

TEMA 3. CSS

1º DAW
IES Fco. Ayala
Blanca Ceballos

CONTENIDO

- INTRODUCCIÓN
- PRIORIDAD
- SINTAXIS
- SELECTORES
- MODELO DE CAJAS
- FORMATO PARA:
 - TEXTO
 - LISTAS
 - TABLAS
 - LAYOUT
- MISCELÁNEA

CONTENIDO

- **INTRODUCCIÓN**
- PRIORIDAD
- SINTAXIS
- SELECTORES
- MODELO DE CAJAS
- FORMATO PARA:
 - TEXTO
 - LISTAS
 - TABLAS
 - LAYOUT
- MISCELÁNEA

INTRODUCCIÓN

CSS: Cascading Style Sheets - Hojas de estilo en cascada

- No es un lenguaje de marcas, es un **lenguaje de presentación**, por lo tanto varía la sintaxis de HTML.
- Su principal objetivo es manejar el aspecto y formato de los documentos, liberando a HTML de las tareas de presentación.
- Ventajas de separar el contenido de la presentación:
 - Disminuye el código a escribir, evita duplicados.
 - Facilita la generación de código y su mantenimiento.
 - Mejora la legibilidad de los documentos.

INTRODUCCIÓN

CSS surge en torno al 1970, pero su uso generalizado se produce más tarde con el auge de Internet.

Al igual que con HTML, la ausencia de estándar hizo que hubiera varias formas de establecer el estilo, provocando diferencias de visualización en los navegadores. Fue el WC3 quien establece el primer estándar, **CSS1**, en 1996.

Las mejoras de esta versión aparecen en **CSS2** en 1998, y esta versión sufrió una revisión, **CSS2.1**, que es la vigente.

Desde el 1998 hasta la actualidad, se trabaja en implantar **CSS3**, aunque la mayoría de navegadores ya lo soportan.

CONTENIDO

- INTRODUCCIÓN
- **PRIORIDAD**
- SINTAXIS
- SELECTORES
- MODELO DE CAJAS
- FORMATO PARA:
 - TEXTO
 - LISTAS
 - TABLAS
 - LAYOUT
- MISCELÁNEA

PRIORIDAD

A parte de las hojas de estilo que crean los diseñadores, existen otras dos: las del navegador y las del usuario.

La del **navegador** es la primera que se aplica y se establece para el estilo por defecto.

La del **usuario** se configura mediante opciones avanzadas del navegador. Son útiles para mayores o con deficiencia visual, ya que aumentan el tamaño de la letra o el zoom.

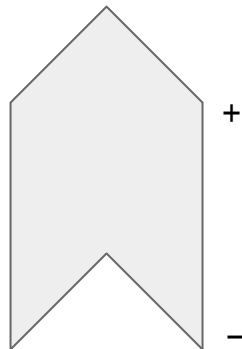
PRIORIDAD

La jerarquía a aplicar es:

Hojas de estilo del **diseñador**

Hojas de estilo del **usuario**

Hojas de estilo del **navegador**



Los valores por defecto del navegador pueden variar de uno a otro, por lo que es recomendable inicializar todos los valores, aunque sean 0 o nulos.

CONTENIDO

- INTRODUCCIÓN
- PRIORIDAD
- **SINTAXIS**
- SELECTORES
- MODELO DE CAJAS
- FORMATO PARA:
 - TEXTO
 - LISTAS
 - TABLAS
 - LAYOUT
- MISCELÁNEA

SINTAXIS - INCLUIR CSS EN UN DOCUMENTO

Existen tres formas de incluir CSS en un documento:

1. **CSS en línea:** usar el atributo `style` en la propia línea +
(no se utiliza llaves).
2. **CSS interno:** se incluye el código en la cabecera, dentro de la etiqueta `style`.
3. **CSS externo:** se crea un archivo css externo y se vincula al documento mediante `<link>` o `@import`.

* Esta prioridad puede variar con “**!important**” en el valor del atributo, y sería el que prevalecería.



Prioridad -

SINTAXIS - INCLUIR CSS EN UN DOCUMENTO

Ejemplos:

1. **CSS en línea:**

```
<p style="font-family: Verdana;"> hola mundo </p>
```

2. **CSS interno:**

```
<head>  
  <style type="text/css">  
    body {font-family: Courier New;}  
    h1 {font-family: Arial; font-size: x-large}  
  </style>  
</head>
```

SINTAXIS - INCLUIR CSS EN UN DOCUMENTO

Ejemplos:

3. **CSS externo:**

- En la cabecera:

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

ó

```
<style type="text/css">  
  @import "/css/estilos.css";  
</style>
```

SINTAXIS - INCLUIR CSS EN UN DOCUMENTO

Ejemplos:

3. **CSS externo:**

- Doc. externo será del tipo:

```
body {margin: 0px;}  
td {color: #000000;  
    font-size: 12 px;}  
a {color: #FF6600;  
   font-weight: bold;}
```

SINTAXIS - CONSTRUIR REGLAS CSS

Cada regla sirve para definir un estilo. Consiste en la unión del selector (elemento) más una declaración entre llaves. A su vez, cada declaración se compone de una o más parejas de atributo, valor y finalizado en ;. Para los comentarios se usa **`/* Comentarios */`**.

```
selector { atributo1: valor1;  
          atributo2: valor2;  
          ...  
          atributoN: valorN;}
```

```
Ej.: h1 {font-family: Verdana;  
        color: red;}
```

CONTENIDO

- INTRODUCCIÓN
- PRIORIDAD
- SINTAXIS
- **SELECTORES**
- MODELO DE CAJAS
- FORMATO PARA:
 - TEXTO
 - LISTAS
 - TABLAS
 - LAYOUT
- MISCELÁNEA

SELECTORES (ELEMENTOS)

En CSS existen distintos tipos de selectores:

- Universal
- De tipo
- Descendiente
- Hijo
- Adyacente
- De atributos
- De clase
- ID
- Pseudo-clase
- Pseudo-elementos

SELECTORES (ELEMENTOS)

- **Universal:** selecciona todos los elementos de la página.
Se indica con un `*`.
Ej.: `* { margin: 0px }`
- **De tipo:** selecciona los elementos que coincidan con la etiqueta.
Ej.: `p {font-family: Verdana; color: red;}`
- **Descendiente:** D es un elemento descendiente del elemento E, si D está contenido entre la apertura y cierre de E:
`<E> <D> ___ </D> </E>`
El selector descendiente se construye con 2 o más selectores simples separados por un espacio:
E D {declaración}, selecciona todos los elementos D que son descendientes de E a cualquier nivel.

SELECTORES (ELEMENTOS)

- **Descendiente:** `<E> <D> ___ </D> </E>`

E D {declaración}

Ej.: `p a {font-size: 50px}`

En body:

```
<p> Texto1 </p>
```

```
<p><a> Texto2 </a></p>
```

```
<p><span class="nieto"><a> Texto3 </a></span></p>
```

¿Qué sucede en este código?

SELECTORES (ELEMENTOS)

- **Descendiente:** `<E> <D> ___ </D> </E>`

`E D {declaración}`

Si quisiéramos excluir de la selección a los hijos, y seleccionar a partir de los nietos en adelante, combinaríamos con el selector universal:

Ej.: `p * a {font-size: 50px}`

```
<p> Texto1 </p>
```

```
<p><a> Texto2 </a></p>
```

```
<p><span class="nieto"><a> Texto3 </a></span></p>
```

¿Qué sucede en este código?

SELECTORES (ELEMENTOS)

- **Hijo:** selecciona el primer descendiente de un elemento, es decir, al hijo, excluyendo de la selección al resto de descendientes.

Ej.: **p > a** {font-size: 50px}

```
<p> Texto1 </p>
```

```
<p><a> Texto2 </a></p>
```

```
<p><span class="nieto"><a> Texto3 </a></span></p>
```

¿Qué sucede en este código?

SELECTORES (ELEMENTOS)

- **Adyacente:** selecciona elementos que son “hermanos” (tienen el mismo padre), y son adyacentes (consecutivos) en el código:

E + M {declaración;} → Selecciona a todos los elementos de tipo M, tales que E y M tienen el mismo padre y E precede inmediatamente a M en el código (salvo comentarios).

Ej.: `h1 + h2 {font-size: 50px}`

SELECTORES (ELEMENTOS)

Ej.: h1 + h2 {font-size: 50px}

```
<body>
  <h1>Título1</h1>
  <h2>Subtítulo</h2>
  ...
  <h2>Otro subtítulo</h2>
  ...
</body>
```

¿Qué sucede en este código?

SELECTORES (ELEMENTOS)

- **De atributos:** se puede realizar de 4 formas distintas:
 - **[atributo]:** selecciona los elementos que tengan ese atributo independientemente del valor que tome (puede tener otros atributos).
 - **[atributo = valor]:** selecciona los elementos que tengan al atributo con el valor especificado (puede tener otros atributos).
 - **[atributo ~= valor]:** selecciona los elementos que tengan al atributo con al menos uno de los valores como el especificado (puede tener otros atributos).
 - **[atributo |= valor]:** selecciona los elementos que tengan el atributo y que la palabra comience con el valor especificado (sólo sirve para el atributo lang).

Ej.:
`span[class] {font-family: Arial;}`
`span[class="azulon"] {color: blue;}`
`span[class~="grandote"] {font-size: 50px;}`
`span[lang|= "es"] {color: red;}`

SELECTORES (ELEMENTOS)

- **De clase:** selecciona una instancia en particular de un elemento. Se usa el atributo **class**.
Se construye de la forma **selector_simple.clase** ó **.clase:**
Ej.:
`p.grandote{font-size: 50px}` ó `.grandote{font-size: 50px}`

```
<h3 class = "grandote"> Texto1 </h3>
```

```
<p class = "grandote"> Texto2 </p>
```

```
<p class = "peque"> Texto3 </p>
```

¿Qué sucede en un caso y en el otro?

SELECTORES (ELEMENTOS)

- **ID:** selecciona una sólo línea de todo el documento. Se puede utilizar el selector de clase, pero es menos eficiente. Se usa el atributo **id**. Se construye de la forma **#clase:**

Ej.:

```
#peque {font-size: 8px}
```

```
<p id="peque"> Texto4 </p>
```

SELECTORES (ELEMENTOS)

- **Pseudo-elementos:** los de CSS2 son (sintaxis de CSS3):
 - **::first-line** se aplica a la primera línea de texto.
 - **::first-letter** se aplica a la primera letra de texto.
 - **::before** sirve para generar texto antes del contenido de un elemento.
 - **::after** sirve para generar texto después del contenido de un elemento.

Ej.: `h1::before {content: "Antes del encabezado ---"};`
`p::first-line {font-size: 20px;}`

Ejercicio 1: vamos a probar todos los selectores vistos hasta ahora, a excepción de los atributos, en un doc con CSS interno, es decir, en la etiqueta `style`.

SELECTORES (ELEMENTOS)

Ejercicio 2: Pseudo-clase. → Se entrega en Classroom.

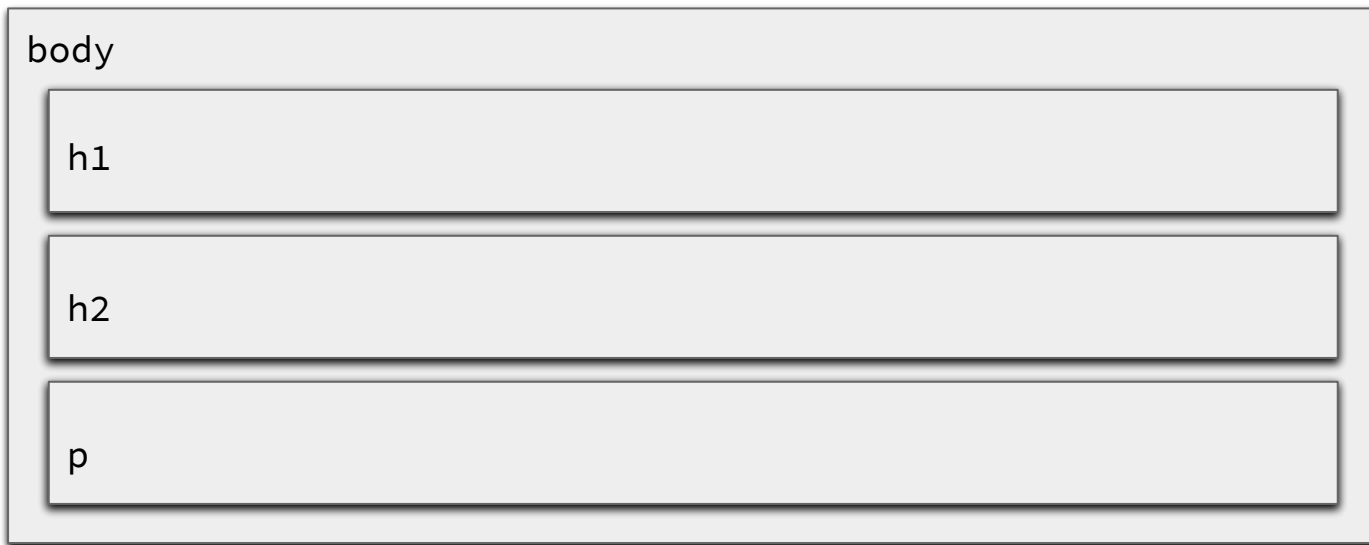
Cada alumn@ realizará la búsqueda de los selectores pseudo-clase (:link, :visited, :hover, :active, :focus), y pondrá un ejemplo práctico de cada uno de ellos.

CONTENIDO

- INTRODUCCIÓN
- PRIORIDAD
- SINTAXIS
- SELECTORES
- **MODELO DE CAJAS**
- FORMATO PARA:
 - TEXTO
 - LISTAS
 - TABLAS
 - LAYOUT
- MISCELÁNEA

MODELO DE CAJAS

Es un elemento fundamental en CSS, ya que todos los elementos del documento HTML se representan automáticamente mediante una caja (a excepción de *head*).

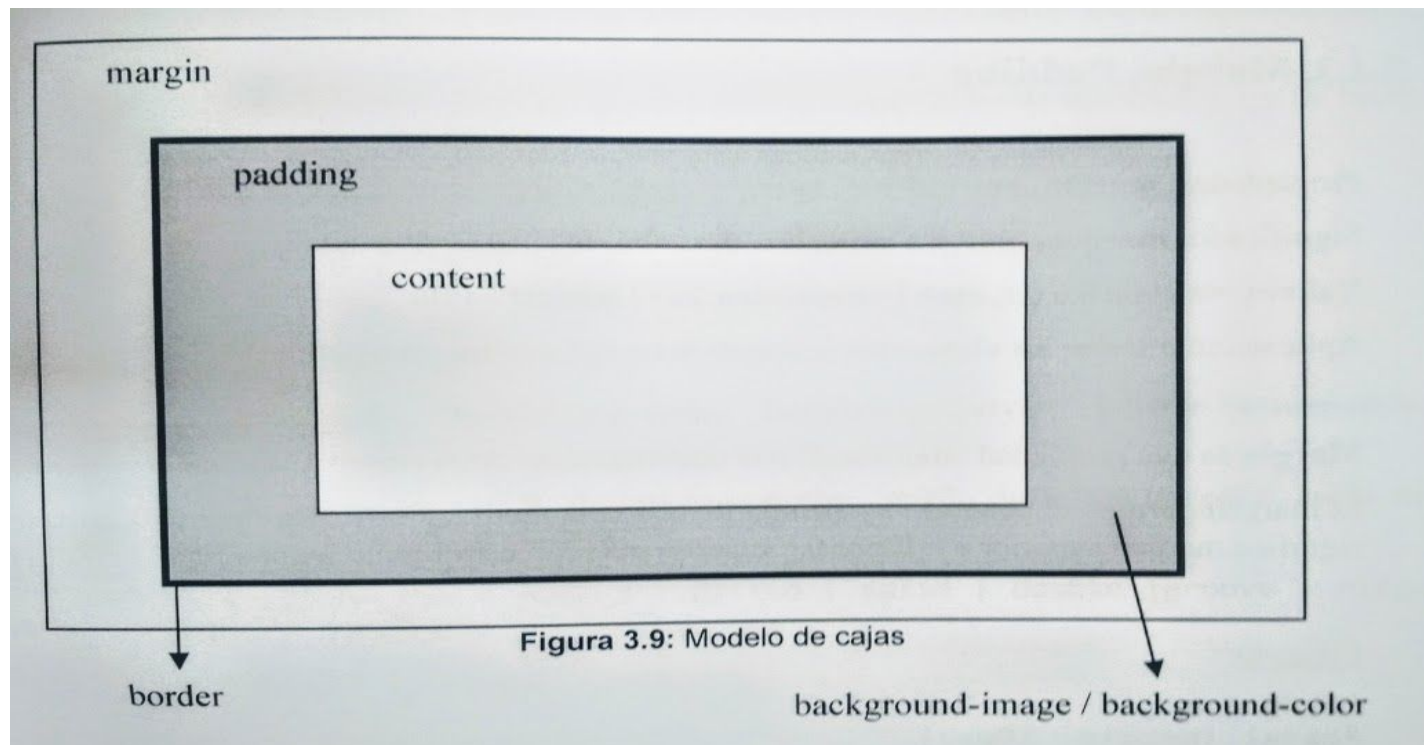


MODELO DE CAJAS

Las partes que componen cada caja son, por orden de visualización:

1. **Contenido** (Content): texto, imágenes, listas, etc.
2. **Relleno** (Padding): espacio entre el contenido y el borde.
3. **Borde** (Border): línea que encierra completamente el contenido y su relleno.
4. **Imagen de fondo** (Background-image): imagen que se muestra por detrás del contenido y el espacio de relleno.
5. **Color de fondo** (Background-color): color que se muestra por detrás del contenido y el espacio de relleno.
6. **Margen** (Margin): espacio libre entre la caja y las cajas sucesivas.

MODELO DE CAJAS



MODELO DE CAJAS

A tener en cuenta:

- Padding y margin son transparentes.
- En el espacio de padding se muestra el color/imagen de fondo.
- En el espacio de margin se muestra el color/imagen del elemento padre.
- Si ningún elemento padre tiene definido un color/imagen de fondo, se muestra el de la página.
- Si una caja tiene imagen y color de fondo, la imagen prevalece sobre el color.
- Si la imagen de fondo no cubre totalmente la caja, o tiene zonas transparentes, entonces se visualiza el color de fondo.

MODELO DE CAJAS

- **Ancho y alto:** width/height

Valores: n° px | n° % | inherit | auto

pixeles | porcentaje con respecto al padre |
heredado del padre | valor por defecto del navegador.

- **Margin:** margin, margin-top, margin-right, margin-bottom, margin-left

Valores: n° px | n° % | inherit | auto

pixeles | porcentaje con respecto al padre |
heredado del padre | valor por defecto del navegador.

MODELO DE CAJAS

- **Padding:** padding, padding-top, padding-right, padding-bottom, padding-left

Valores: n° px | n° % | inherit | auto

pixeles | porcentaje con respecto al padre |
heredado del padre | valor por defecto del navegador.

MODELO DE CAJAS

- **Borde:** se les puede establecer tres tipos de propiedades.
 - **Ancho:** border-width, border-top-width, border-right-width, border-bottom-width, border-left-width.
Valores: n° px | thin | medium | thick | inherit
 - **Color:** border-color, border-top-color, border-right-color, border-bottom-color, border-left-color.
Valores: color | transparent | inherit
 - **Estilo:** border-style, border-top-style, border-right-style, border-bottom-style, border-left-style
Valores: none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset | inherit

MODELO DE CAJAS

- **Color:** se puede especificar con palabras reservadas o de forma numérica. Ej.: `p {color: yellow}` ó `p {color: #ff0}`
- **Fondo:** zona ocupada por el contenido y el relleno (content + padding). Para hacerlo de la *página entera*, se debe de hacer en el elemento *body*.
 - **background-color:** color de fondo.
 - **background-image:** imagen de fondo.
 - **background-repeat:** repite la imagen de fondo hasta rellenar el elemento.
 - **background-position:** indica el desplazamiento de la imagen desde la esquina superior izquierda, mediante el desp. horizontal y el vertical.
 - **background-attachment:** comportamiento de la imagen con respecto al scroll-bar.

MODELO DE CAJAS

- **Posicionamiento:** hay 5 formas de establecer el posicionamiento en pantalla de una caja: estático, relativo, absoluto, fijo y flotante. Para hacerlo, se usa el atributo position, top, bottom, right, left y float.
- position: establece la posición de la caja en la pantalla
Valores: static | relative | absolute | fixed | inherit
- top, bottom, right, left: establece el desplazamiento vertical y horizontal de la caja
Valores: n° px | n° % | inherit | auto

MODELO DE CAJAS

- float: establece la posición flotante de la caja

Valores: none | left | right | inherit

Posicionamiento estático: si los elementos son de bloque las cajas se colocan verticalmente una debajo de otra, dentro del elemento contenedor. Si los elementos a diferenciar son de línea, las cajas se colocarán una junta a otra horizontalmente dentro del elemento de bloque. Si las cajas no caben en una línea, entonces ocuparán las siguientes, y sino se alinean horizontalmente.

MODELO DE CAJAS

Posicionamiento relativo: {position: relative}.

Consiste en desplazar la caja una cantidad con respecto al posicionamiento estático.

Posicionamiento absoluto: {position: absolute}.

Desplaza la caja una cantidad con respecto a su contenedor.

Posicionamiento fijo: {position: fixed}.

similar al absoluto, pero cuando se desplaza la página, las cajas se mantienen fijas. Útil para encabezados o pies de página, o menús de navegación.

MODELO DE CAJAS

Posicionamiento flotante: {float: left | right | inherit | none}.

Desplaza las cajas todo lo posible hacia la izquierda o derecha de la línea donde se encuentra. El resto de cajas ocupan el espacio dejado por el float, y no hay solapamiento, float lo tiene en cuenta. Útil para imágenes.

Ej.: `img {width: 60px; margin: 20px; float: right;}`

MODELO DE CAJAS

Visualización: hay cuatro propiedades: display, visibility, z-index y overflow.

Display: inline, block, none, list-item, run-in, inline-block, table, inline-table, table-row-group, table-header-group, table-footer-group, table-row, table-column-group, table-column, table-cell, table-caption, inherit.

Inline/Block: muestra el elemento como elemento de línea/bloque. (Enlaces en forma de menú)

None, permite que el elemento desaparezca.

MODELO DE CAJAS

Visibility: visible, hidden, collapse, inherit.

Visible: muestra la caja, por defecto.

Hidden: oculta la caja, pero deja el hueco.

Collapse: se utiliza para filas/columnas de una tabla, y se muestran varios contenidos en una misma fila/columna.

Inherit: heredado.

MODELO DE CAJAS

Overflow: controla el desbordamiento de una caja. Valores: visible, hidden, scroll, auto, inherit. Muy útil para los elementos *div*.

Visible: permite que el contenido se desborde, por defecto.

Hidden: oculta lo que se desborda.

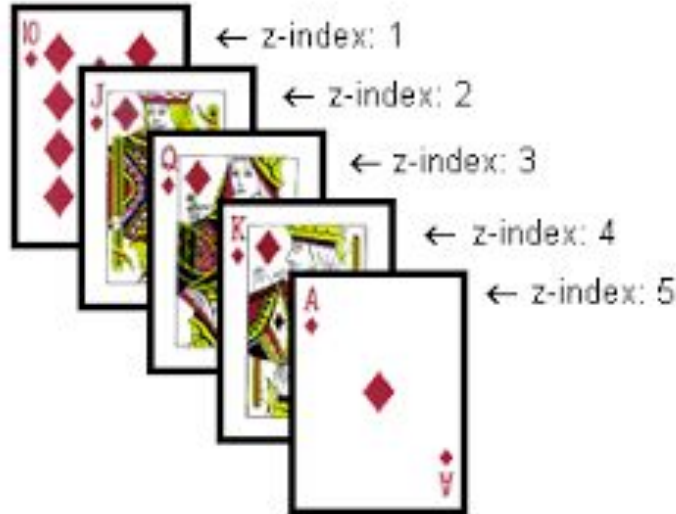
Scroll: mantiene el tamaño de la caja, y crea barras de desplazamiento.

Auto: mantiene el tamaño de la caja, y crea barras de desplazamiento si es necesario.

Inherit: heredado.

MODELO DE CAJAS

Z-Index: establece la posición de la caja respecto al eje z, es decir, la profundidad. Valores: auto, n°entero, inherit. Muy útil para que cuando pase el ratón, esa imagen salga en primer plano.



CONTENIDO

- INTRODUCCIÓN
- PRIORIDAD
- SINTAXIS
- SELECTORES
- MODELO DE CAJAS
- **FORMATO PARA:**
 - **TEXTO**
 - LISTAS
 - TABLAS
 - LAYOUT
- MISCELÁNEA

FORMATO PARA: TEXTO

Propiedad: valor	Significado
color: <color> inherit	Color del texto
font-family: <fuente> inherit	Tipo de fuente de letra
font-size: <absoluto> <relativo> <%> inherit	Tamaño de letra
font-weight: normal bold bolder lighter 100 200 300 400 500 600 700 800 900 inherit	Grosor de letra
font-style: normal italic oblique inherit	Estilo de letra
font-variant: normal small-caps inherit	Mayúsculas en pequeño
text-align: left right center justify inherit	Alineación horizontal
line-height: normal <numero> <medida> <porcentaje> inherit	Interlineado
text-decoration: none (underline overline line-through blink) inherit	Decoración

text-transform: capitalize uppercase lowercase none inherit	Transformación
text-shadow: none h-shadow v-shadow blur color	Sombreado del texto
vertical-align: baseline sub super top text-top middle bottom text-bottom <porcentaje> <medida> inherit	Alineación vertical
text-indent: <medida> <porcentaje> inherit	Tabula las primeras líneas
letter-spacing: normal <medida> inherit	Espaciado entre letras
white-space: normal nowrap pre pre-line pre-wrap inherit	Tratamiento de los espacios en blanco
word-spacing: normal <medida> inherit	Espaciado entre palabras

CONTENIDO

- INTRODUCCIÓN
- PRIORIDAD
- SINTAXIS
- SELECTORES
- MODELO DE CAJAS
- **FORMATO PARA:**
 - TEXTO
 - **LISTAS**
 - TABLAS
 - LAYOUT
- MISCELÁNEA

FORMATO PARA: LISTAS

Propiedad: valor	Significado
<code>list-style-type: disc circle square decimal decimal-leading-zero lower-roman upper-roman lower-greek lower-latin upper-latin armenian georgian lower-alpha upper-alpha none inherit</code>	Viñeta usada para los elementos de la lista
<code>list-style-position: inside outside inherit</code>	Posición de la viñeta
<code>list-style-image: <url> none inherit</code>	Sustituye la viñeta por una imagen
<code>List-style: (propiedad shorthand)</code>	Combina las anteriores

CONTENIDO

- INTRODUCCIÓN
- PRIORIDAD
- SINTAXIS
- SELECTORES
- MODELO DE CAJAS
- **FORMATO PARA:**
 - TEXTO
 - LISTAS
 - **TABLAS**
 - LAYOUT
- MISCELÁNEA

FORMATO PARA: TABLAS

Propiedad: valor	Significado
<code>border-collapse: collapse separate inherit</code>	Determina la fusión de bordes
<code>border-spacing: <medida> <medida> inherit</code>	Separación de bordes horizontal y vertical
<code>empty-cells: show hide inherit</code>	Celdas vacías
<code>caption-side: top bottom inherit</code>	Posición del título

CONTENIDO

- INTRODUCCIÓN
- PRIORIDAD
- SINTAXIS
- SELECTORES
- MODELO DE CAJAS
- **FORMATO PARA:**
 - TEXTO
 - LISTAS
 - TABLAS
 - **LAYOUT**
- MISCELÁNEA

FORMATO PARA: LAYOUT

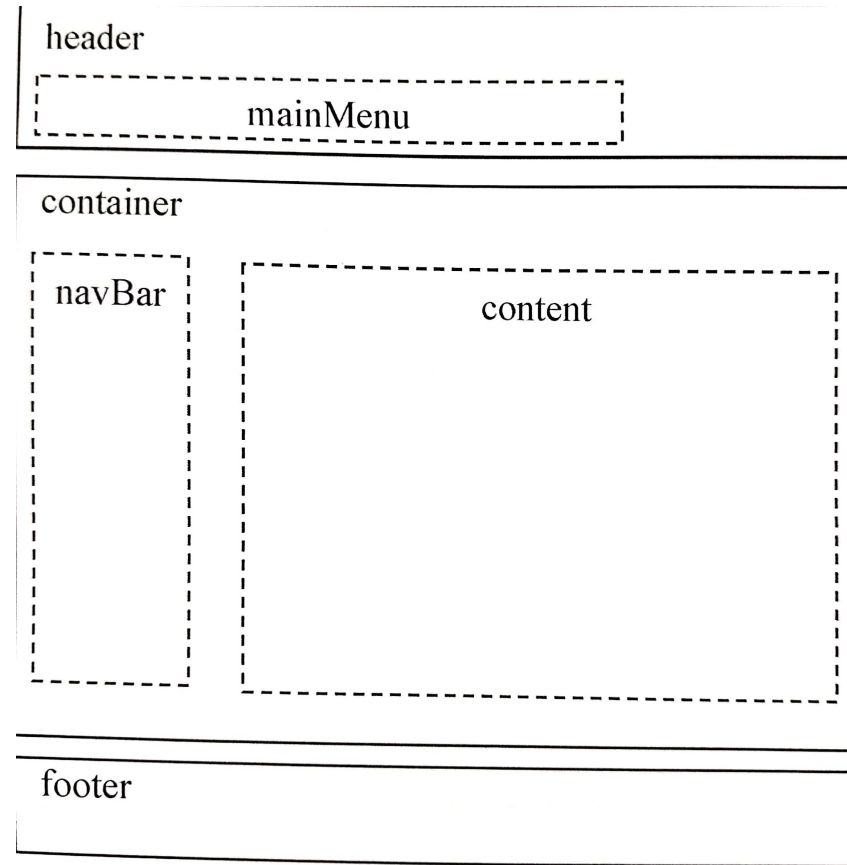


Figura 3.31: Capas usadas en un diseño a 2 columnas

CONTENIDO

- INTRODUCCIÓN
- PRIORIDAD
- SINTAXIS
- SELECTORES
- MODELO DE CAJAS
- FORMATO PARA:
 - TEXTO
 - LISTAS
 - TABLAS
 - LAYOUT
- **MISCELÁNEA**

MISCELÁNEA

Lo correcto para escribir CSS, es hacerlo en varias hojas de estilos independientes, y dentro de cada hoja, agruparlas en base a la función que realizan. Por ejemplo, estilos globales (html, body), estilos de layout, estilos para cada una de las capas, etc.

Webs interesantes:

browsershows.org: visualiza nuestro sitio en varios navegadores sin necesidad de instalarlo.

www.csszengarden.com: diseño mediante CSS.

www.openwebdesing.org: plantillas CSS e imágenes gratuitas.

css.maxdesing.com.au: tutoriales en línea para el aprendizaje de CSS.