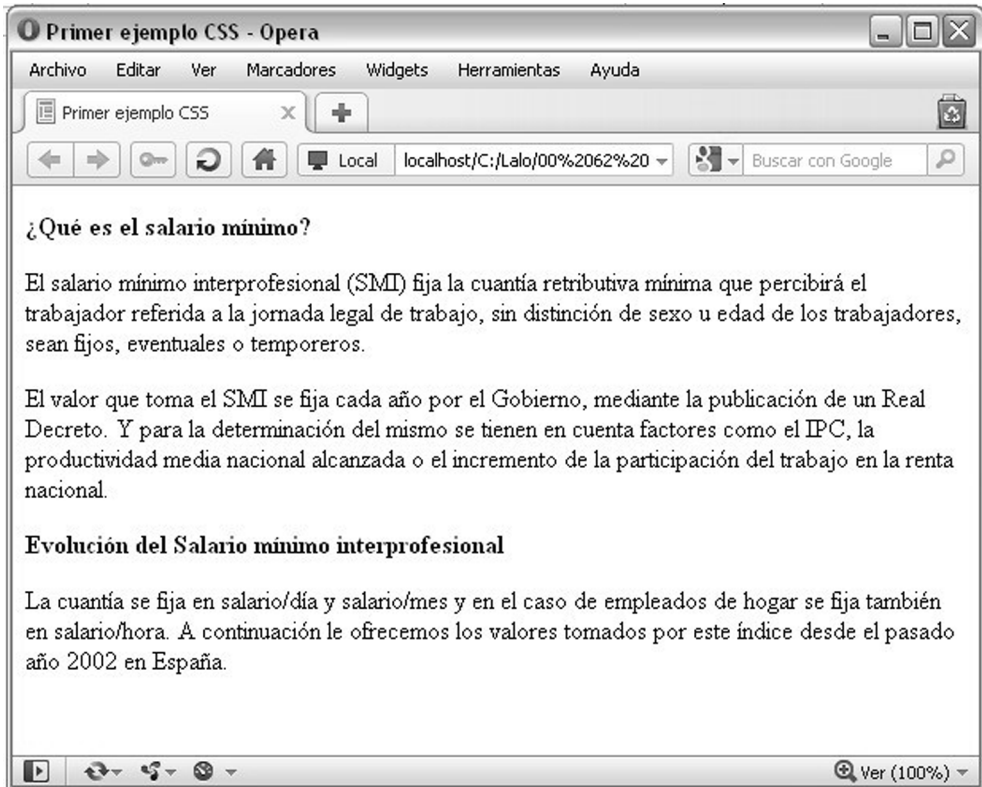
cambiar la hoja de estilo, por ejemplo, un estilo para su visualización en pantalla y otro para su impresión en papel.

## Interrelación HTML y estilos

Veamos primero cómo se aplica el estilo definido en línea en un documento HTML:

```
<!DOCTYPE html>
<html>
<head>
<title>Primer ejemplo CSS</title>
</head>
<body>
<p><b>¿Qué es el salario mínimo?</b></p>
<p>El salario mínimo interprofesional (SMI) fija la
cuantía retributiva mínima que percibirá el trabajador
referida a la jornada legal de trabajo, sin distinción
de sexo u edad de los trabajadores, sean fijos,
eventuales o temporeros.</p>
<p>El valor que toma el SMI se fija cada año por el
Gobierno, mediante la publicación de un Real Decreto. Y
para la determinación del mismo se tienen en cuenta
factores como el IPC, la productividad media nacional
alcanzada o el incremento de la participación del
trabajo en la renta nacional.</p>
<p><b>Evolución del Salario mínimo
interprofesional</b></p>
<p>La cuantía se fija en salario/día y salario/mes y en
el caso de empleados de hogar se fija también en
salario/hora. A continuación le ofrecemos los valores
tomados por este índice desde el pasado año 2002 en
España.</p>
</body>
</html>
```

Si esto lo visualizamos en el navegador veríamos lo siguiente:



Tal como se puede observar hay dos cabeceras que se destacan con letra negrita.

---

*Por ahora no nos preocupemos por los detalles de la codificación, lo importante es captar los diferentes modos de aplicar los estilos.*

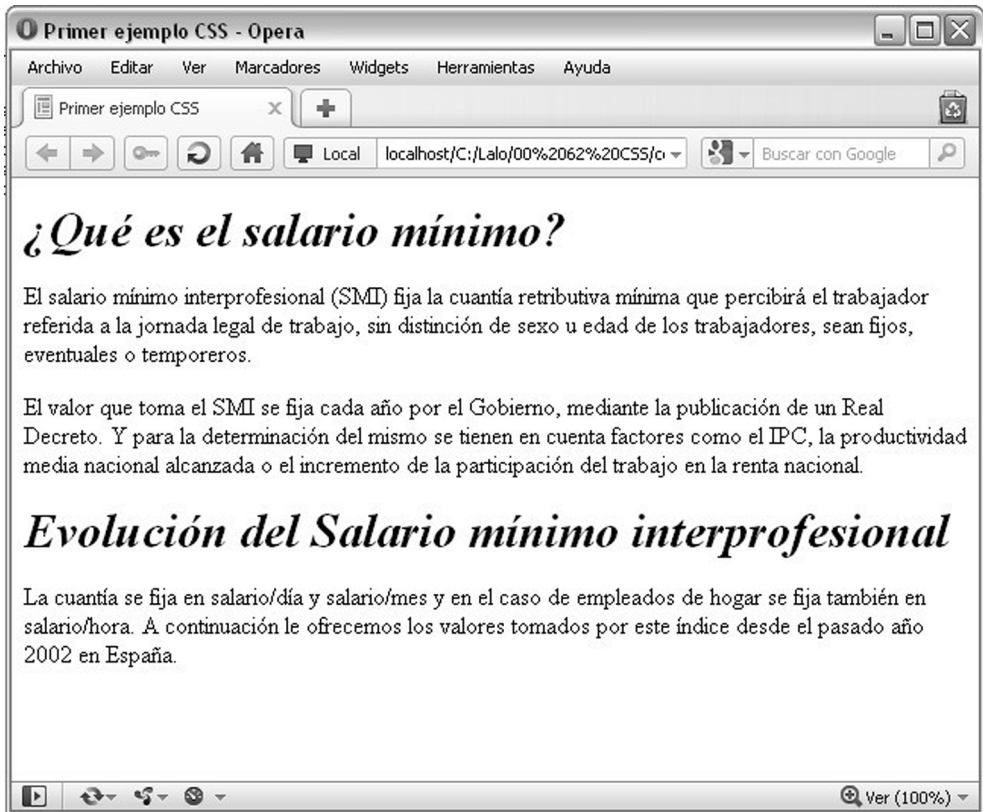
---

Ahora supongamos que se quiere que esas dos cabeceras sean en cursiva y en una letra más grande, sin utilizar CSS (sólo HTML puro), tendríamos que modificar los dos párrafos con un código similar a este:

```
...
<p><font size="+3"><i><b>¿Qué es el salario
mínimo?</b></i></font></p>
```

...

Este cambio provoca esta visualización:



Los cambios han afectado dos sitios, ya que hay sólo dos cabeceras, pero el lector ya se puede dar una idea del trabajo que significa intercalar las etiquetas en el sitio correcto y después no olvidarse de las etiquetas de cierre. Pensemos qué trabajo tendríamos si el documento tuviese 20 cabeceras o más.

## Primera definición CSS

Ahora veremos cómo se haría lo mismo pero con instrucciones CSS:

```
<!DOCTYPE html>
<html>
<head>
<title>Primer ejemplo CSS con estilo</title>
<style type="text/css">
    p.cabecera {font-weight: bold; font-style: italic;
                font-size: xx-large;}
</style>
</head>
```

**Vistazo parcial del libro**  
**Salto de páginas...**



## 2: Definición de estilos

CSS funciona mediante la definición de reglas; dicho en otros términos, con declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML.

---

*Ya hemos visto en el capítulo anterior que estas reglas que definen el estilo pueden estar en un archivo externo, en un bloque interno o definidos en línea dentro del elemento HTML; ahora nos centraremos en el formato de las reglas.*

---

Cada regla está compuesta de dos partes:

- **Selector:** Es el enlace entre el documento y el estilo; especifica los elementos afectados por la declaración que se indicará a continuación. El selector puede indicar distintos tipos de información: en su modo más simple puede ser una etiqueta HTML, una clase definida por el usuario o un nombre de un identificador. Pero también puede indicarse de modo compuesto, por ejemplo, etiqueta y clase.
- **Declaración:** Establece las propiedades y valores que definen el estilo. Empieza por una llave de apertura "{" y sigue por una lista de uno o más pares "propiedad: valor" finalizados cada uno por punto y coma y finalmente con una llave de cierre "}".

```
SELECTOR {DECLARACIÓN....}
```

La declaración está compuesta por una o más propiedades, separadas por punto y coma; cada propiedad normalmente tiene un valor asignado (puede tener más de un valor).

Entre el nombre de la propiedad y el o los valores se utiliza el separador de dos puntos.

En este ejemplo, el selector es el elemento BODY y la declaración posee cuatro propiedades, tres con un valor asociado y la última (border) con tres valores asociados.

```
BODY {color: red; background: blue; font-family:
      Arial; border: thin dotted black;}
```

## Legibilidad

Para que el código CSS se pueda leer con mayor facilidad se puede escribir de la siguiente manera y el resultado será el mismo que en el caso anterior:

```
BODY      {color: red;
           background: blue;
           font-family: Arial;
           border: thin dotted black;}
```

---

*Al utilizar una línea para cada propiedad el código es más fácil de mantener.*

---

## Etiqueta <STYLE>

La etiqueta <STYLE> representa el modo más directo para aprender el uso de las hojas de estilo, aunque ya hemos visto que no es el único mecanismo para implementar CSS dentro de un documento HTML (o XML).

<STYLE> es una de las etiquetas permitidas dentro del bloque <HEAD> de un documento HTML.

---

*La etiqueta <STYLE> por sí misma no produce contenido para el usuario.*

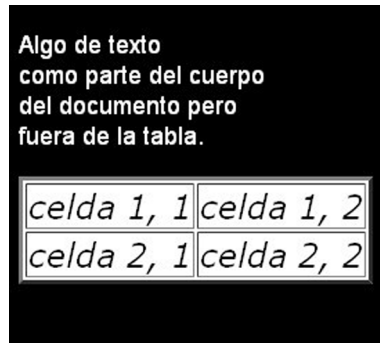
---

A continuación definiremos un estilo para la etiqueta <BODY>, es decir, todo el cuerpo del documento seguirá el estilo que definamos, salvo que en alguna etiqueta interior, encerrada dentro de <BODY>, haya una definición diferente. Además añadiremos una definición de estilo para el elemento <TABLE>.

Veamos cómo es el código (es el archivo cap0200.html dentro del código que se puede descargar desde [www.edgardandrea.com/libros](http://www.edgardandrea.com/libros)):

```
<!DOCTYPE html>
<html>
<head>
  <title>Uso básico de estilo</title>
  <style type="text/css">
    body {color: white; background: black;
          font-family: Arial; font-size: large}
    table {color: black; background: white;
          font-family: Verdana; font-size: x-large;
          font-style: italic}
  </style>
</head>
<body>
<p>Algo de texto <br />
como parte del cuerpo <br />
del documento pero <br />
fuera de la tabla. </p>
<table border=3>
<tr><td>celda 1, 1</td><td>celda 1, 2</td></tr>
<tr><td>celda 2, 1</td><td>celda 2, 2</td></tr>
</table>
</body>
</html>
```

El resultado de este código es el siguiente:



En la etiqueta `<STYLE>` se utiliza el atributo `type` para indicar qué lenguaje se utiliza para gestionar la hoja de estilo; si no se especifica este atributo se presupone `text/css`.

### Práctica

- ➔ Utilice el ejemplo anterior (`cap0200.html`) para comprobar el efecto de aplicar o no el estilo. Elimine todo el bloque `<STYLE>` y compare la diferencia.

## Atributos en la etiqueta `<STYLE>`

Además del atributo `type`, la etiqueta `<STYLE>` puede utilizar estos otros atributos:

- **media**: Indica el dispositivo de visualización en el que se aplica el estilo. De manera predeterminada, se asume `all`.
- **title**: Para identificar las hojas de estilo por un nombre.
- **scoped**: Es un valor booleano; si se informa, indica que los estilos van dirigidos al subárbol dependiente del elemento padre al que se define el estilo; el caso contrario a esto sería aplicarlo a todo el documento.

## Definiciones dentro del bloque `<STYLE>`

Dentro del bloque `<STYLE></STYLE>` se pueden incluir una o más definiciones de estilo; en este ejemplo hemos incluido dos, una para el cuerpo y otra para las tablas que pudieran aparecer en el documento.

Tal como se puede observar en el código, se utiliza la estructura siguiente:

```
selector {propiedad: valor; ...propiedad: valor}
```

El selector identifica la parte del documento afectada por la declaración de estilo. Si esa parte no existiese en el documento, no se genera ningún error, por ejemplo, si no hubiese ninguna tabla simplemente el estilo no se aplica.

---

*Si definimos un estilo no existe obligación de utilizarlo en el documento, eso no provoca ningún error.*

---

Cada nombre de propiedad se separa de su valor con dos puntos y de la siguiente propiedad con punto y coma. Toda la declaración se encierra entre llaves.

En este ejemplo, se utiliza un selector por declaración, pero podríamos incluir una lista de selectores, a los que se les aplicará un mismo estilo, por ejemplo:

```
h1, h2, h3, h4 {color: red;
                background: white;
                font-family: Helvetica ;
                font-size: x-large}
```

Con este estilo todos los encabezamientos, desde h1 a h4, compartirán las mismas propiedades especificadas.

Con el mismo resultado final, esto se podría haber codificado de la siguiente manera:

```
h1      {color: red;
        background: white;
        font-family: Helvetica ;
        font-size: x-large}
h2      {color: red;
        background: white;
        font-family: Helvetica ;
        font-size: x-large}
h3      {color: red;
        background: white;
        font-family: Helvetica ;
        font-size: x-large}
h4      {color: red;
        background: white;
        font-family: Helvetica ;
        font-size: x-large}
```

**Vistazo parcial del libro**  
**Salto de páginas...**

## 4: Herencia y cascada

El funcionamiento de las hojas de estilo en cascada tiene uno de sus pilares en el mecanismo de la herencia, que a veces se confunde con el mecanismo de cascada que es un concepto diferente. En este capítulo aprenderemos cuál es la diferencia entre esos dos mecanismos.

La aplicación de estilos puede generarse desde distintos orígenes y esto puede llegar a complicar seriamente la determinación del estilo que finalmente se aplica debido a los conflictos se pueden crear. No obstante, existe un conjunto de reglas que nos permiten determinar qué estilo se aplica finalmente. Para llegar a dominar cómo se aplican las reglas es necesario conocer los posibles orígenes de las definiciones. Además debemos estudiar las reglas de la herencia, el mecanismo en cascada y la definición de la especificidad.

---

*Pero antes de pensar que éste es un tema muy complejo vayamos paso a paso y veremos que todo tiene su lógica.*

---

### Herencia

Dentro del ámbito de un documento HTML codificado con CSS, la herencia es el mecanismo que permite asignar a los hijos algunas de las propiedades de su padre.

```
<!DOCTYPE html>
<html>
<head>
<title>Herencia</title>
<style>
div {color: grey;}
</style>
</head>

<body>
<div>
<h1>Tabla 1</h1>
<table>
<tr><td>Fila 1 Celda 1</td><td>Fila 1 Celda 2</td>
</tr>
<tr><td>Fila 2 Celda 1</td><td>Fila 2 Celda 2</td>
</tr>
</table>
</div>

<p>Texto en negro
</p>

</body>
</html>
```

En este ejemplo se puede comprobar el efecto de la herencia. El elemento `<DIV>` es el que tiene un estilo definido y sus hijos reciben el mismo estilo (color gris) por el mecanismo de herencia. En cambio, el párrafo `<P>` está fuera del bloque `<DIV>` y, por lo tanto, el color del texto es el definido por defecto por el agente de usuario.

El cuerpo `<BODY>` tampoco recibe la herencia porque no es hijo de `<DIV>` sino su padre.





## Cascada

El concepto cascada es diferente al concepto herencia, se refiere a la capacidad de definir los estilos a partir de diferentes orígenes, que se unifican para dar como resultado un estilo determinado.

El origen de la definición de los estilos puede ser cualquiera de los siguientes y ordenados por prioridad de elección:

- Estilos definidos por el autor
- Estilos de usuario
- Estilos del agente de usuario

Veamos cada uno de estos orígenes.

### Estilos definidos por el autor

Son los estilos que el autor incluye en el propio documento, sea definidos en la parte superior del documento, enlazados en hojas de estilos independientes o en elementos definidos en línea. La suma de estos estilos dan como resultado la presentación elegida por el autor.

Estos estilos son los que hemos estudiado hasta el momento, definido en un bloque <STYLE>, en una etiqueta <LINK> que enlaza con un archivo css o un estilo en línea. También hemos estudiado cómo se aplica la precedencia.

### Estilos de usuario

Son los estilos especificados por el usuario del documento. Estos estilos se eligen a partir de un conjunto de hojas de estilo locales que se seleccionan en las opciones del navegador.

Para especificar una hoja de estilo de usuario con Internet Explorer se puede utilizar la opción Accesibilidad de Opciones de Internet.



## Estilos del agente de usuario

Son los estilos aplicados de manera predeterminada cuando no hay otros estilos especificados para un elemento del documento. Suelen ser definiciones básicas, por ejemplo, texto en color negro sobre fondo blanco, etc.

La prioridad de aplicación de los estilos sigue el orden especificado anteriormente, salvo que se utilice la palabra clave `!important` que puede crear una excepción y que hemos estudiado en el capítulo 2.

```
div {color: red !important;}
```

Con esta codificación se puede revertir el orden de precedencia (autor-usuario) y dar prioridad a la definición del usuario.

**Vistazo parcial del libro**  
**Salto de páginas...**

## 6: Formateo del texto

Aunque los contenidos multimedia están tomando cada vez mas sitio dentro de las páginas web, el contenido por excelencia sigue basándose en el texto. CSS tiene amplias capacidades para ofrecer funciones de formateo de texto y este capítulo lo dedicaremos a estudiar las propiedades que nos pueden gestionar la presentación del texto dentro del documento, lo que implica caracteres, palabras, párrafos y espacios.

### **Alineación horizontal del texto (text-align)**

El texto se puede alinear tanto de modo horizontal o vertical. Para la alineación horizontal CSS utiliza la propiedad text-align, la que puede tomar cuatro valores: left (valor predeterminado), right, center o justify. El efecto de estos valores de justificación del texto lo veremos en la siguiente práctica.

La alineación consiste en colocar el texto sobre uno de los márgenes (left o right), centrado u ocupando todo el ancho disponible (justify). En este ejemplo utilizamos párrafos <P>, pero la alineación también se puede referir al contenido interno de las celdas de una tabla.

```
<!DOCTYPE html>
<html>
<head>
<title>Familia de fuentes</title>

<style>

#idleft {text-align: left;}
#idright {text-align: right;}
#idcenter {text-align: center;}
#idjustify {text-align: justify;}

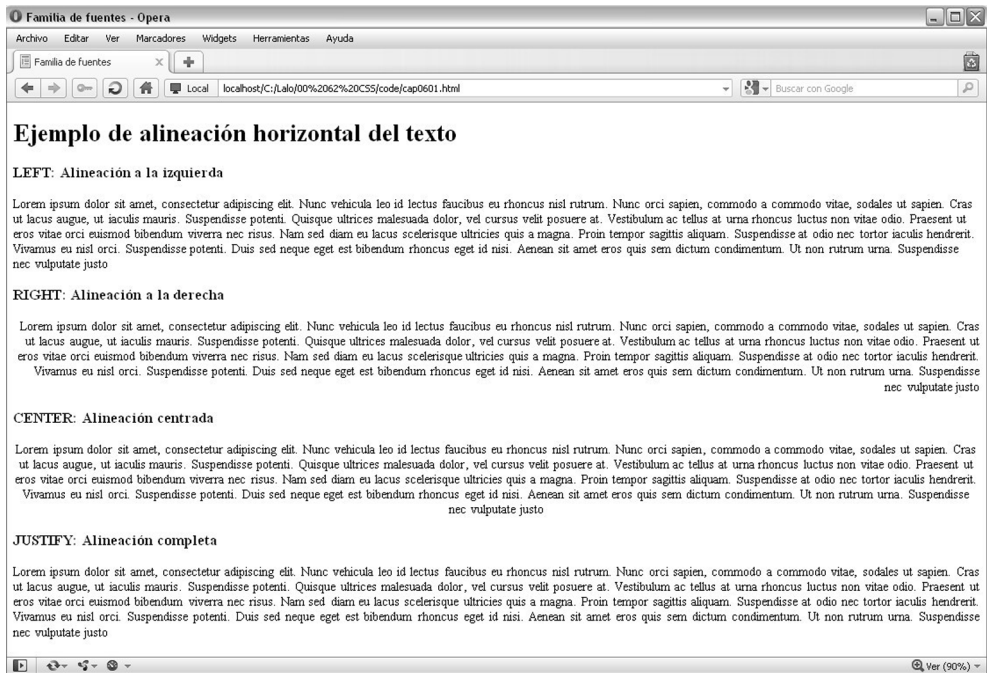
</style>

</head>
<h1>Ejemplo de alineación horizontal del texto</h1>
<body>
<div>
<h3>LEFT: Alineación a la izquierda</h3>
<p id=idleft>
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Nunc vehicula leo id lectus faucibus eu rhoncus nisl
rutrum. Nunc orci sapien, commodo a commodo vitae,
sodales ut sapien. Cras ut lacus ...
</p>
<h3>RIGHT: Alineación a la derecha</h3>
<p id=idright>
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Nunc vehicula leo id lectus faucibus eu rhoncus nisl
rutrum. Nunc orci sapien, commodo a commodo vitae,
sodales ut sapien. Cras ut lacus ...
</p>
<h3>CENTER: Alineación centrada</h3>
<p id=idcenter>
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Nunc vehicula leo id lectus faucibus eu rhoncus nisl
rutrum. Nunc orci sapien, commodo a commodo vitae,
sodales ut sapien. Cras ut lacus ...
</p>
<h3>JUSTIFY: Alineación completa</h3>
<p id=idjustify>
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Nunc vehicula leo id lectus faucibus eu rhoncus nisl
rutrum. Nunc orci sapien, commodo a commodo vitae,
sodales ut sapien. Cras ut lacus ...
```

```

</p>
</body>
</html>

```




---

*Si se quiere trabajar sobre este ejemplo, abrir el archivo es `cap0601.html`.*

---

## Alineación vertical del texto (vertical-align)

La alineación vertical del texto tiene más posibilidades que la alineación horizontal.

La propiedad que nos permite controlar desde CSS la alineación vertical es `vertical-align` y puede tomar los siguientes valores:

- **baseline**: Es el valor predeterminado. El texto utiliza la línea base para alinearse respecto a otros objetos que lo rodean.
- **sub**: Este valor hace que el texto descienda hasta el nivel utilizado por los subíndices. Este valor no afecta el tamaño del texto sino su posicionamiento vertical.

- **super**: Este valor hace que el texto ascienda hasta el nivel utilizado por los superíndices. Este valor no afecta el tamaño del texto sino su posicionamiento vertical.
- **top**: Este valor hace que la parte superior del cuadro en donde se inscribe el elemento quede alineado con la parte superior del cuadro de su elemento padre.
- **text-top**: Este valor hace que la parte superior del cuadro en donde se inscribe el elemento quede alineado con la parte superior del texto de su elemento padre.
- **middle**: Este valor hace que el texto quede alineado usando el medio del texto y la línea media de los objetos que lo rodean.
- **bottom**: Este valor hace que la parte inferior del cuadro en donde se inscribe el elemento quede alineado con la parte inferior del cuadro de su elemento padre.
- **text-bottom**: Este valor hace que la parte inferior del cuadro en donde se inscribe el elemento quede alineado con la parte inferior del texto de su elemento padre.
- **length**: El valor indicado en esta propiedad indica que el elemento suba (si es positivo) o baje (si es negativo) la cantidad especificada.
- **percentage**: El valor indicado en esta propiedad indica que el elemento suba (si es positivo) o baje (si es negativo) el porcentaje indicado respecto a la altura de la línea del elemento.

En los siguientes ejemplos (archivo cap0602.html) queda claro el efecto de cada valor.

En este código se define una letra más pequeña para el texto alineado verticalmente para que se visualice mejor el efecto, además de colocar un borde de 1px para que se compruebe perfectamente la alineación:

```
<!DOCTYPE html>
<html>
<head>
<title>Alineación vertical</title>

<style>
body *      {border: 1px solid black;}
p           {font-size: 150%;}
p *        {font-size: 50%;}
.baseline   {vertical-align: baseline;}
.sub        {vertical-align: sub;}
```

```
.super      {vertical-align: super;}
.top        {vertical-align: top;}
.text-top   {vertical-align: text-top;}
.middle     {vertical-align: middle;}
.bottom     {vertical-align: bottom;}
.text-bottom {vertical-align: text-bottom;}
.length     {vertical-align: -1.5em;}
.percentage {vertical-align: -150%;}
```

```
</style>
```

```
</head>
```

```
<h1>Ejemplo de alineación vertical del texto</h1>
```

```
<body>
```

```
<p>Baseline: xxxx<span class=baseline>yyyyyy</span>zzz</p>
```

```
<p>Sub: xxxx<span class=sub>yyyyyy</span>zzz</p>
```

```
<p>Super: xxxx<span class=super>yyyyyy</span>zzz</p>
```

```
<p>Top: xxxx<span class=top>yyyyyy</span>zzz</p>
```

```
<p>Text-top: xxxx<span class=text-top>yyyyyy</span>zzz</p>
```

```
<p>Middle: xxxx<span class=middle>yyyyyy</span>zzz</p>
```

```
<p>Bottom: xxxx<span class=bottom>yyyyyy</span></p>
```

```
<p>Text-bottom: xxxx<span class=text-bottom>yyyyyy</span>zzz</p>
```

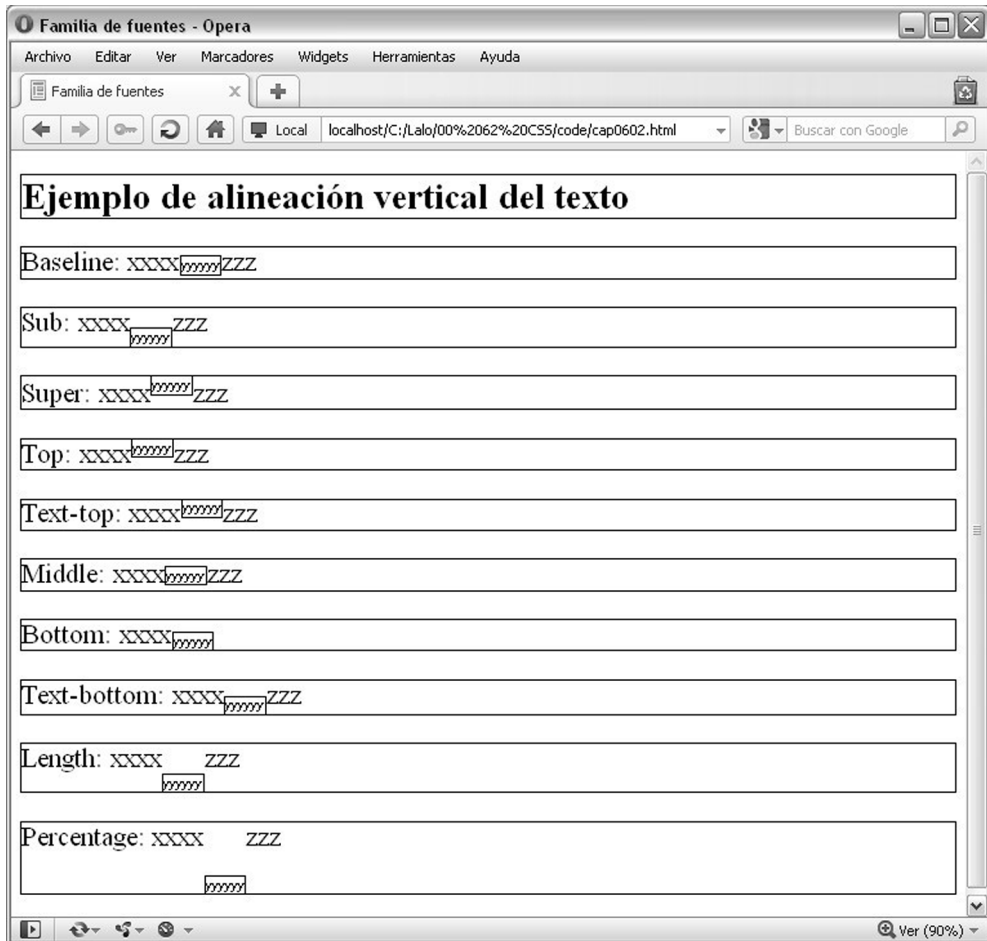
```
<p>Length: xxxx<span class=length>yyyyyy</span>zzz</p>
```

```
<p>Percentage: xxxx<span class=percentage>yyyyyy</span>zzz</p>
```

```
</body>
```

```
</html>
```





## Sangría del texto (text-indent)

Cuando hablamos de sangría nos referimos exclusivamente al margen que se fuerza en la primera línea del párrafo. Si se quieren definir otros márgenes se debe utilizar otro conjunto de propiedades margin que veremos más adelante.

La propiedad text-indent nos permite definir la sangría de la primera línea del texto. Se puede asignar con valores de longitud (cm, in, px, etc.) o porcentajes respecto al ancho del elemento.

En este ejemplo (cap0603.html) vemos el efecto de esta propiedad:

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

```
<head>
<title>Sangría de texto</title>
<style>
#idindent1 {text-indent: 50px;}
#idindent2 {text-indent: 30%;}
#idindent3 {text-indent: 2cm;}
</style>
</head>

<h1>Ejemplo de sangría</h1>
<body>
<div>
<h3>50 px de sangría</h3>

<p id=idindent1>
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Nunc vehicula leo id lectus faucibus eu rhoncus nisl
rutrum. Nunc orci sapien, commodo a commodo vitae,
sodales ut sapien. Cras ut lacus...
</p>

<h3>30% de sangría</h3>
<p id=idindent2>
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Nunc vehicula leo id lectus faucibus eu rhoncus nisl
rutrum. Nunc orci sapien, commodo a commodo vitae,
sodales ut sapien. Cras ut lacus...
</p>

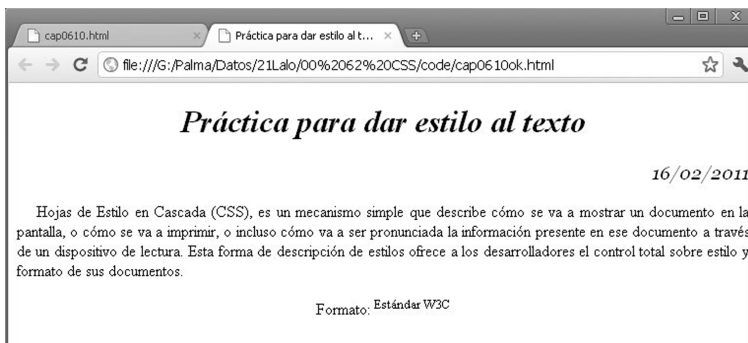
<h3>2cm de sangría</h3>
<p id=idindent3>
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Nunc vehicula leo id lectus faucibus eu rhoncus nisl
rutrum. Nunc orci sapien, commodo a commodo vitae,
sodales ut sapien. Cras ut lacus...
</p>

</body>
</html>
```



## Práctica

- Utilizando como base el texto del archivo cap0610.html, sin estilo, aplicar las propiedades de estilo de texto adecuadas para conseguir los efectos que se muestran en la siguiente figura (observar la justificación utilizada):



**Vistazo parcial del libro**  
**Salto de páginas...**

## 10: Posicionamiento y diseño

CSS nos suministra varios mecanismos, para ser más precisos son 4 maneras diferentes, para resolver el posicionamiento de los elementos en una página. El método que elijamos dependerá principalmente de cómo queremos que se relacione la posición del elemento con los otros elementos que lo rodean y cómo puede afectarlos. Ahora estudiaremos estos cuatro modelos de posicionamiento.

- Posicionamiento estático
- Posicionamiento relativo
- Posicionamiento absoluto
- Posicionamiento fijo

---

*No todos los agentes de usuario responden de la misma manera a las propiedades de posicionamiento; antes de adoptar un determinado modelo de posicionamiento se debe probar en los distintos tipos de navegadores (y sus versiones) para confirmar el comportamiento.*

---

### **Propiedades relacionadas con el posicionamiento**

Las principales propiedades de esta categoría son las siguientes:

- **position:** Esta propiedad no define la posición sino el modo en que se colocan los elementos de un documento HTML en la ventana del navegador. Los valores permitidos son los siguientes:
  - *static:* Valor predeterminado, el elemento se coloca según el flujo natural del los elementos, sin afectarle el valor de otras propiedades, como: top, bottom, etc.).
  - *fixed:* El elemento se colocará en una posición fijada por las propiedades de distancias a los bordes de la ventana (top, bottom, left y right).
  - *relative:* El elemento se colocará en una posición relativa a la que le correspondería en posicionamiento normal y en base a distancias indicadas por las propiedades top, bottom, left o right, que pueden tener valores negativos o positivos.
  - *absolute:* El elemento se colocará en una posición relativa al primer elemento padre (de nivel superior en la jerarquía) que tenga un posicionamiento diferente a static y en base a distancias indicadas por las propiedades top, bottom, left o right, que pueden tener valores negativos o positivos.
- **top, bottom, left y right:** Estas propiedades establecen el margen del elemento respecto a su posición normal. Puede tener los valores: auto (valor predeterminado y que asume el valor normal que le corresponda al elemento), unidades de longitud o porcentajes referidos al tamaño del elemento contenedor. Por ejemplo: top:0 ; left:0; position: absolute, correspondería a la esquina superior izquierda de la ventana.
- **z-index:** Indica el nivel de profundidad en la colocación superpuesta de elementos en la ventana (pila de elementos). Cuanto mayor sea el valor asignado más arriba aparecerá. Si tenemos dos elementos superpuestos con valor 1 y 2, el elemento con valor 2 se verá por encima del elemento de valor 1.
- **visibility:** Especifica si el elemento es visible o no. El valor predeterminado es visible, los otros valores posibles son: hidden (oculto) o collapse (que se aplica para ocultar filas y columnas de tablas).

## Posicionamiento de los elementos

El posicionamiento de los elementos se controla con cuatro propiedades específicas que indican las distancias a cada uno de los cuatro lados: top, bottom, left y right.

Estas propiedades son fáciles de entender pero su comportamiento depende fundamentalmente del valor de la propiedad position (si no se informa se asume static).

---

*Importante: veremos que el efecto de estas propiedades depende del modelo de posicionamiento elegido (static, relative, absolute o fixed), tal como estudiaremos más adelante.*

---

Para informar cualquiera de estas propiedades se utiliza el siguiente formato:

```
<lado>: <distancia> | <porcentaje>;
```

Por ejemplo;

```
top: 50px;  
left: 30px;
```

Cuando se utiliza una unidad de distancia (por ejemplo, px, cm, etc) es el valor en el lado indicado (top, bottom, etc.) hasta la línea en donde debería estar en el flujo normal (cuando es posicionamiento relative) o hasta la línea del lado correspondiente en la ventana de visualización o viewport (cuando es posicionamiento absolute o fixed). Esto lo veremos en más detalle al estudiar los distintos posicionamientos.

---

*Si se indica un porcentaje, será un valor relativo a su ancho (para right o left) o su altura (para top o bottom). Pero como el uso de porcentajes puede deparar sorpresas debido a la lógica aplicada, después que estudiemos los cuatro modelos veremos unos ejemplos para que quede claro el uso de este tipo de valor.*

---

### Práctica

➔ Usar el archivo cap10000.html como base para informar las propiedades top, left, bottom, right en alguno de los párrafos (por ejemplo, en la clase p.c1) y observar el comportamiento.

---

*¿No sucede nada? Correcto, esas propiedades no tienen efecto con position static, que es el valor predeterminado.*

---

## Posicionamiento estático (static)

El posicionamiento estático de los elementos es el más fácil de comprender: los elementos se colocan en el sitio que les corresponde dentro del flujo natural.

Veamos un ejemplo (archivo cap1008.html) y quedará claro:

```
<!DOCTYPE html>
<html>
<head>
<title>Posicionamiento estático</title>

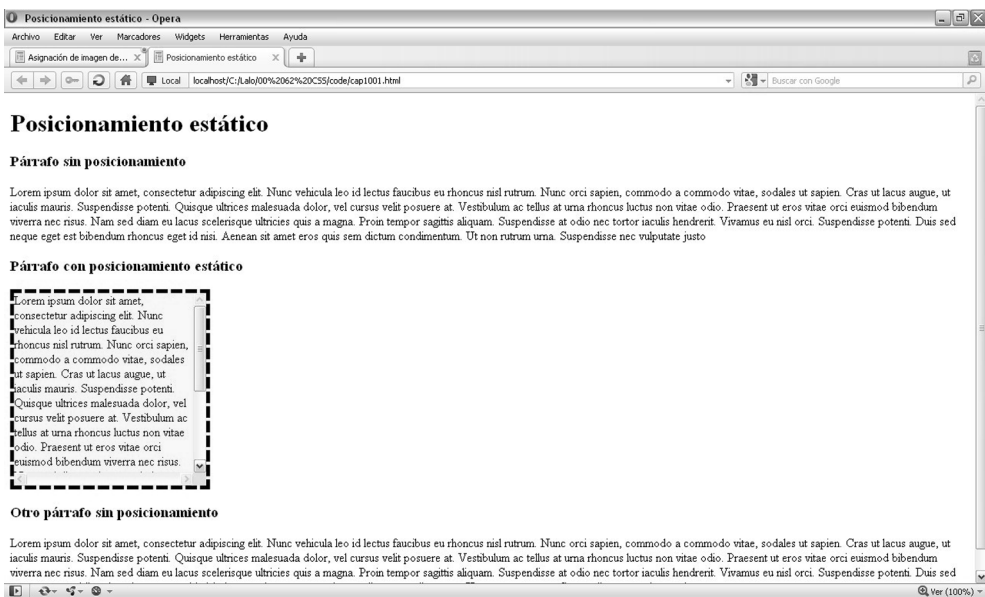
<style>
#id1 {width: 250px;
      height: 250px;
      border: 4pt black dashed;
      background-color: yellow;
      position: static;
      overflow: scroll;
      top: 3000px;}
</style>

</head>
<h1>Posicionamiento estático</h1>
<body>
<div>
<h3>Párrafo sin posicionamiento</h3>
<p>
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Nunc vehicula leo id lectus faucibus eu rhoncus nisl
rutrum. Nunc orci sapien, commodo a commodo vitae,
sodales ut sapien. Cras ut lacus augue...
</p>
<h3>Párrafo con posicionamiento estático</h3>
<p id=id1>
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Nunc vehicula leo id lectus faucibus eu rhoncus nisl
rutrum. Nunc orci sapien, commodo a commodo vitae,
sodales ut sapien. Cras ut lacus augue...
</p>
<h3>Otro párrafo sin posicionamiento</h3>
<p>
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Nunc vehicula leo id lectus faucibus eu rhoncus nisl
```



```
rutrum. Nunc orci sapien, commodo a commodo vitae,
sodales ut sapien. Cras ut lacus augue...
</p>
</body>
</html>
```

Hemos colocado una propiedad top con un valor muy alto (3000px) para mostrar que en el posicionamiento estático esta propiedad (y también sucede con la propiedad left) no se toma en cuenta; lo que manda es el posicionamiento static que le corresponde al elemento en el flujo del texto o del bloque .



Los otros dos párrafos se comportan de igual modo, ya que static es la opción por defecto.

## Posicionamiento relativo (relative)

El posicionamiento relativo se utiliza para colocar el elemento en un posición diferente a la que le tocaría siguiendo el flujo natural. La nueva posición utiliza las propiedades top y left para desplazar el elemento respecto al lugar que le tocaría normalmente.

*Importante:* el valor de `top` y `left` no es una distancia respecto a los bordes de la ventana, sino respecto a la posición que le tocaría normalmente.

---

Veamos un ejemplo similar al anterior (archivo `cap1009.html`), pero con posicionamiento relativo de 20px (top) y 300px (left):

```
<!DOCTYPE html>
<html>
<head>
<title>Posicionamiento relativo</title>

<style>
#id1 {width: 250px;
      height: 250px;
      border: 4pt black dashed;
      background-color: yellow;
      position: relative;
      overflow: scroll;
      top: 20px;
      left: 300px;}
</style>

</head>
<h1>Posicionamiento relativo</h1>
<body>
<div>
<h3>Párrafo sin posicionamiento</h3>
<p>
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Nunc vehicula leo id lectus faucibus eu rhoncus nisl
rutrum. Nunc orci sapien, commodo a commodo vitae,
sodales ut sapien. Cras ut lacus augue...
</p>
<h3>Párrafo con posicionamiento relativo</h3>
<p id=id1>
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Nunc vehicula leo id lectus faucibus eu rhoncus nisl
rutrum. Nunc orci sapien, commodo a commodo vitae,
sodales ut sapien. Cras ut lacus augue...
</p>
<h3>Otro párrafo sin posicionamiento</h3>
<p>
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Nunc vehicula leo id lectus faucibus eu rhoncus nisl
```

```
rutrum. Nunc orci sapien, commodo a commodo vitae,
sodales ut sapien. Cras ut lacus augue...
</p>
</body>
</html>
```

---

*Este cambio de posicionamiento se debe "leer" como 20px por debajo y desplazar el margen izquierdo 300px desde donde tendría que posicionarse naturalmente*

---

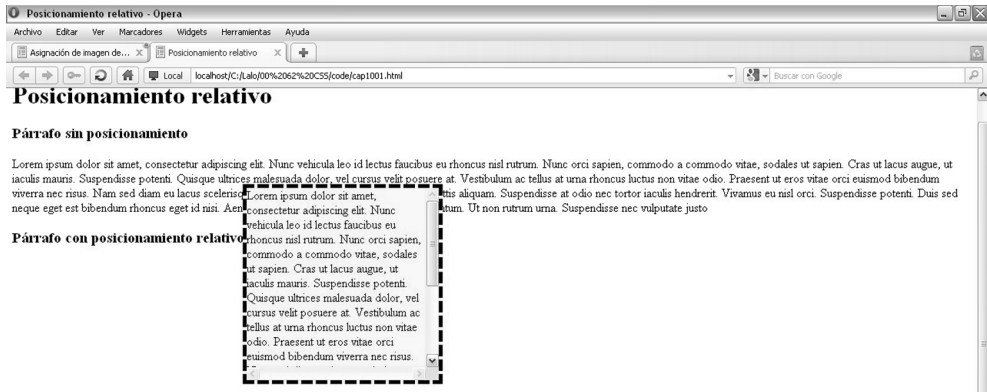


El posicionamiento relativo utilizado sin cuidado puede provocar superposiciones, por ejemplo, es lo que sucede con este estilo:

```
<style>
#id1 {width: 250px;
      height: 250px;
      border: 4pt black dashed;
      background-color: yellow;
      position: relative;
      overflow: scroll;
      top: -100px;
      left: 300px;}
</style>
...

```

El valor negativo de la propiedad top significa que al valor de la propiedad se le resta ese valor. Se obtiene este resultado:



En este ejemplo hemos utilizado las propiedades top y left, también podemos utilizar bottom o right para indicar el posicionamiento relativo.

---

*El uso de posicionamientos relativos en negativo puede producir solapamientos (obviamente, en algunos casos puede ser el efecto deseado).*

---

### Ejemplo de posicionamiento relativo

Veamos el efecto del posicionamiento relativo respecto al estático. El primer encabezamiento es static y el segundo es relativo (es decir, relativo al static). Este código corresponde al archivo cap1010.html.

```
<html>
<head>

<style type="text/css">
h2.relATIVO
{
position:relative;
/* mover el margen izquierdo 50px */
left:50px;
}
</style>

</head>
<body>
<h2>Este encabezado se posiciona de modo static (flujo
normal)</h2>
```

**Vistazo parcial del libro**  
**Salto de páginas...**

## 12: HTML dinámico y JavaScript

No podemos profundizar demasiado en el lenguaje JavaScript ya que es un tema que requiere más espacio del que se dispone en este libro, pero al menos incluiremos una introducción para que el lector pueda comenzar a comprender el código.

JavaScript es uno de los lenguajes que se denomina como lenguaje de script y que tiene como ventaja la facilidad en como se combina con el lenguaje HTML para permitir que una página web eleve su nivel de interactividad.

JavaScript por sí solo sería incapaz de interactuar con el documento html si no fuese por el apoyo del modelo de objetos de documento (DOM). El modelo DOM pone a disposición del lenguaje de script todos los objetos del documento para que puedan ser gestionados desde código de programa lo que hace que las páginas tengan comportamiento dinámico.

### **Tareas propias del lenguaje script**

Si nos tomamos el trabajo de ver el código de cualquier página web podremos comprobar que tenemos muchas probabilidades de encontrarnos con líneas de algún script escrito en JavaScript intercaladas con el típico código HTML y también con lenguajes que se ejecutan en el servidor, como PHP o ASP.NET.

JavaScript se utiliza para diversas tareas auxiliares, por ejemplo, para realizar una validación primaria de la entrada de datos o para responder a ciertas acciones del ratón o del teclado. JavaScript se suele utilizar principalmente para resolver la gestión de la publicidad; la visualización de banners o las animaciones Flash son ejemplos de estos usos. También se utiliza JavaScript para añadir los enlaces patrocinados de un sitio web.

La facilidad de JavaScript para responder a las acciones de los usuarios sin necesidad de que el proceso se realice en el servidor hace que sea la principal opción elegida por el desarrollador cuando se trata de diseñar el comportamiento del menú de navegación de un sitio web. JavaScript, en colaboración con Ajax, permite dar un paso más en la optimización de un sitio.

---

*JavaScript es uno de los principales mecanismos para lograr que una aplicación web tenga una velocidad de respuesta que se parezca a una aplicación local.*

---

## **CSS y HTML dinámico**

Hemos estudiado que CSS resulta ser una herramienta de gran ayuda para crear documentos correctamente formateados y presentados. Las propiedades de CSS se pueden modificar dinámicamente lo que permite mejorar más aún la estética de los documentos. Desde código script podemos actuar sobre el estilo CSS a través del modelo DOM que nos permite acceder a las partes de un documento desde un programa, es decir, desde fuera del código HTML.

Aunque al principio, en CSS1, no todos los navegadores compartían exactamente el mismo modelo DOM esta situación se mejoró en CSS2. De todas maneras, Mozilla e Internet Explorer siguen sin compartir un modelo totalmente común.

A las propiedades CSS se accede generalmente de la misma manera, utilizando su nombre de propiedad, por ejemplo, color. Sin embargo, cuando en el nombre de la propiedad CSS aparece un guión, para usar esa propiedad en un script se le debe eliminar ese guión y cambiar a mayúsculas la letra que sigue al guión, por lo tanto, font-family, dentro de un script se convierte en fontFamily.

Por lo tanto, para acceder a la propiedad font-family de un elemento con un identificador denominado id1 utilizaríamos un código como el siguiente:

```
id1.style.fontFamily = "Arial";
```

Y ¿cómo se inserta este código en una página HTML?

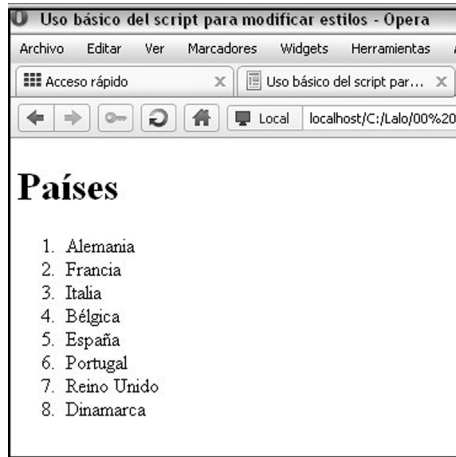
Esta instrucción JavaScript debe formar parte de alguna función del script de la página, por ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<title>Uso básico del script para modificar estilos
</title>

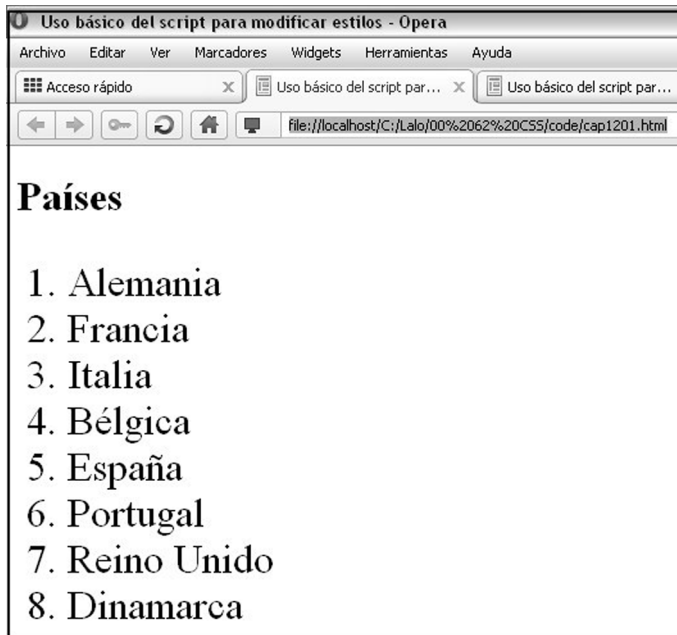
<script>
function CambiarFuente(obj) {
    obj.style.fontSize = "xx-large";
}
</script>

</head>
<h1>Países</h1>
<body>
<div onClick="CambiarFuente(this);">
<ol >
<li>Alemania</li>
<li>Francia</li>
<li>Italia</li>
<li>Bélgica</li>
<li>España</li>
<li>Portugal</li>
<li>Reino Unido</li>
<li>Dinamarca</li>
</ol>
</div>
</body>
</html>
```





Al hacer clic sobre el texto veremos que automáticamente se produce el cambio de tamaño de la fuente.



Al analizar el código se puede observar que se utiliza el evento `onClick()` para llamar a una función de JavaScript que está definida dentro de un bloque `<SCRIPT></SCRIPT>`.

La instrucción siguiente, dentro de la función CambiarFuente(), cambia el valor de la propiedad font-size de CSS:

```
obj.style.fontSize = "xx-large";
```

Si se observa la etiqueta <DIV> se verá que no existe una definición del estilo, y mucho menos una definición del tamaño de la fuente.

¿Qué hace entonces la instrucción de asignación a la propiedad font-size cuando la propiedad no existe?

La respuesta es que si la propiedad está definida se la reemplaza por el nuevo valor, pero si no existe la asigna directamente. Por lo tanto, desde JavaScript podemos trabajar sobre cualquiera de las propiedades CSS, estén o no definidas en el estilo.

---

*Pero antes de seguir con el código script que nos permite acceder a una propiedad CSS estudiaremos los conceptos de JavaScript.*

---

## JavaScript, un lenguaje cliente

La programación de JavaScript se basa en la edición de scripts que se integran al código HTML de la página y que se pueden visualizar en cualquier navegador mediante la opción Ver código fuente. El nombre de la opción varía entre los distintos navegadores, pero la funcionalidad final es visualizar el código fuente. Es decir, el usuario puede visualizar el código JavaScript, este detalle siempre lo debemos tener presente cuando pensemos en el aspecto de seguridad; dentro del código JavaScript no deberíamos incluir datos que podrían ser aprovechados maliciosamente por hackers (por ejemplo, la inclusión del nombre de la cadena de conexión a una base de datos o contraseñas).

JavaScript es un lenguaje interpretado (que es lo opuesto a un lenguaje compilado), por lo que el código se analiza y se ejecuta en el navegador del cliente. No existe una etapa de compilación del código como sucede en otros lenguajes de programación, por ejemplo, C++.

Una de las características básicas de JavaScript es que se ejecuta en el navegador del cliente; en este caso, lo opuesto sería que se ejecutase en el servidor, como lo hace, por ejemplo, PHP o ASP.NET. Esta característica tiene sus ventajas y sus limitaciones. La principal ventaja es el tiempo de respuesta de la aplicación: las acciones JavaScript se resuelven de modo local y no requieren el viaje hasta el servidor, por lo que el tiempo de espera

**Vistazo parcial del libro**  
**Salto de páginas...**

# Índice

## Símbolos

!important, notación 32  
#rrggbb 67  
:active 298  
:active, pseudo clase 51  
:after 225  
:before 225  
:first-child 53  
:first-letter 225  
:first-line 225  
:focus, pseudo clase 52  
:hover 298  
:hover, pseudo clase 51  
:link 298  
:link, pseudo clase 51  
:visited 298  
:visited, pseudo clase 51  
<noscript> 264  
<script> 261  
<STYLE> 36  
<uri> 319  
@import 304  
@media 303  
@page 305

## A

absolute, propiedad de estilo 180  
ActiveBorder 319  
ActiveCaption 319  
Ajax 240  
alfa, canal 69  
alineación horizontal del texto 101  
alineación vertical del texto 103  
alineación y posicionamiento 132  
almacenamiento de caché 263  
altura de la línea 91  
altura de las letras 82  
all 28, 302  
ámbito de las variables 265  
ámbito del lenguaje 240  
ampliación de texto 289  
ancestros 46  
ancho de banda limitado 28  
ancho del borde 153

ángulo de inclinación de un tipo 82  
animaciones Flash 236  
aplicación de estilo en selectores 40  
aplicación JavaScript 249  
AppWorkspace 319  
archivo externo js 261  
archivos externos 263  
ascendente 82  
asignar estilo 60  
ASP.NET 239  
aspectos básicos de CSS 21  
atributo class 26, 42  
atributo href 29  
atributo media 28, 303  
atributo type 28  
atributos en tablas HTML 123  
aural 28, 302  
auto 205

## B

Background 319  
background 164, 177  
background, propiedad de estilo 163, 164  
background-attachment 173, 177  
background-attachment, propiedad de estilo 163  
background-color 166, 177  
background-image, propiedad de estilo 163  
background-position 174, 177  
background-position, propiedad de estilo 164  
background-repeat 172, 177  
background-repeat, propiedad de estilo 164  
background-image 177  
baseline 103  
blink 119  
bloque de instrucciones 264  
booleano, tipo 267  
border 125, 157  
border-bottom 157  
border-bottom-color 156  
border-bottom-style 154  
border-bottom-width 153  
border-color 156

- border-collapse 127, 157
- border-left 157
- border-left-color 156
- border-left-style 154
- border-right 157
- border-right-color 156
- border-right-style 154
- border-right-width 153
- border-spacing 126, 157
- border-style 154
- border-top 157
- border-top-color 156
- border-top-style 154
- border-top-width 153
- border-width 153
- bordes 152
- bordes de una tabla 125
- bordes, padding y márgenes 159
- bottom 104
- bottom, propiedad de estilo 180
- Braille 302
- braille 28
- break, sentencia 270
- bucles de ejecución en JavaScript 271
- ButtonFace 319
- ButtonHighlight 319
- ButtonShadow 319
- ButtonText 319

## C

- caché 263
- caja de formato 147
- cálculo de la especificidad 78
- caller, propiedad 276
- cambio del tamaño 196
- campo de acción de márgenes, bordes y rellenos 149
- canal alfa 69
- capitalize 117
- caption, título en tablas 132
- CaptionText 319
- cascada 73
- case() 270
- celda vacía 127
- class 26
- class, atributo 42
- class para aplicar estilos 40
- cláusula case() 270
- clear, propiedad 111

- close-quote 229
- cm: Centímetros 64
- codificación del color 156
- codificación JavaScript 244
- color de fondo 164, 166
- color de frente 164
- color del borde 156
- color, propiedad 164
- color, propiedad de estilo 163
- color y fondo 163
- colores de la interfaz 319
- collapse 215
- combinación de condiciones 269
- comentarios 59, 256
- comentarios en Javascript 256
- comparación de valores 268
- compatibilidad puesta en duda 242
- complementos en los navegadores 244
- conceptos básicos de Javascript 254
- conurrencia de estilos 62
- conexión HTML/JavaScript mediante eventos 253
- conjunto de reglas 73
- contenido que excede el contenedor 205
- contenido y presentación 23
- content, propiedad 223
- contornos 161
- contracción del borde 127
- control de audición 70
- control de bucles 271
- control del tamaño del elemento 201
- cookies 246
- corrección de errores 254
- crosshair 318
- CSS desde JavaScript 289
- currentStyle 285
- cursive 83
- cursor 318

## D

- dashed 154
- decisión de la precedencia 32
- declaración de estilo 39
- decoración de texto 119
- decoración y transformación de texto 121
- definición CSS 25
- definición de una clase 42
- definición en línea 29
- definir los estilos 27

descendente 82  
desigualdad 268  
desplazamiento de imágenes 173  
determinación de la prioridad 78  
diferencia entre mayúsculas y minúsculas 266  
dimensionar la tabla 131  
display, propiedad CSS 216  
do en JavaScript 271  
do/while, sentencia 274  
documento para impresión 305  
Dotted 154  
double 154

## E

edición CSS en Firefox 326  
edición de código CSS 324  
editores de texto 244  
efecto de la herencia 74  
ejecución de una aplicación JavaScript 249  
elementos flotantes 109  
em, unidad de medida 65  
embossed 28, 302  
empty-cells 128  
entorno de pruebas 248  
entornos diferentes 243  
equivalencia de atributos CSS y tablas HTML 125  
espaciado de bordes 157  
espaciado de letras 113  
espaciado de letras y de palabras 113  
espaciado de palabras 113  
espaciado horizontal 83  
espaciado vertical 83  
espacios en blanco 115  
especificación del tipo de medio 303  
especificidad 78  
estilo con identificador 43  
estilo de la fuente 89  
estilo del borde 154  
estilos de usuario 75  
estilos definidos por el autor 75  
estilos del agente de usuario 75  
estructura de un documento HTML 46  
estructura del código 255  
evaluación aritmética 281  
ex, unidad de medida 65  
extensión para desarrollador Web 245

## F

fantasy 84  
fishing 241  
fixed, propiedad de estilo 180  
float 111, 196  
flotación de los elementos 196  
flujo normal del texto 109  
fondo con imágenes 175  
fondos de color 169  
font, propiedad 94  
font-family, propiedad de estilo 82  
font-size, propiedad de estilo 83  
font-style, propiedad de estilo 83  
for en JavaScript 271  
for, sentencia 271  
foreground 164  
fuente 81  
fuentes de texto 92  
fuentes del sistema 95  
función llamante 276  
función rgb() 165  
funciones 275  
funciones en JavaScript 275

## G

getComputedStyle 285  
glifo 82  
GrayText 319  
groove 154

## H

handheld 28, 302  
help 318  
herencia 73  
herencia del estilo 50  
hermanos 48  
herramientas de desarrollo de Internet Explorer 247  
hidden 205, 215  
hide 128  
Highlight 320  
HighlightText 320  
hoja de estilo en cascada 30  
hoja de estilo externa (<LINK>) 28  
hoja de estilo interna 27  
hojas de estilo 35  
href, atributo 29  
HSL 68, 69  
HSLA 69

HTML y estilos 23

## I

icon 321  
identificador 43  
if en JavaScript 270  
if, sentencia 268  
if/else, sentencia 269  
igualdad 268  
imágenes como fondo 170, 175  
imágenes en viñetas 144  
import, sentencia 62  
importación de una hoja de estilo 304  
impresión en papel 301  
impresoras Braille 302  
in: Pulgadas 64  
InactiveBorder 320  
InactiveCaption 320  
InactiveCaptionText 320  
inclusión de código JavaScript 260  
incremento del contador 231  
incremento/decremento por unidades 273  
indefinido 267  
InfoBackground 320  
InfoText 320  
inline, estilo 62  
inline-block 216  
inline-table 217  
inset 154  
interrelación HTML y estilos 23

## J

Java 240  
JavaScript 239  
JavaScript no es Java 240  
JavaScript, un lenguaje cliente 239  
js archivo externo 261

## L

landscape 306  
left, propiedad de estilo 180  
length 104  
lenguaje en el servidor 240  
lenguaje interpretado 249  
letter-spacing 83, 113  
limitación a un atributo 45  
límites de tamaños 201  
line-height 83  
line-through 119

línea base 82  
LINK 28, 61  
list-item 138  
list-style-image 138  
list-style-position 138  
list-style-type 138  
listas 135  
listas desordenadas 136  
listas numeradas 233  
listas ordenadas 136  
lo que JavaScript no hace 240  
longitud de los nombres 259  
lowercase 117

## LL

llamada recursiva 276

## M

malsoftware 242  
mantenimiento evolutivo 254  
marcas de corte 306  
márgenes 158, 159  
margin 159, 306  
margin-bottom 158  
margin-left 159  
margin-right 159  
margin-top 158  
marks 307  
max-height 202  
max-width 202  
mayor o igual 269  
mayor que 268  
mayúsculas 266  
mayúsculas y minúsculas 117  
media, atributo 28  
media, en etiqueta STYLE 38  
menor o igual 269  
menor que 268  
Menu 320  
menu 321  
menú horizontal 137  
MenuText 320  
message-box 321  
middle 104  
min-height 201  
min-width 201  
minúsculas 266  
mm: Milímetros 64  
modelo DOM 235

modo de definir estilo 27  
modos de definir los estilos 27  
monospace 83  
motores de búsqueda 282  
move 318  
muestra la celda 128

## N

NaN, not a number 267  
nomenclatura de variables y funciones 257  
nomenclatura de las funciones Javascript 258  
nomenclatura de las variables Javascript 258  
notación !important 32  
notación !important: 32  
nowrap 115  
numeración automática de los elementos 230  
numérico, tipo 267

## O

oculta la celda 128  
ocultar y mostrar texto 291  
onMouseOut 292  
onMouseOver 292  
opacidad 69  
open-quote 229  
operador ternario en JavaScript 271  
operador typeof 267  
operadores de comparación de valores 268  
operadores lógicos Y u O 269  
orden de las definiciones 52  
orphan 308  
outline 161  
outline-color 161  
outline-style 161  
outline-width 161  
outset 154  
overflow 205  
overline 119

## P

padding 126, 150, 159  
padding-bottom 150  
padding-left 150  
padding-right 150  
padding-top 150  
padres e hijos 48  
page-break-after 307  
page-break-before 307

page-break-inside 307  
palabra clave de colores 156, 164  
palabras reservadas de Javascript 259  
para qué se necesitan los estilos 22  
parámetros de la función 276  
parpadeo 119  
pc: Picas 65  
PDA 302  
percentage 104  
PHP 239  
pointer 318  
porcentaje de color 156, 165  
posicionamiento 180  
posicionamiento absoluto 187  
posicionamiento de la imagen 174  
posicionamiento de los elementos 181  
posicionamiento estático 182  
posicionamiento fijo 189, 190  
posicionamiento mediante porcentajes 191  
posicionamiento relativo 184, 186  
position, propiedad de estilo 176, 180  
pre 115  
pre-line 115  
pre-wrap 115  
precedencia 32  
previsualizaciones en pantalla 28  
primera línea del párrafo 106  
print 28, 302  
prioridad en etiquetas LINK 77  
progress 318  
projection 28, 302  
propiedad float 196  
propiedad margin 306  
propiedad marks 307  
propiedad size 306  
propiedades asociadas al color y al fondo 93, 94, 97  
propiedades border 156  
propiedades de fuente del texto 82  
proyección de transparencias 302  
pseudo clase :first-child 53  
pseudo clase :lang 53  
pseudo clases 51, 298  
pseudo clases para la etiqueta <A> 51  
pseudo elementos 53, 223  
pt: Puntos 65  
px: Pixel 65



**R**

redimensionamiento del texto 88  
referencia a una clase 40  
regla @import 304  
regla @media 303  
regla @page 305  
reinicialización del contador 231  
relative, propiedad de estilo 180  
relleno 150, 151  
repetición de la imagen  
172, 173, 174, 175, 178  
rgb(x,x,x) 68  
RGBA 68  
ridge 154  
right, propiedad de estilo 180  
robo de información personal 242  
run-in 217

**S**

salida braille 28  
salida impresa 301  
sangría de la primera línea 106  
sangría del código 255  
sangría del texto 106  
sans-serif 83  
scoped, en etiqueta STYLE 38  
screen 28, 302  
script JavaScript 260  
script, modo indirecto 261  
scroll 205  
Scrollbar 320  
selección por jerarquía 49  
selector 35  
selector limitado 44  
selector universal 44  
sentencias JavaScript 268  
separación de letras 113  
separación entre contenido y presentación 23  
serif 83  
show 128  
small-caption 321  
software indeseado 241  
solid 154  
src, atributo en script 261  
static, propiedad de estilo 180  
status-bar 321  
string, tipo 267  
style, atributo en etiquetas 62  
sub 103

subrayado 119  
super 104  
superposición de elementos 208, 214  
switch en JavaScript 270  
switch, sentencia 270

**T**

tabla con la propiedad display 221  
table-cell 217  
table-column 217  
table-footer-group 217  
table-header-group 217  
table-layout 131  
table-row 217  
table-row-group 217  
tachado 119  
tamaño de la fuente 86  
tamaño de la página 305  
tareas de mantenimiento 254  
tareas del lenguaje script 235  
teletipo 302  
teletipos 28  
televisión 28  
text 318  
text-align 101  
text-bottom 104  
text-decoration 299  
text-decoration, propiedad de estilo 119  
text-indent 106  
text-indent, propiedad de estilo 106  
text-shadow 119  
text-top 104  
text-transform 117  
ThreeDDarkShadow 320  
ThreeDFace 320  
ThreeDHighlight 320  
ThreeDLightShadow 320  
ThreeDShadow 320  
tipo de las variables 266  
tipos de datos en Javascript 267  
tipos de fuentes 83  
title, en etiqueta STYLE 38  
top 104  
top, propiedad de estilo 180  
transferencia de servidor local a remoto 247  
troyanos 241  
tty 28, 302  
tv 28, 302  
type, atributo 28

type, en etiqueta STYLE 38  
typeof 267

## U

underline 119  
unidades de medida 64, 86  
unidades por porcentajes 65  
uppercase 117  
uso del caché 263

## V

validación de código CSS 324  
valor decimal de color 156, 165  
valor hexadecimal de color 156, 164  
valores de color 67  
valores de texto 70  
variables 265  
ventaja del uso de caché 261  
Ver código fuente 239  
vertical-align 103  
virus 241  
visibility, propiedad de estilo 180  
visible 205, 215  
visualización de banners 236

## W

wait 319  
while en JavaScript 271  
while, sentencia 274  
white-space 115  
widow 308  
Window 320  
window: getComputedStyle 285  
WindowsFrame 320  
WindowText 320  
word-spacing 113  
worms 241

## X

x-resize 319

## Z

z-index 208  
z-index, propiedad de estilo 180