

DAM

# Linguaxes de Marcas e Sistemas de Información

UD2 **CSS**

PROFESORA | CRISTINA PUGA FDEZ

UNIDADE | UD2

TRIMESTRE | 1

MÓDULO | LMSXI

## Tabla de contenido

2.1	Introducción.....	3
2.2	Las hojas de estilo .....	4
2.2.1	Sintaxis.....	4
2.2.2	Unidades de medida.....	5
2.2.3	Colores.....	6
2.2.4	Comentarios .....	6
2.2.5	Definición y uso de clases.....	7
2.3	Incorporación de estilos en un documento HTML o XHTML.....	8
2.3.1	Definir CSS en un archivo externo enlazado .....	9
2.3.2	Definir CSS en un archivo externo importado.....	12
2.3.3	Definir CSS en un documento HTML .....	13
2.3.4	Incluir CSS en los elementos HTML .....	14
2.4	Tipos de selectores .....	15
2.4.1	De elemento .....	15
2.4.2	De identificador .....	16
2.4.3	De clase.....	16
2.4.4	Pseudoclases .....	17
2.4.5	Selector universal .....	19
2.5	Funcionamiento de CSS .....	19
2.5.1	Modelo de cajas .....	19
2.5.2	Herencia .....	22
2.5.3	Posicionamiento .....	23
2.6	Visualización .....	25
2.7	Atributos básicos .....	26
2.7.1	Atributos de tamaño para todos los elementos .....	26
2.7.2	Atributos de fuente .....	26
2.7.3	Atributos de texto .....	28
2.7.4	Atributos de fondo .....	29
2.7.5	Atributos de caja .....	30
2.7.6	Atributos de tabla.....	33
2.7.7	Enlaces.....	34
2.8	CSS 3.....	36
2.9	Herramientas de desarrollo .....	37

## 2. CSS

---

### 2.1 Introducción

**CSS** (Cascade Style Sheets) es un lenguaje estándar definido por el W3C para definir la presentación de los documentos Web (la forma en que se muestran en pantalla los distintos elementos de HTML), de forma que permite mantener por separado los contenido (HTML) y su presentación (CSS).

Esto aporta varias ventajas:

- Mayor control en el diseño de las páginas (extiende la capacidad de HTML).
- Reutilización y gestión centralizada del formato: se puede cambiar el estilo de todo un sitio web modificando un único archivo.
- Documentos más pequeños y manejables: los documentos HTML serán mucho más sencillos, ya que se elimina toda la definición de formato en su definición simplificándolos considerablemente.
- Mayor accesibilidad: permite establecer distintas presentaciones en función del dispositivo empleado para visualizar la página.
- Está soportado por prácticamente todos los navegadores

Tras la definición de CSS3 y HTML5, prácticamente todos atributos de presentación de los elementos HTML se declararon obsoletos por el W3C, ya que se recomienda delegar en CSS toda la gestión del aspecto de la web.

CSS fue propuesto por vez primera por Håkon Wium Lie en octubre de 1994, cuando trabajaba con Tim BernersLee (el creador de HTML) en el CERN. En aquella época, también se propusieron otros lenguajes de estilos, pero en 1996 el W3C decidió publicar la primera versión del estándar CSS (CSS1).

Posteriormente se publicaron otras 3 versiones del estándar:

- CSS 2, publicada en 1998
- CSS 2.1, publicada en 2011
- CSS 3, publicada en 2015 y versión utilizada en la actualidad.

## 2.2 Las hojas de estilo

### 2.2.1 Sintaxis

El lenguaje CSS permite definir hojas de estilos. Una hoja de estilo es un conjunto de reglas definidas en lenguaje CSS que describen cómo deberían mostrarse por pantalla los distintos elementos HTML. Las reglas constan de los siguientes componentes:

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

Cada uno de los estilos que componen una hoja de estilo CSS, se denomina regla. Cada regla está formada por:

- Selector: indica el elemento o elementos HTML a los que se aplica la regla CSS
- Llave de apertura, {
- Declaración: especifica los estilos que se aplican a los elementos, separando por ; cada par "propiedad: valor".
  - Propiedad: permite modificar la apariencia de un atributo de elemento.
  - Valor: indica el nuevo valor del atributo modificado en el elemento.
- Llave de cierre, }.

#### Ejemplo:

```
p {color: azul;}
```

En este caso el selector es "p", la declaración es: "color: azul" y dentro de esto podemos diferenciar entre la propiedad "color" y el valor "azul".

Un archivo CSS puede contener infinitas reglas CSS, cada regla puede contener múltiples selectores y cada declaración puede estar compuesta de diferentes declaraciones.

CSS ignora los espacios en blanco y los saltos de línea, por lo que las propiedades pueden indicarse todas seguidas o con sangrado para su mejor comprensión:

#### Ejemplo:

```
h1 {color: #EEEEEE;background-color: green;}  
/* muestra las cabeceras de nivel 1 en gris y con fondo verde  
*/ p {  
    font-size: 12px;
```

```
font-family: Verdana, sans-serif;
}
/* muestra los párrafos con tipo de letra Verdana o sans-serif
de tamaño 12 */
```

## 2.2.2 Unidades de medida

CSS permite indicar unidades de medida absolutas o relativas.

- Las unidades **absolutas** permitidas son:
  - Milímetros (mm).
  - Centímetros (cm): cada centímetro son 10 mm.
  - Pulgadas (in): cada pulgada equivale a 2,54 cm.
  - Puntos (pt): cada punto son 1/72 in.
  - Picas (pc): cada pica son 12 pt.
- Las unidades **relativas** pueden ser:
  - Element (em): expresa el tamaño relativo al tamaño de la fuente utilizada (si se emplea una tipografía de 12pt, 1em equivaldría a 12pt y 0.5em a 6pt)
  - X-height (ex): expresa el tamaño relativo al de la letra "x" del tipo y tamaño de letra del elemento
  - Píxel (px): expresa el tamaño relativo a la resolución de la pantalla del dispositivo en que se visualiza la página

### Ejemplo:

```
body {
  margin: 0.5in;
}
p {
  font-size: 14pt;
  margin: 1em;
}
```

En general, se recomienda el uso de unidades relativas siempre que sea posible, ya que mejora la accesibilidad de la página y permiten que los documentos se adapten fácilmente a cualquier medio.

Normalmente se utilizan:

- píxel y porcentajes para definir el layout del elemento (el ancho de las columnas y elementos de la página)
- em y porcentajes para el tamaño de letra de los textos.

### 2.2.3 Colores

Los colores en CSS se pueden indicar de diversas formas. Las más usadas son:

- Las palabras clave: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, orange, purple, red, silver, teal, white y yellow
- Los códigos RGB hexadecimales que correspondan a cada color, precedidos por el símbolo #

<b>maroon</b> #800000	<b>red</b> #ff0000	<b>orange</b> #ffa500	<b>yellow</b> #ffff00	<b>olive</b> #808000
<b>purple</b> #800080	<b>fuchsia</b> #ff00ff	<b>white</b> #ffffff	<b>lime</b> #00ff00	<b>green</b> #008000
<b>navy</b> #000080	<b>blue</b> #0000ff	<b>aqua</b> #00ffff	<b>teal</b> #008080	
<b>black</b> #000000	<b>silver</b> #c0c0c0	<b>gray</b> #808080		

```
p {
  color: blue;
}
h1 {
  color: #AD447C;
}
```

### 2.2.4 Comentarios

Los comentarios en CSS se incluyen entre los caracteres /\* y \*/.

```
/* Este es un comentario en CSS */
/* Este es un
```

```
comentario CSS de varias líneas */
```

## 2.2.5 Definición y uso de clases

Cuando las reglas de estilo se asocian con un documento HTML utilizando un archivo externo o se incluyen en el contenido de la etiqueta STYLE en encabezados de documentos, los estilos se pueden definir y asociar con ciertos elementos del documento.

La siguiente sintaxis se utiliza para definir una clase:

```
.class_azul {cor: azul}
```

Para asociar un elemento HTML con una clase, deberá usar el atributo CLASS cuando use ese elemento en el documento HTML de la siguiente forma:

```
<h3 class = "clase_azul"> El encabezado del tercer nivel ahora es azul </h3>
```

Para restringir la clase a un elemento en particular, simplemente se coloca el elemento delante del punto al definir la regla. Por ejemplo, para restringir el uso de la clase en párrafos tendremos:

```
p.class_azul {cor: azul}
```

Un ejemplo de un documento XHTML que usa este método para incluir formatos es:

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

  <head>
    <title>Ejemplo del uso de clases en hojas de
estilo</title>
    <style type="text/css"> .clase_azul{color:blue}
p.clase_roja{color:#ff0000; font-style:italic; font-
weight:bolder; font-family:courier;} </style>
  </head>

  <body>
```

```

    <h3 class="clase_azul">Ejemplo del uso de clases en
    hojas de estilo</h3>
    <p>Cualquier elemento sobre el que apliquemos la clase
    clase_azul tendrá el texto azul.</p>
    <p class="clase_azul"> Incluso el párrafo.</p>
    <p class="clase_roja">Sobre el párrafo podemos
    aplicar la clase clase_roja y el texto será rojo, en
    negrita cursiva y la familia del texto courier.</p>
    <h3 class="clase_roja"> Pero este texto no
    aparecerá formateado ya que la clase clase_roja
    sólo actúa sobre párrafos.</h3>
    </body>
</html>

```

El cual, al ser publicado en un navegador, tendremos:

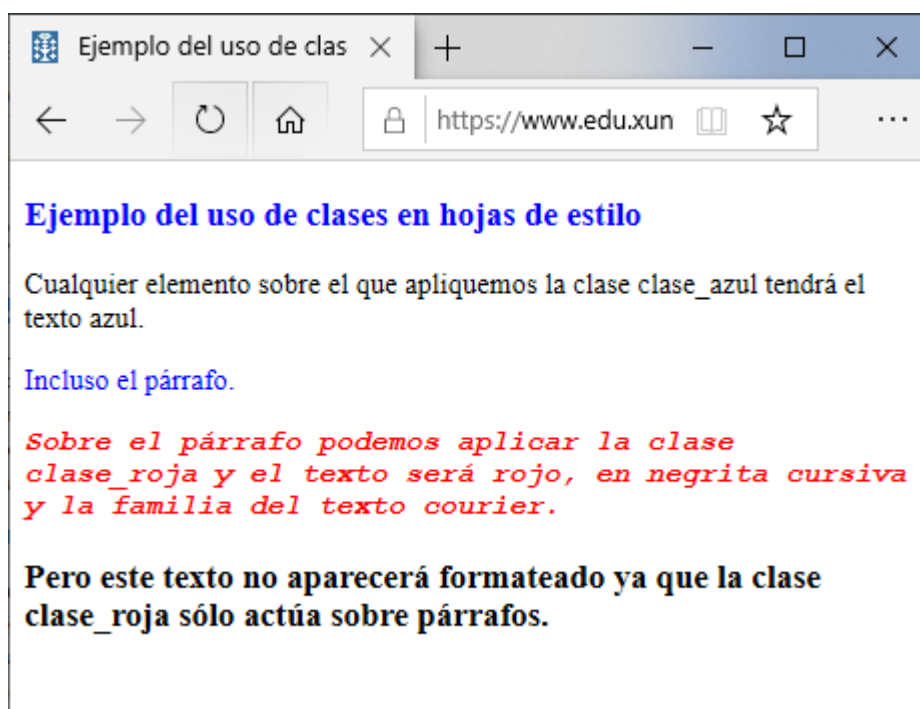


Figura 1 – Ejemplo clases

## 2.3 Incorporación de estilos en un documento HTML o XHTML

Existen tres opciones para emplear estilos CSS en un documento HTML o XHTML.

### 1. Definir CSS en un archivo externo.

En este caso, todos los estilos CSS se incluyen en uno o más archivos de texto sin formato, cuya extensión es .css, a la que se vinculan las páginas HTML mediante el elemento <link> en el encabezado del archivo HTML.



Dado que una página web puede tener múltiples archivos CSS asociados, es aconsejable agrupar los archivos CSS en un directorio.

El navegador descarga los archivos CSS externos, además de la página web asociada a ellos, y aplica los estilos al contenido de la página antes de mostrar su contenido.

Así es como se incluye CSS en las páginas HTML más utilizadas. La principal ventaja es que se puede incluir el mismo archivo CSS en una multitud de páginas HTML, de modo que se garantiza la aplicación homogénea de los mismos estilos en todas las páginas que componen un sitio web.

Además, el mantenimiento del sitio web se simplifica al máximo, ya que cambiar a un solo archivo CSS cambia instantáneamente los estilos de todas las páginas HTML asociadas.

Se puede hacer de dos maneras diferentes:

- Mediante enlaces.
- Importando el fichero CSS.

## **2. Incluir CSS en el documento HTML.**

Este método se utiliza cuando se definen pocos estilos o cuando desea incluir estilos específicos en una página HTML particular que complementen los estilos generales de todas las páginas del sitio web.

Tiene la desventaja de que para modificar los estilos definidos, es necesario modificar todas las páginas que incluyen el estilo que se va a cambiar.

## **3. Incluir CSS en los elementos HTML.**

El último método para incluir estilos CSS en documentos HTML es el peor y el menos utilizado, ya que, para cambiar un formato, debe cambiar todos los elementos asociados con él.

Solo se usa en ciertas situaciones en las que se debe incluir un estilo muy específico para un solo elemento específico.

### **2.3.1 Definir CSS en un archivo externo enlazado**

Para realizar una página web usando un archivo CSS externo, se deben seguir los siguientes pasos:

- Crear un archivo de texto plano con las definiciones de los formatos.
- Guardar dicho archivo de texto con la extensión .css

- Enlazar el archivo CSS externo mediante la etiqueta <link> en la cabecera de la página web.

El elemento <link> puede tener definidos cuatro atributos cuando se enlaza un archivo CSS:

- rel, indica el tipo de relación que tiene el archivo enlazado y la página HTML. Para los archivos CSS, siempre se utiliza el valor stylesheet
- type, indica el tipo de recurso enlazado. Para los archivos CSS su valor es siempre text/css
- href, indica la URL del archivo CSS que contiene los estilos. Puede ser relativa o absoluta y puede referenciar a un recurso interno o externo al sitio web.
- media, indica el medio en el que se van a aplicar los estilos del archivo CSS.

La definición por medio de CSS vinculado es en general la más recomendable, ya que tiene la ventaja de que una misma hoja de estilos se puede emplear en múltiples documentos HTML, y modificando un solo archivo CSS permite modificar instantáneamente todas las páginas que lo enlazan.

Un ejemplo del uso de archivos externos CSS vinculados, para la construcción de páginas web sería:

- El fichero Ejemplo2.html contendría:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Ejemplo de estilos CSS en un archivo externo</title>
<link rel="stylesheet" type="text/css" href="css/estilos.css"
  media="screen" />
</head>

<body>
<h1>Ejemplo 2</h1>
<p>Un párrafo de texto.</p>
</body>
</html>
```

- El fichero “estilos.css” que guardaríamos en la carpeta /css, tendría el siguiente contenido:

```
p {
  color: red;
  font-family: Verdana;
}
```

```
h1 {  
  color: #00ff00;  
}
```

- El resultado de abrir en un navegador el fichero Ejemplo2.html sería:



Figura 2- Ejemplo 2

La etiqueta *link* ofrece además la posibilidad de definir distintos estilos en función del medio de presentación empleado para visualizar la página, por medio del atributo “media”.

Este atributo puede tomar los siguientes valores:

- all: Todos los medios definidos.
- braille: Dispositivos táctiles que empleen el sistema Braille
- embosed: Impresoras braille
- handheld: Dispositivos de mano: móviles, PDAs, etc.
- print: Impresoras y navegadores en modo "Vista Previa para Imprimir"
- projection: Proyector y dispositivos para presentación
- screen: Pantallas de ordenador
- speech: Sintetizadores para navegadores de voz empleados por personas discapacitadas
- tty: Dispositivos textuales limitados como teletipos y terminales de texto
- tv: Televisores y dispositivos con resolución baja.

### 2.3.2 Definir CSS en un archivo externo importado

Utilizando el elemento **<style>** en lugar de **<link>**, si se utiliza una regla de tipo **@import** seguida de una cadena de texto encerrada entre comillas simples o dobles que se corresponde con la URL del archivo CSS, y contiene dicha cadena entre paréntesis, se puede utilizar un fichero .css externo.

Las siguientes reglas **@import** son equivalentes para un fichero **formatos.css** que está en el directorio **css**:

- **@import 'css/formatos.css';**
- **@import "css/ formatos.css"**
- **@import url('css/ formatos.css');**
- **@import url("css/ formatos.css");**

El ejemplo anterior quedaría:

- El fichero **Ejemplo3.html** contendría:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Ejemplo de estilos CSS en un archivo externo usando
  style</title>
<style type="text/css">@import 'css/estilos.css'</style>
</head>

<body>
<h1>Ejemplo 3</h1>
<p>Un párrafo de texto.</p>
</body>
</html>
```

- El fichero **"estilos.css"** que guardaríamos en la carpeta **/css**, sería el mismo que en el ejemplo anterior:

```
p {
  color: red;
  font-family: Verdana;
}

h1 {
  color: #00ff00;
}
```

- El resultado de abrir en un navegador el fichero Ejemplo2.html sería:



Figura 3- Ejemplo 3

La etiqueta *link* nos permite importar hojas de estilos que están en nuestro propio servidor, que es lo más habitual, pero existe también la posibilidad de importar hojas de estilos accesibles a través de internet, utilizando “@import”:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>CSS vinculado</title>
    @import url(http://estilos/miestilo.css);
  </head>
  <body>
    <p>Texto de ejemplo</p>
  </body>
</html>
```

### 2.3.3 Definir CSS en un documento HTML

Consiste en incluir la hoja de estilos CSS en el propio documento HTML, empleando la etiqueta `<style>` que se puede incorporar en la cabecera del documento.

Esta opción es adecuada cuando se van a emplear un número reducido de estilos que completen los estilos por defecto de todas las páginas web del sitio.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>CSS incrustado en la cabecera del propio
    documento</title>
    <style>
      p {color: red; font-size: 12pt;}
    </style>
  </head>
  <body>
    <p>Texto de ejemplo</p>
  </body>
</html>
```

```
</style>
</head>
<body>
  <p>Texto de ejemplo</p>
</body>
</html>
```

Al guardar el ejemplo como Ejemplo4.html y abrirlo desde un navegador, tendríamos:

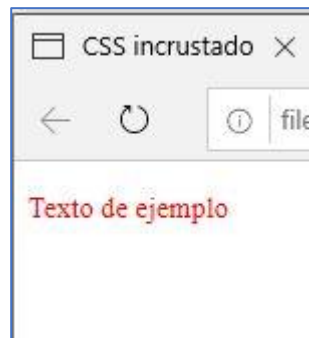


Figura 4- Ejemplo 4

### 2.3.4 Incluir CSS en los elementos HTML

El estilo de cada elemento se define en el atributo “style” del propio elemento. Esta opción se suele utilizar cuando queremos “sobreescribir” reglas definidas a más alto nivel en un estilo concreto (si no queremos que un <p> se muestre como el resto de <p> del documento, por ejemplo), pero no se recomienda ya que no es reutilizable y además no separa la presentación de cada elemento de su contenido (de hecho, el atributo style está deprecated desde la versión 1.1 de XHTML)

Un ejemplo de un documento XHTML en el que se utiliza este método para incluir formatos es:

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

  <head>
    <title>Ejemplo de estilos CSS en el propio
    fichero</title>
  </head>

  <body>
    <h3 style="color: green;">Ejemplo del uso de estilos
    CSS en el propio fichero</h3>
```

```
<p style="color: orange; font-family: Verdana;">En
esta ocasi&ocute;n el p&aacute;rrafo de texto aparece de
color naranja y el encabezado h3 aparece en verde, mientras
que en los ejemplos vistos hasta ahora el color de la letra
era negro.</p>
</body>
</html>
```

Al publicarlo en un navegador, tenemos:



Figura 5 – Ejemplo 4

## 2.4 Tipos de selectores

Existen varios tipos de selectores:

### 2.4.1 De elemento

Definen un estilo para todos los elementos del tipo que se indique: `p {font-size: 12pt;}`. Si queremos definir un estilo común para varios elementos, podemos indicarlos separados por comas. Podemos también anidar los elementos, de forma que solamente se aplique un estilo cuando se da una combinación concreta de elementos:

- aplicaría el estilo a todos los elementos `div` y a los `p`:

```
div,p {font-size: 12pt;}
```

- aplicaría el estilo a los elementos `p` contenidos en un `div`:

```
div p{font-size:12pt}
```

## 2.4.2 De identificador

Se representan precedidos por una almohadilla (#), y permiten definir un estilo para los elementos que tengan un identificador específico.

```
#nombre {font-size: 12pt;}
```

Aplicaría el estilo al elemento que tenga el id="nombre", es decir, a:

```
<p id="nombre">Texto del párrafo</p>
```

## 2.4.3 De clase

Se representan precedidos por un punto (.) y permiten definir un estilo para todos los elementos que tengan asignados una clase (class) concreta.

```
.definicion {font-size: 12pt;}
```

Aplicaría el estilo a todos los elementos que tengan class="definicion" :

```
<p class="definicion"> Texto del párrafo</p>
```

Para poder emplear clases con elementos que no estén contenidos en una etiqueta HTML existen dos etiquetas, <div> y <span> que permiten definir contenedores, a los que aplicar un estilo concreto.

La diferencia entre ambos es:

- <div> es una etiqueta de bloque (su contenido se muestra en una nueva línea y ocupa el 100% del ancho del elemento contenedor)
- <span> es una etiqueta de línea (su contenido se muestra a continuación del anterior y ocupa solamente lo que ocupe su contenido).

Estos dos elementos por sí mismos no hacen nada, pero permiten agrupar otros elementos para mostrarlos conjuntamente o asignarles un estilo común.

Un ejemplo:

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



```
<head>
  <title>Ejemplo span e div</title>
  <style>
    .destacado {color: red;}
    .normal {font-size: 15pt;}
    #azul {color: blue;}      </style>
</head>
<body>
  <h1>Cabecera de <span class="destacado">primer</span>
nivel</h1>
  <div class="normal">
    <p>Este es el <span class="destacado">primer</span>
párrafo.</p>      <p>Este es el <span
class="destacado">segundo</span> párrafo.</p>      </div>
    <p>Este es el <span class="destacado">tercer</span>
párrafo.</p>
    <p>Este es el <span id="azul">cuarto</span> párrafo.</p>
  </body>
</html>
```

Y si abrimos en un navegador el fichero del ejemplo que hemos guardado como EjemploClase.html, se mostraría:



Figura 6 – Ejemplo del uso de clases

## 2.4.4 Pseudoclases

Permiten aplicar diferentes estilos a un elemento en función de alguna característica (su estado,

la posición que ocupa, etc). Se representan indicando a continuación del elemento, el símbolo de dos puntos (:) y la característica a la que se aplicará (a:link – para enlaces no visitados, por ejemplo).

Se usan mucho para dar estilo a los enlaces (el orden de los elementos en este caso es importante: a:hover tiene que ir después de a:link y a:visited , y a:active debe ir después de a:hover).

- Aplicar el estilo a los enlaces no visitados.

```
a:link{color:green;}
```

- Aplicar el estilo a los enlaces visitados.

```
a:visited{color:green;}
```

- Aplicar el estilo al enlace cuando se posa el ratón sobre él. Se pueden usar para cualquier elemento, no solo para enlaces.

```
a:hover{color:green;}
```

- Aplicar el estilo al enlace cuando se pulsa.

```
a:active{color:green;}
```

- Aplicar el estilo al primer p que esté dentro de un div

```
div p:first-child{color:green;}
```

- Aplicar el estilo al último p que esté dentro de un div

```
div p:lastchild{color:green;}
```

## 2.4.5 Selector universal

Se representa con un asterisco (\*) y permite aplicar un estilo a todos los elementos de una página:

```
* {font-size: 12pt;}
```

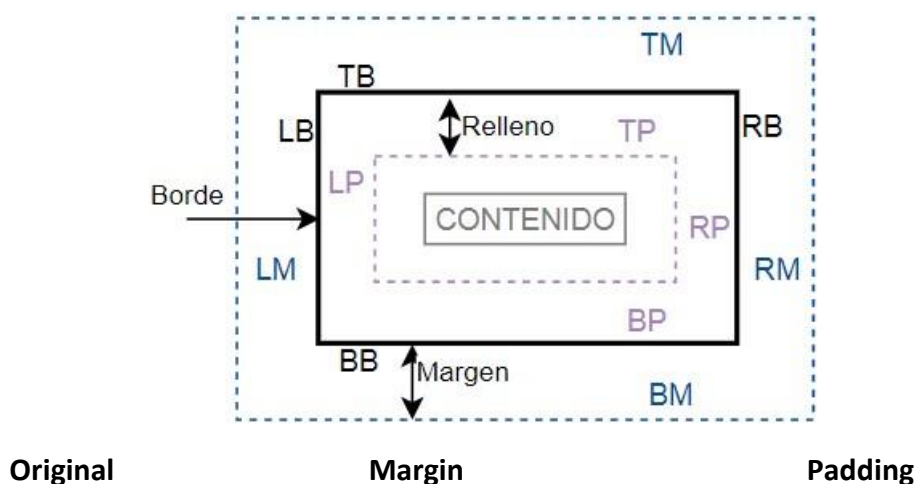
## 2.5 Funcionamiento de CSS

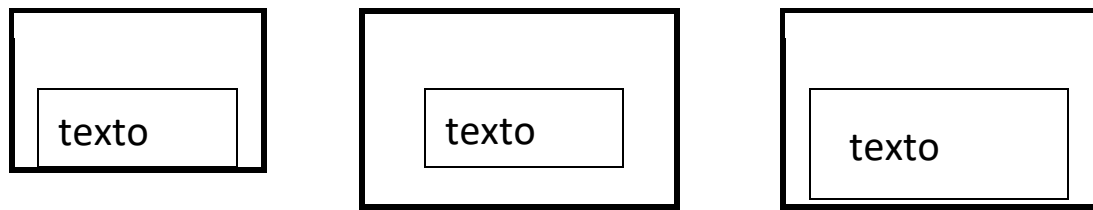
### 2.5.1 Modelo de cajas

Desde el punto de vista de CSS, todos los contenidos de una página web se representan por medio de una caja, que se crea automáticamente cuando se inserta un elemento HTML. Normalmente estas cajas no son visibles porque por defecto no se muestra ni su borde ni su color de fondo.

Lo que hace un selector es identificar la caja a la que se aplica el formato. Las cajas tienen:

- **Contenido (content):** El contenido de la caja: texto, imagen, etc.
- **Relleno (padding):** Espacio entre el contenido y el borde.
- **Borde (border):** El borde de la caja, la línea que encierra su contenido y su relleno.
- **Imagen de fondo (background image):** imagen que se muestra tras el contenido y el relleno.
- **Color de fondo (background color):** color que se muestra tras el contenido y el relleno.
- **Margen (margin):** La separación entre la caja y el resto de los elementos de la página (el elemento que contiene la caja o los otros elementos incluidos en su misma “caja padre”).





Los componentes de la caja se pueden dividir en cuatro partes: top, bottom, right y left. Por tanto, en la ilustración anterior, las siglas significan:

- LM -> Left Margin
- RB -> Right Border
- TP -> Top Padding
- etc.

Las hojas de estilo pueden aplicar formato (p.e. color, anchura, tipo de línea, etc.) a cualquiera de los componentes de esta caja, así como posicionar estas cajas en el conjunto de la página.

De este modo, si queremos que un elemento cualquiera muestre el borde en forma de puntos de color azul, podemos generar esta declaración:

```
p { border-style: dotted; border-color: blue }
```

De forma que si incluimos un párrafo en la página se mostrará así:



Podríamos también ampliar el relleno entre el contenido y el borde o especificar un tamaño de punto más pequeño:

```
p { border-style: dotted; border-width: 2px; border-color: blue; padding: 15px; }
```

De forma que veríamos el párrafo de la forma:

## Ejemplos de hojas de Estilo

### Ejemplo:

```
<html>
<head>
<style>
div {
  background-color: lightgrey;
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
</style>
</head>
<body>

<h2>Ejemplo del Modelo de Caja</h2>

<p>El modelo de caja CSS es esencialmente una que envuelve
cada elemento HTML. Consiste en: bordes, relleno, márgenes y
el contenido real.</p>

<div>Este texto es el contenido del cuadro. Hemos agregado un
relleno de 50 píxeles, un margen de 20 píxeles y un borde
verde de 15 píxeles. Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut aliquip ex ea commodo
consequat. Duis aute irure dolor in reprehenderit in
voluptate velit esse cillum dolore eu fugiat nulla pariatur.
Exceptional sint occassioneecat cupidatat non proident, sunt in
culpa qui officia deserunt mollit anim id est laborum.</div>

</body>
</html>
```

## Ejemplo del Modelo de Caja

El modelo de caja CSS es esencialmente una que envuelve cada elemento HTML. Consiste en: bordes, relleno, márgenes y el contenido real.

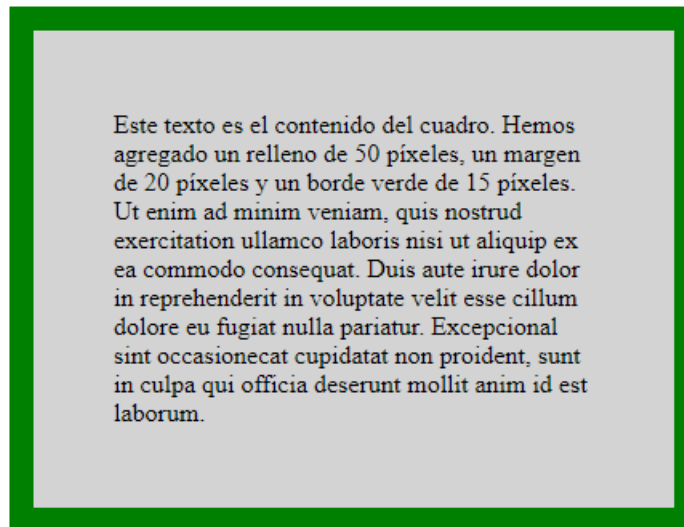


Figura 7 – Ejemplo Modelo de Caja

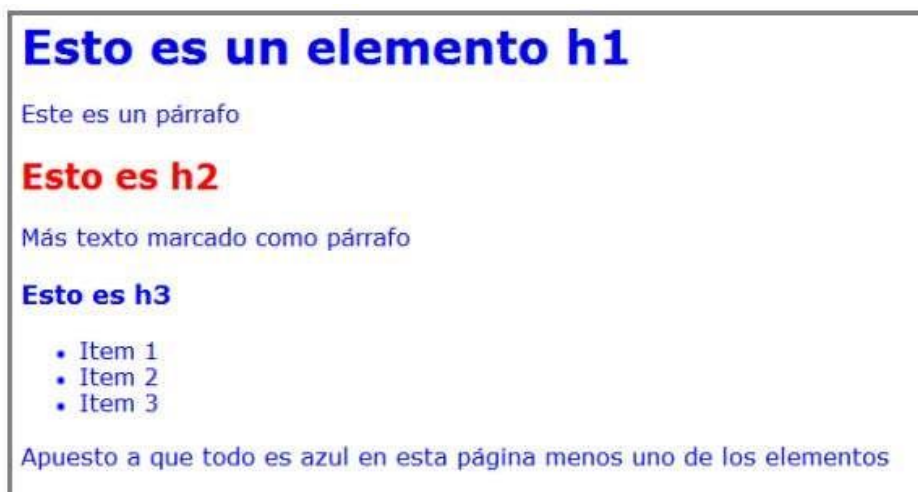
### 2.5.2 Herencia

Las cajas están contenidas dentro de otras cajas y heredan los estilos de la caja contenedora si no hay una regla específica para la caja contenida. Por ejemplo, si declaramos un tipo de letra para el elemento body, todos los elementos que están dentro, como h1, h2, p, etc. heredarán este tipo de letra.

Por lo tanto, si declaramos el siguiente código CSS:

```
body { color: blue; }  
h2 { color: red; }
```

Todo el texto que se incluya en el body tendrá color azul, excepto los h2 que tendrán color rojo:



### 2.5.3 Posicionamiento

Los navegadores web posicionan automáticamente todas las cajas HTML, pero CSS permite modificar este posicionamiento por defecto.

En CSS se definen cinco modelos distintos de posicionamiento:

- **Estático o normal:** El que utilizan los navegadores por defecto si no se les indica lo contrario.
- **Relativo:** Variante del anterior que consiste en posicionar la caja de forma estática e indicar el desplazamiento relativo con respecto a ese lugar.
- **Absoluto:** La posición de la caja se establece de forma absoluta con respecto al elemento que lo contiene. El resto de los elementos ignoran dicha posición.
- **Fijo:** Variante de la anterior en la que la caja se convierte en un elemento inamovible dentro de la pantalla. Aunque que se desplace el documento con las barras del navegador, el elemento no se moverá.
- **Flotante:** La caja “flota” sobre la página, de forma que no sigue el posicionamiento normal de la página (si tiene otra caja a continuación, ésta ocupará el lugar que le correspondía a la caja) y se sitúa lo más cerca posible del borde izquierdo o derecho de la página (según se le indique). Las cajas flotantes tienen en cuenta a las otras cajas flotantes existentes.
- **Sticky:** la posición del elemento depende de la barra de scroll: mientras su posición sea visible en la página se mostrará, pero si se mueve la barra de scroll hacia abajo, el elemento se seguirá mostrando pegado al margen superior de la página.

Los principales atributos para especificar el posicionamiento de un elemento son los siguientes:

Propiedad	Descripción	Ejemplo
<b>position</b>	Selecciona el tipo de posicionamiento de un elemento. Puede tomar los valores:  static (estático), relative (relativo), absolute (absoluto), fixed (fijo) o sticky.	position: relative
<b>top</b>	Indica el desplazamiento vertical desde arriba	top: 10px
<b>bottom</b>	Indica el desplazamiento vertical desde abajo	bottom: 10px
<b>right</b>	Indica el desplazamiento horizontal desde la derecha	right: 10px
<b>left</b>	Indica el desplazamiento horizontal desde la izquierda	left: 10px
<b>float</b>	Establece el tipo de posicionamiento flotante. Puede tomar los valores: left (izquierda), right (derecha) o none (anula el posicionamiento flotante).	float: left
<b>clear</b>	Establece qué lados del elemento no deben ser adyacentes a elementos flotantes.	clear: right

Ejemplo del uso de atributos de posicionamiento:

```

<!DOCTYPE html>
<html>
<head>
<style>
h2 {
  position: absolute;
  left: 100px;
  top: 150px;
}
</style>
</head>
<body>

<h1>Atributos de posicionamiento</h1>

<h2>Esto es una cabecera con "position: absolute"</h2>

<p>Con posicionamiento absoluto, un elemento se puede colocar
en cualquier lugar de una página. El encabezado a
continuación se coloca a 100 píxeles desde la izquierda de la
página y 150 píxeles desde la parte superior de la
página.</p>

</body>

```



```
</html>
```

## Atributos de posicionamiento

Con posicionamiento absoluto, un elemento se puede colocar en cualquier lugar de una página. El encabezado a continuación se coloca a 100 píxeles desde la izquierda de la página y 150 píxeles desde la parte superior de la página.

**Esto es una cabecera con "position: absolute"**

Figura 8 – Ejemplo del uso de atributos de posicionamiento

## 2.6 Visualización

Los atributos existentes para determinar la forma de visualizar un elemento en CSS son:

Propiedad	Descripción	Ejemplo
<b>display</b>	Permite indicar la forma en que se muestra un elemento. Si se oculta un elemento mediante esta propiedad desaparece completamente, y el resto de elementos ocuparán su lugar. Los valores que puede tomar son: inline (en línea), block (en bloque), none (oculto y sin ocupar un lugar en la página).	display: none
<b>visibility</b>	Permite ocultar un elemento de la página de forma que los otros elementos no ocupen su lugar. Sus posibles valores son: visible, hidden (oculto) y collapse (para ocultar columnas o filas de una tabla de forma que las siguientes ocupen su lugar).	visibility: hidden
<b>overflow</b>	Permite controlar los contenidos que sobresalen de un elemento. Puede tomar los valores: visible, hidden (ocultos), scroll (se muestran las barras de desplazamiento para poder visualizar el contenido sobrante) y auto (como scroll pero solo añade las barras de desplazamiento si es necesario).	overflow: scroll

<b>z-index</b>	Cuando colocamos unos elementos encima de otros, este atributo nos permite especificar el orden en que se visualizarán. Puede tener valor negativo o positivo. Los elementos con mayor valor se muestran por encima de los que tienen menor valor.	z-index: 3
----------------	--	------------

## 2.7 Atributos básicos

En los siguientes apartados veremos los principales atributos utilizados en CSS, tales como:

### 2.7.1 Atributos de tamaño para todos los elementos

Propiedad	Descripción	Ejemplo
<b>width</b>	Define la anchura del elemento.	width: 20px;
<b>height</b>	Define la altura del elemento.	height: 25px;
<b>max-width y min-width</b>	Anchura máxima y mínima del elemento	max-width: 900px; min-width: 500px;
<b>max-height y min-height</b>	Altura máxima y mínima del elemento	max-height: 900px; min-height: 500px;

### 2.7.2 Atributos de fuente

En esta sección veremos los diversos atributos que podemos usar con respecto a las fuentes de nuestro documento, que son:

Propiedad	Descripción	Ejemplo
<b>font-size</b>	Define el tamaño de la fuente.	font-size: 10pt;
<b>font-family</b>	Define la familia tipográfica del texto.	font-family: arial;
<b>font-weight</b>	Define el largo de los caracteres: normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900.	font-weight: bold;
<b>font-style</b>	Define el estilo de la fuente: normal, italic o oblique.	font-style: italic;
<b>font-variant</b>	Determina si la fuente es normal o mayúsculas pequeñas. Los valores que puede tomar son: normal o small-caps	font-variant: normal
<b>line-height</b>	Alto de la línea y por lo tanto, espaciado entre líneas.	line-height: 12pt
<b>font</b>	Permite establecer todas las propiedades anteriores en el siguiente orden: font-style, font-variant, font-weight,	font:italic 900 12px Verdana

font-size[line-height], font family. Los valores deben estar separados por espacios. No se requiere el uso de todos los valores
---

Un ejemplo de un documento XHTML en el que se utiliza este método para incluir formatos es:

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

  <head>
    <title>Ejemplo de atributos CSS de fuente</title>
    <style type="text/css">
      body { background-color: black; color:yellow;
font-family: courier }
      p { color: #ffffff; font:italic 900 12px Verdana;
}
    </style>
  </head>

  <body>
    <h3>Ejemplo del uso de atributos de fuente</h3>
    <p>El texto de cualquier elemento es de la familia
Courier y normal, salvo el del p&acute;rrafo que es Verdana,
negrita y de tamaño 12 px.</p>
  </body>
</html>
```

Que si guardamos como Ejemplo6.html y lo visualizamos en un navegador tendremos:

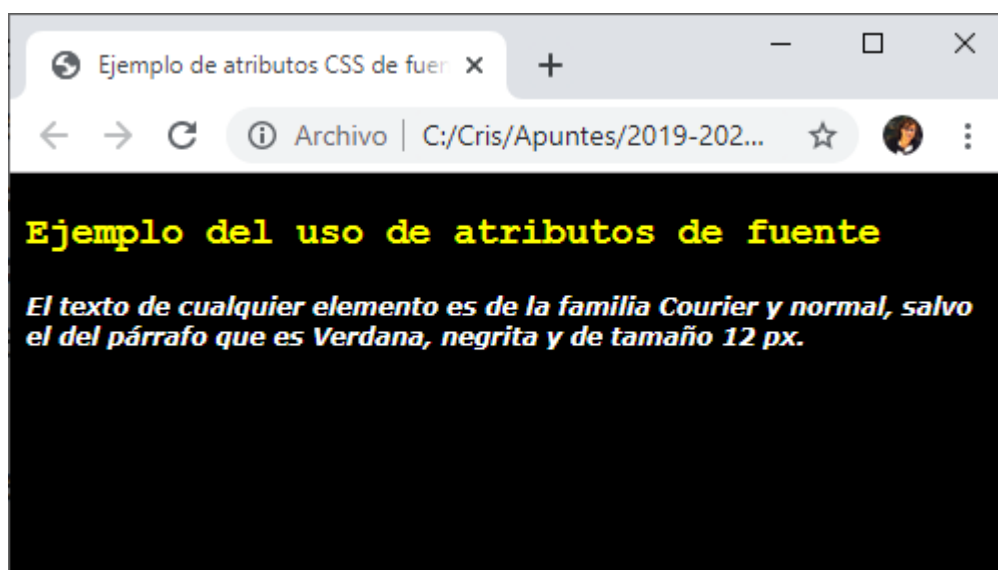


Figura 9 – Atributos de texto

Se pueden ver más ejemplos en: [https://www.w3schools.com/css/css\\_font.asp](https://www.w3schools.com/css/css_font.asp)

### 2.7.3 Atributos de texto

En el apartado anterior vimos los atributos relacionados con las fuentes y en este vamos a ver los relacionados con el texto.

Propiedad	Descripción
text-decoration	Establece si el texto está subrayado, sobrerayado o tachado. Los valores que puede tomar son: none, underline, overline, linethrough o blink.
text-align	Indica la alineación del texto. Aunque las hojas de estilo permiten el justificado de texto no funciona en todos los sistemas. Los valores que puede tomar son: left, right, center o justify.
text-indent	Determina el sangrado del texto. Los valores que toma son una longitud, en unidades CSS, o un porcentaje de la establecida.
text-transform	Permite transformar el texto, haciendo que tenga la primera letra en mayúsculas de todas las palabras, todo en mayúsculas o minúsculas. Los valores que puede tomar son: none capitalize uppercase lowercase initial inherit;
word-spacing	Determina el espaciado entre las palabras. Los valores que puede tomar es un tamaño
letter-spacing	Determina el espaciado entre letras. Los valores que puede tomar es un tamaño.
vertical-align	Establece la alineación vertical del texto. Los valores posibles son: baseline, sub, super, top, text-top, middle, bottom, textbottom o un porcentaje.
line-height	Altura de la línea. Puede establecerse mediante un tamaño o un porcentaje.
color	Define el color del texto.

Un ejemplo de uso de estos atributos es:

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

  <head>
    <title>Ejemplo de atributos CSS de texto</title>
    <style type="text/css">
      h3 { text-decoration:underline; text-align:
center; text-transform: capitalize }
      p { text-indent: 50%; }
    </style>
  </head>

  <body>
    <h3>Ejemplo del uso de atributos de texto</h3>
```

```

    <p>El texto de del encabezado de tercer nivel
    est&aacute; subrayado, centrado y la primera letra de cada
    palabra es may&uacute;cula.</p>
    <p>El p&aacute;rrafo est&aacute; tabulado</p>
  </body>
</html>

```

Lo que veríamos en un navegador como:

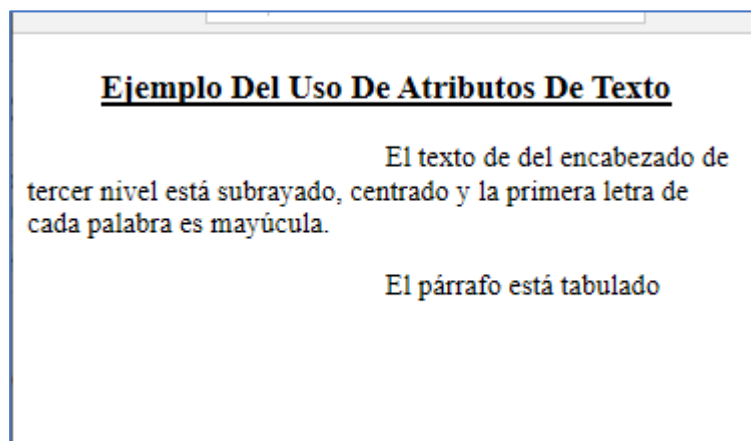


Figura 10 – Ejemplo de uso atributos de texto

Se pueden ver más ejemplos en: [https://www.w3schools.com/css/css\\_text.asp](https://www.w3schools.com/css/css_text.asp)

## 2.7.4 Atributos de fondo

Los atributos CSS para el fondo de un elemento son:

Propiedad	Descripción	Ejemplo
<b>background-color</b>	Define el color de fondo de un elemento.	background-color: #FF3245; background-color: blue;
<b>background-image</b>	Define la imagen de fondo de un elemento.	background-image: url(imagen_fondo.jpg);
<b>background-repeat</b>	Define la forma de repetir la imagen en caso de que no cubra el total del espacio del fondo: repeatx (en el eje horizontal), repeat-y (en el eje vertical), repeat (en ambos ejes), no-repeat (sin repetición).	background-repeat: repeat-x;

Ejemplo de uso:

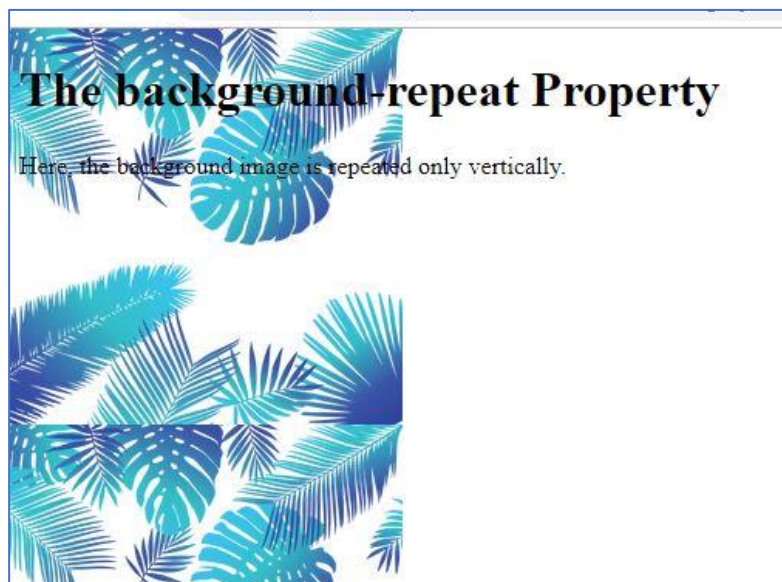
```

<!DOCTYPE html>
<html>
<head>
<style>

```

```
body {  
  background-image:  
  url("https://images.vexels.com/media/users/3/148192/isolated/  
  lists/86893a77fa8081fe8f65df3cbc90e9ad-fondo-de-hojas-  
  tropicales-azules.png");  
  background-repeat: repeat-y;  
}  
</style>  
</head>  
<body>  
  
<h1>The background-repeat Property</h1>  
<p>Here, the background image is repeated only vertically.</p>  
  
</body>  
</html>
```

Cuya visualización en un navegador nos mostraría:



*Figura 11 – Ejemplo del uso de atributos de fondo*

Podéis ver más ejemplos del uso de estos atributos en:

[https://www.w3schools.com/css/css\\_background.asp](https://www.w3schools.com/css/css_background.asp)

## 2.7.5 Atributos de caja

A continuación, vamos a ver otros atributos muy importantes y que utilizaremos muy frecuentemente y que no son ni más ni menos que los atributos de caja:

Propiedad	Descripción	Ejemplo
<b>margin</b>	Define el margen del elemento. Permite definir 1 (todo), 2 (vertical-horizontal) o 4 valores (superior, derecho, inferior e izquierdo en ese orden).	margin:10px 5px 15px 20px;
<b>margin-left</b>	Define el margen exterior a la izquierda.	margin-left: 20px;
<b>margin-right</b>	Define el margen exterior a la derecha.	margin-right: 20px;
<b>margin-top</b>	Define el margen exterior superior.	margin-top: 20px;
<b>margin-bottom</b>	Define el margen exterior inferior.	margin-bottom: 20px;
<b>padding</b>	Define el relleno del elemento. Permite definir 1 (todo), 2 (vertical-horizontal) o 4 valores (superior, derecho, inferior e izquierdo en ese orden).	padding:10px 5px 15px 20px;
<b>padding-left</b>	Define el relleno izquierdo del elemento.	padding-left: 20px;
<b>padding-right</b>	Define el relleno interior a la derecha.	padding-right: 20px;
<b>padding-top</b>	Define el relleno interior superior.	padding-top: 20px;
<b>padding-bottom</b>	Define el relleno interior inferior.	padding-bottom: 20px;
<b>border-width</b>	Define el largo del borde.	border-width: 2px;
<b>border-color</b>	Define el color del borde.	border-color: #AA1356; border-color: red;
<b>border-style</b>	Define el tipo de borde de la caja. Puede tomar los valores: none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset.	border-style: double;
<b>border</b>	Define en una fila las tres características del borde (border-width, border-style y border-color)	border: 5px solid red;
<b>border-radius</b>	Redondea el borde	border-radius: 4px;
<b>resize</b>	para permitir o impedir cambiar el tamaño de un elemento	resize:none;

¿Cuál sería un posible código XHTML para el documento de la imagen que se muestra a continuación?

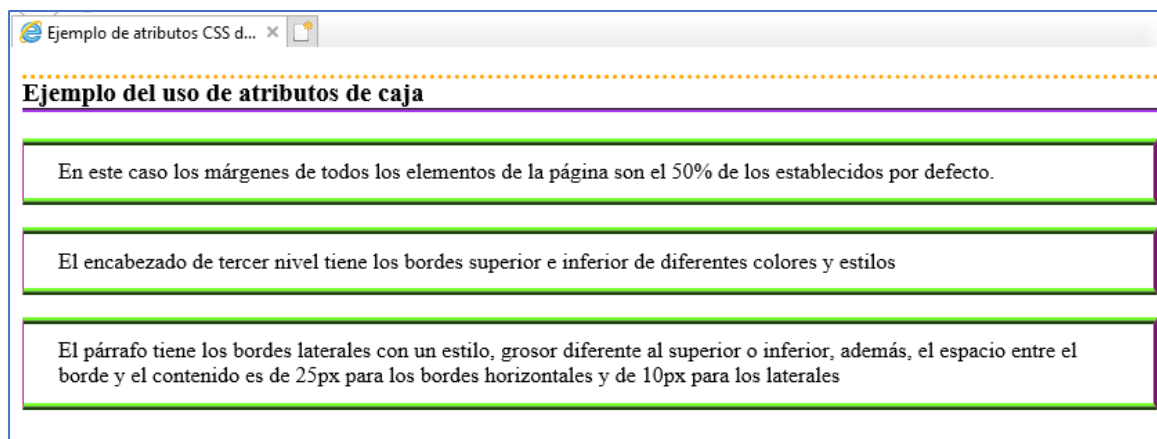


Figura 12 – Ejemplo de atributos de caja

Una posible forma de hacerlo sería:

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

  <head>
    <title>Ejemplo de atributos CSS de caja</title>
    <style type="text/css">
      body{ margin: 50%, 50%, 50%, 50%; }
      h3 { border-top-color: orange; border-bottom-
color: #b428ff; border-top-style: dotted; border-bottom-
style: groove; }
      p { border-color: #73ff28 #ff28df #73ff28 #ff28df;
border-style: ridge outset ridge outset; border-width: thick
medium thick thin; padding: 10px 25px 10px 25px;}
    </style>
  </head>

  <body>
    <h3>Ejemplo del uso de atributos de caja</h3>
    <p>En este caso los márgenes de todos los
elementos de la página son el 50% de los establecidos
por defecto.</p>
    <p>El encabezado de tercer nivel tiene los bordes
superior e inferior de diferentes colores y estilos </p>
    <p>El párrafo tiene los bordes laterales con un
estilo, grosor diferente al superior o inferior,
además, el espacio entre el borde y el contenido es de
25px para los bordes horizontales y de 10px para los
laterales </p>
  </body>
</html>
```



## 2.7.6 Atributos de tabla

Propiedad	Descripción	Ejemplo
<b>border-collapse</b>	Especifica cómo se mostrarán los bordes: collapse (fusiona bordes de tablas adyacentes), separate (cada celda muestra sus 4 bordes), inherit (hereda)	border-collapse:collapse;
<b>border-spacing</b>	Establece la separación entre bordes (solo con border-collapse:separate). Permite definir 1 (todo) o 2 valores (horizontal y vertical).	border-spacing:5px;
<b>empty-cells</b>	Tratamiento de las celdas vacías: show (muestra background y border), hide (no muestra nada), initial (valor por defecto), inherit (hereda)	empty-cells:hide;
<b>caption-side</b>	Posición del título: top, bottom, inherit o initial	caption-side:bottom;

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<style>
table, td, th {
    border: 1px solid black;
}

table {
    border-collapse: collapse;
    width: 100%;
}

th {
    height: 50px;
}
</style>
</head>
<body>

<h2>Atributos height y width</h2>
<p>Establecer el ancho de la tabla y el alto de la fila del
encabezado de la tabla:</p>
```

```
<table>
  <tr>
    <th>Nombre</th>
    <th>Apellido</th>
    <th>Ganancia</th>
  </tr>
  <tr>
    <td>Juan</td>
    <td>Martinez</td>
    <td>100€</td>
  </tr>
  <tr>
    <td>Lois</td>
    <td>Salgado</td>
    <td>150€</td>
  </tr>
</table>

</body>
</html>
```

### Atributos height y width

Establecer el ancho de la tabla y el alto de la fila del encabezado de la tabla:

Nombre	Apellido	Ganancia
Juan	Martinez	100€
Lois	Salgado	150€

Figura 13 – Ejemplo del uso de los atributos de tabla

Podéis ampliar los ejemplos de atributos de tablas en:

[https://www.w3schools.com/css/css\\_table.asp](https://www.w3schools.com/css/css_table.asp)

## 2.7.7 Enlaces

Los enlaces tienen 4 pseudoclases que permiten especificar:

Propiedad	Descripción	Ejemplo
<b>:link</b>	No visitado	a:link {border: medium solid #CC0000;}
<b>:visited</b>	Visitado	a:visited {color:green}
<b>:hover</b>	Cuando el ratón está por encima del enlace	a:hover { text-decoration: none; }
<b>:active</b>	Cuando se hace clic en el enlace	A:active {background-color: rgb(255,174,174);}

En ocasiones, puede resultar útil incluir un pequeño icono al lado de un enlace para indicar el tipo de contenido que enlaza, como por ejemplo un archivo comprimido o un documento con formato PDF.

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<style>
/* enlace no visitado */
a:link {
  color: red;
}

/* enlace visitado */
a:visited {
  color: green;
}

/* ratón sobre el enlace */
a:hover {
  color: hotpink;
}

/* enlace pulsado */
a:active {
  color: blue;
}
</style>
</head>
<body>

<p><b><a href="https://www.w3schools.com/css/default.asp"
  target="_blank">Esto es un enlace</a></b></p>
<p><b>Nota:</b> a:hover TIENE que aparecer después de a:link y
  a:visited
  en la definición CSS para ser efectivo.</p>
```

```
<p><b>Nota:</b> a:active TIENE que aparecer después de a:hover  
en la definición CSS para ser efectivo.</p>  
  
</body>  
</html>
```

Esto es un enlace

**Nota:** a:hover TIENE que aparecer después de a:link y a:visited en la definición CSS para ser efectivo.

**Nota:** a:active TIENE que aparecer después de a:hover en la definición CSS para ser efectivo.

Figura 14- Ejemplo del uso de atributos de enlaces

## 2.8 CSS 3

CSS3 es una ampliación de CSS 2.1, que acepta todas las funcionalidades de la versión previa y añade nuevas funcionalidades. Esta versión de CSS fue dividida en módulos independientes, de forma que permitiese trabajar en cada módulo modo de manera independiente.

Algunas de las nuevas funcionalidades que aporta son:

- Superposición de imágenes de fondo.
- Esquinas redondeadas.
- Gradientes y degradados de colores (background: linear-gradient(to right, red, yellow);)
- Incorporar imágenes en los bordes - Sombras.
- Transparencias de color.
- Definir texto en varias columnas.
- División de las palabras con guiones a final de línea.

- Instalación de otros tipos de fuentes (@font-face{font-family: 'Fira';src:url('../fonts/fira-sans/FiraSans-Four.otf');})
- Nuevo modelo de caja flexible.
- Nuevos selectores.

## 2.9 Herramientas de desarrollo

Un **preprocesador de CSS** es una herramienta que nos permite escribir pseudo-código CSS que luego será convertido a CSS real. Podríamos decir que se trata de un lenguaje de programación que genera código CSS. Los preprocesadores de CSS más usados actualmente son:

- Sass (<http://sass-lang.com>)
- LESS (<http://lesscss.org>)
- Stylus (<https://learnboost.github.io/stylus>)

En los siguientes ejemplos se usará LESS para mostrar los ejemplos de uso del procesador, por ser más fácil de instalar y tener una sintaxis más estricta.

El uso de preprocesadores CSS permite:

- Usar variables (como colores o estilos de texto) y utilizarlas en el documento, de forma que podríamos emplear textos del mismo tipo y diferentes tamaños, por ejemplo, y si quisiésemos cambiar el tipo de texto, lo haríamos en un único punto.

### Código LESS

```
@corPpal: #3366CC;
@estiloBorde: dotted;
@largoBorde: 3px;
body {
  color: @corPpal;
  border: @largoBorde
  @estiloBorde @corPpal;
}
```

### Código CSS generado

```
body {
  color: #3366cc;
  border: 3px dotted
#3366cc; }
```

- Definir código anidado: si queremos definir un estilo para un elemento que tenga un antecedente determinado, lo podemos hacer dentro del código del propio antecedente:

#### Código LESS

```
section {
  color: blue;
  h1 {
    font-size: 1.5em;
    strong {
      color: red;
    }
  }
}
```

#### Código CSS generado

```
section {
  color: blue;
}
section h1 {
  font-size: 1.5em;
}
section h1 strong {
  color: red;
}
```

- Utilizar mixins (los equivalentes de los preprocesadores CSS a las funciones de los lenguajes de programación), que permiten reutilizar código e incluso emplear parámetros para adaptar el resultado en función del valor que se le pase.

#### Código LESS

```
.cuadro(@borde: 2px) {
  border: @borde solid #F00;
  color: #F00;
}
.cuadroProfesor {
  padding: 20px;
  margin: 4px;
  .cuadro(); /* Si no paso
parámetros, usa el valor por
defecto */
}
.cuadroAlumno {
  padding: 20px;
  margin: 4px;
  .cuadro(1px); /* aplica los
estilos definidos pero con
borde 1px */
}
```

#### Código CSS generado

```
.cuadroProfesor { padding:
20px; margin: 4px;
border: 2px solid #ff0000;
color: #F00;
}
.cuadroAlumno { padding:
20px; margin: 4px;
border: 1px solid #ff0000;
color: #F00;
}
```

- Herencia: al aplicar algunos estilos comunes a varios elementos y después tratar de aplicar estilos individuales a mayores, el código CSS queda redundante o desordenado. Los preprocesadores CSS permiten usar la herencia de forma que el resultado es más legible:

#### Código LESS

```
.bloque {
  margin: 10px 5px;
}
p {
  .bloque;
  color: green;
}
ul, ol {
  .bloque;
  color: darkgreen;
}
```

#### Código CSS generado

```
.bloque {
  margin: 10px 5px;
}
p {
  margin: 10px 5px;
  color: green;
}
ul, ol {
  margin: 10px 5px;
  color: darkgreen;
}
```

- Importar contenido de archivos. CSS permite importar fuentes o otros estilos de otros archivos CSS, pero eso supone que la página web cargue más lenta. Para evitar este problema, manteniendo las ventajas de mantener los estilos en archivos separados, los preprocesadores CSS permiten importar ficheros con extensión .less, de forma que al generar el CSS resultante incorpora en un único fichero todos los estilos definidos en los archivos .less importados.

#### Código LESS

```
@import
"aimportar.less";
p {
  color: aquamarine;
}
```

#### Código CSS generado

```
body {
  background-color: blueviolet;
}
p {
  color: aquamarine;
}
```

- Aplicar funciones a colores, de modo que podemos transformar los colores en otros similares pero con un color más claro, oscuro, saturado, desaturado, etc.

#### Código LESS

```
texto1 {
  color: lighten(@color, 10%);/*
```

#### Código CSS generado

```
texto1 {
  color: #668cff;
```

```

color 10% más claro que @color */
} texto2 {
    color: darken(@color, 10%); /*
color 10% más oscuro que @color */
} texto3 {
    color: saturate(@color, 10%); /*
color 10% más saturado que
@color */
}
Texto4 {
    color: spin(@color, 10); /* un
tono un 10% mayor que @color */
}
@color1: #ffff66;
@color2: #0000cc; Texto5 {
    color: mix(@color1, @color2); /*
mezcla de @color1 y @color2 */ }

```

```

}
texto2 {
    color: #0040ff;
} texto3 {
    color: #3366ff;
} texto4 {
    color: #3344ff;
}
Texto5 {
    color: #808099;
}

```

- Definir operaciones y usar funciones de cadenas, de modo que podemos calcular el tamaño de un elemento sumando determinados pixels al tamaño de otro elemento, por ejemplo:

#### Código LESS

```

.peImagen {
    width: image-width("imagen.jpg")
+ 10px;
background-color: coral;
}

```

#### Código CSS generado

```

.peImagen {
    width: 650px;
    background-color: coral;
}

```

- Anidar media queries en las clases: las media queries son reglas que permiten la aplicación de unos estilos u otros en función de determinadas condiciones (el ejemplo más habitual: de cual sea el dispositivo de visualización). En lugar de tener que concentrar las media queries al final de la hoja de estilos, los preprocesadores permiten incluirlas en las definiciones de estilos de elementos, clases, etc.

#### Código LESS

```

body {
    background-color: blue;
}
@media (max-width: 800px) {

```

#### Código CSS generado

```

body {
    background-color: blue;
}

```



```
background-color: lightblue;
}
```

```
@media (max-width: 800px) {
  body {
    background-color:lightblue;
  }
}
```

## **ANEXO I: Materiales**

### **I. Textos de apoyo o de referencia**

- Wikipedia. <http://www.wikipedia.org>
- W3schools CSS: <https://www.w3schools.com/css/default.asp>

### **II. Recursos didácticos**

- Apuntes en el aula virtual.
- Ordenador personal, con navegador web y conexión a internet.
- Software para elaboración de documentos de texto.