

## Diseño Web — Tecnologías Involucradas

### Capa de Presentación CSS

info@covetel.com.ve<sup>1</sup>

<sup>1</sup>Cooperativa Venezolana de Tecnologías Libres R.S.

## Principios básicos de las hojas de estilo en cascada

Principios

Conceptos fundamentales

- ▶ Mayor control de la tipografía del diseño de la página.
- ▶ Menos trabajo
- ▶ Documentos potencialmente más pequeños
- ▶ Documentos potencialmente más accesibles
- ▶ HTML de presentación ya esta en vía de desaparecer
- ▶ Tiene buen soporte

## Como funciona CSS

- ▶ Se crea el documento XHTML o HTML
- ▶ Se escriben las reglas de estilo para definir el aspecto
- ▶ Se vinculan los estilos al documento.

## Hojas de estilo incrustadas <style> </style>

Un método más compacto para agregar hojas de estilo, es incrustar un bloque de estilo en la parte superior del documento HTML.

### Atributos no comunes

- ▶  
media='all|aural|braille|handheld|print|projection|screen|tty|tv'
- ▶ title='text'
- ▶ type='tipo de contenido' (*requerido*)

```
1 <html>
2 <head>
3 <style type="text/css">
4 h1 {color:red;}
5 p {color:blue;}
6 </style>
7 </head>
8 <body>
9 <h1>Header 1</h1>
10 <p>A paragraph.</p>
11 </body>
12 </html>
```

## Hojas de estilo externas

El método mas eficiente de utilizar CSS es reunir todas las reglas de estilo en un documento de texto independiente y crear vínculos a ese documento desde todas las páginas de un sitio.

Este modo permite hacer cambios de estilo homogéneos en todo el sitio editando un solo documento.

El documento de hoja de estilo es un documento de texto con al menos una regla de estilo.

El documento de hoja de estilo puede contener comentarios, como por ejemplo:

```
1  /* Esto es un comentario */
```

## Utilizar un Link

El método con mejor soporte para hacer referencia a hojas de estilo es crear un vínculo al documento CSS utilizando el elemento `link` en la cabecera `head` del documento.

```
1 <head>  
2   <link rel="stylesheet" type="text/css" href="theme.css" />  
3 </head>
```

El atributo `rel` define la relación del documento externo con el documento actual, el atributo `href` indica la URL del documento de hoja de estilo. Los autores pueden vincular más de una hoja de estilo al documento.

## Importar

Una alternativa a los vínculos es importar una hoja de estilo externa a un documento utilizando la función `@import` en el elemento `style`

```
1 <style type="text/css">
2   <!--
3     @import url(http://www.example.com/stylesheet.css);
4   -->
5 </style>
```



## Conceptos fundamentales

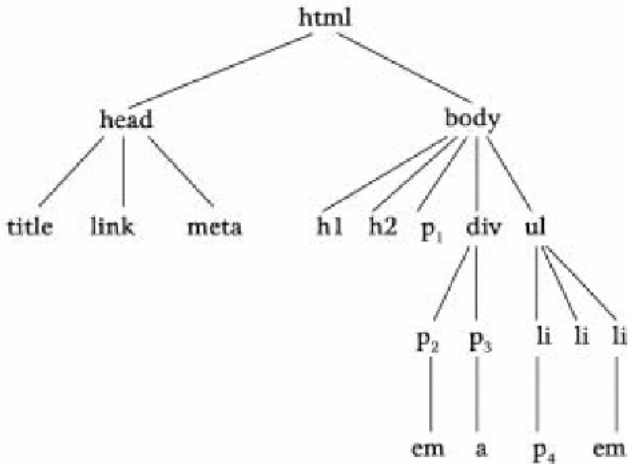
Para familiarizarse con el comportamiento de CSS es importante comprender sus conceptos claves.

- ▶ Estructura y herencia documentales
- ▶ Reglas de estilo en conflicto: la "cascada"
- ▶ Tipos de elementos
- ▶ El modelo de cajas.

## Estructura y herencia documentales

Los documentos XML, XHTML y HTML tienen una jerarquía implícita. Por ejemplo, el documento raíz `html` suele contener un `head` y un `body` y el `body` a su vez contiene distintos elementos a nivel de bloque como párrafos `p`. Esta jerarquía puede visualizarse como un árbol, separándose en ramas desde la raíz. La siguiente figura muestra la estructura de un documento XHTML muy sencillo.

## Estructura y herencia documentales



## La relación padre-hijo

La estructura del documento se convierte en un árbol genealógico cuando se hace referencia a las relaciones entre elementos. Un elemento que esta contemplado directamente por otro elemento es el **hijo** de éste. En la figura anterior el elemento p1 es hijo de body y body es su **padre**. Los elementos que tienen el mismo padre, son **hermanos**. En el ejemplo el elemento l1 es el hijo de u1 y el resto de los elementos l1 son sus hermanos.

La relación **padre-hijo** es fundamental para el funcionamiento de CSS

## Herencia

En la relación con las relaciones estructurales está el concepto de herencia, por la cual la mayoría de los estilos pasan de un elemento a sus descendientes. Esto indica en otras palabras que un hijo puede heredar propiedades de su padre.

Por ejemplo, si una regla de estilo aplica un color para un elemento `u1`, también aplica para para todos sus hijos `li`.

En CSS la mayoría de las propiedades se heredan, pero algunas como márgenes y fondos no.

## Reglas de estilo en conflicto: la cascada

Es posible y común que los elementos de un documento tengan instrucciones de presentación de distintas fuentes. Es posible entonces, que surjan conflictos. El grupo de trabajo que desarrolló CSS ya anticipó esta situación y concibió un sistema jerárquico que asigna distintos pesos a distintas fuentes de información de estilo.

La **cascada** de hojas de estilo en cascada se refiere a lo que ocurre si varias fuentes de información de estilo concurren por el control de los elementos de una página; la información de estilo se transmite hasta que es ignorada por un comando de estilo con más peso.

La cascada ordena proporcionar un conjunto de reglas para resolver conflictos entre hojas de estilo concurrentes. Cuando un agente de usuario encuentra un elemento mira a todas las declaraciones de estilo que pueda tener aplicadas y las ordena de acuerdo al origen de la hoja de estilo, a la especificidad de los selectores y al orden de la regla para determinar cuál aplicar.

## Origen de la hoja de estilo

Los navegadores dan distinto peso a las hojas de estilo de estas fuentes, listadas de menor a mayor peso:

- ▶ Hojas de estilo del agente de usuario
- ▶ Hojas de estilo del lector
- ▶ Hojas de estilo del autor
- ▶ Declaraciones de estilo !important del lector.

## Jerarquía de pesos adicionales I

Tras considerar la fuente de la hoja de estilo hay otra jerarquía de pesos que se aplica a las hojas de estilo creadas por el autor del documento.

Los puntos de origen en las hojas de estilo del autor tienen también distintos pesos (recuerde que todos los estilos del autor ignoran los estilos del agente de usuario y del lector a no ser que el lector marque el estilo como `!important`)

Esta lista señala el peso de las distintas declaraciones de estilo del autor, de menor a mayor peso. En otras palabras, las reglas de estilo que están al final de la lista ignoran a las primeras.

1. **Hojas de estilo externas vinculadas (utilizando el atributo `link`):** Si hay varias hojas de estilo vinculadas, las reglas de estilo de las hojas de estilo listadas por debajo en el documento tendrán preferencia sobre las listadas por encima. Por ejemplo, si un documento HTML vincula a dos hojas de estilo, de este modo:

```
1 <head>
2 <link rel="stylesheet" href="style1.css" type="text/css" />
3 <link rel="stylesheet" href="style2.css" type="text/css" />
4 </head>
```



## Jerarquía de pesos adicionales II

Si una regla de estilo indicada en `style2.css` esta en conflicto con una regla de estilo de `style1.css`, la regla de `style2.css` tendrá preferencia porque la hoja de estilo está listada por debajo en el documento fuente.

2. **Hojas de estilo externas importadas (utilizando `@import`):** La información de estilo importada ignora los estilos vinculados en el header. Si hay varias directivas `@import` las reglas indicadas en las hojas de estilo que estén por debajo en la lista ignoran las que están por encima.
3. **Hojas de estilo incrustadas (con el elemento `<style>`):** Los estilos aplicados a un documento determinado ignoran las reglas establecidas externamente.
4. **Estilos en línea (utilizando el atributo `style=` en una etiqueta de elemento):** Los estilos en línea ignoran todas las demás declaraciones de estilo que puedan hacer referencia a ese elemento, con una excepción:
5. **Declaraciones de estilo marcadas como `!important`:** Cualquier estilo marcado como `!important` ignora todas las reglas de estilo en conflicto. Lo único que puede ignorar una regla importante de una hoja de estilo de autor es una regla importante creada por el usuario.

## Jerarquía de pesos adicionales III

Ejemplo utilizando la directiva `!important`:

```
1 p {color: blue !important; }
```

En la siguiente gráfica podemos ver el orden de prioridades.



## Especificidad del selector I

La cascada continua a nivel de reglas, en el siguiente ejemplo, hay dos reglas que hacen referencia al elemento `strong`

```
1 strong {color: red;}  
2 h1 strong {color: blue;}
```

El agente de usuario asigna distintos tipos de niveles de peso a los distintos tipos de selectores. Cuanto más específico sea el selector más peso se le dará para ignorar las las declaraciones en conflicto. Para el ejemplo anterior todo el texto `strong` del documento se renderizará en rojo. Sin embargo, si el texto `strong` aparece dentro de una cabecera `h1` se renderizará en azul porque un elemento de un contexto determinado es más específico y tiene más peso. Esta es una lista de tipos de selector en orden de peso de menor a mayor.

1. Selectores de elementos y pseudoelementos específicos (por ejemplo `p` o `first-letter`)
2. Selectores contextuales (p.e. `h1 strong`)
3. Selectores de clase (p.e. `p.special`)

## Especificidad del selector II

### 4. Selectores ID (p.e. p#intro)

Recuerde que todas las reglas marcadas como `!important` ignorarán reglas en conflicto sea cual sea su especificidad u orden.

## Orden de las reglas I

Por último, cuando los estilos se han ordenado por autor, método de vinculación y especificidad, puede seguir habiendo conflictos dentro de una única fuente de hoja de estilo. Cuando una hoja de estilo contiene varias reglas en conflicto de igual peso, la que esté de última en lugar tiene más peso e ignora al resto. Por ejemplo, en el siguiente ejemplo, todas las cabeceras de primer nivel del documento serían rojas porque se impone la última regla.

```
1 h1 {color: green;}  
2 h1 {color: blue;}  
3 h1 {color: red;}
```

Ya se habló de este principio de **el último gana** en relación a varios elementos `link` y comandos `@import`. También se aplica en un solo bloque de declaración. Por ejemplo.

```
1 div#side {border-color: gray;  
2     border-color-top: black; }
```

## Elementos de bloque y en línea I

En XHTML, la distinción entre los elementos a nivel de bloque y en línea se basa en reglas de contención o, en otras palabras, en función del anidamiento de elementos en otros documentos. En general, los elementos a nivel de bloque pueden contener elementos en línea y elementos de bloque, mientras que los elementos en línea solo pueden contener texto y otros elementos en línea.

Sin embargo, algunos de estos elementos de bloque deben cumplir reglas especiales en XHTML, como son los párrafos, las cabeceras, y direcciones (address) solo pueden contener elementos en línea y contenido.

En CSS, sin embargo, la noción de **nivel de bloque** y **en línea** es puramente de presentación visual. `block` e `inline` son dos posibles roles de maestra en pantalla utilizados para indicar a los agentes de usuario cómo presentar el documento en el diseño.

## Elementos de bloque y en línea II

Un elemento a nivel de bloque de CSS (`display: block`) siempre genera saltos antes y después de él. Llena el ancho disponible del elemento padre que lo contiene sea el ancho del cuerpo del documento o un espacio menor definido como el de un `div`. No puede emplazar nada junto a un elemento de bloque en el flujo normal del documento.

Los elementos en línea CSS (`display: inline`) no generan saltos de línea. Aparecen en el flujo de la línea y sólo pasarán a otra línea si no tienen espacio.

A diferencia de las nociones de XHTML de bloque y en línea, un elemento a nivel de bloque CSS puede estar anidado en un elemento en línea y viceversa. Al utilizar CSS cualquier elemento XHTML (o XML) puede convertirse a nivel de bloque o en línea.

## Introducción al modelo de cajas I

El modelo de cajas constituye la piedra angular del sistema de formateo visual de CSS. Es un concepto fundamental para comprender el funcionamiento de las hojas de estilo.

De acuerdo al modelo de cajas todos los elementos, a nivel de bloque o en línea, generan una caja rectangular alrededor llamada **caja de elemento** (aunque las cajas de bloque y de línea se tratan ligeramente de manera distinta).

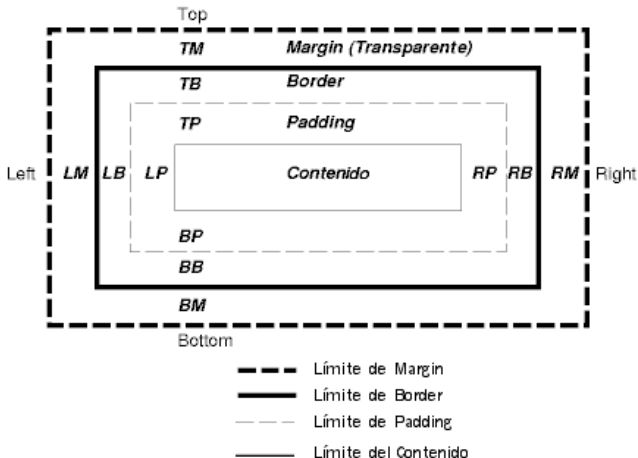
Pueden aplicarse propiedades como bordes, márgenes y fondos (entre otras) a la caja de un elemento. Las cajas también pueden utilizarse para posicionar elementos y diseñar la página.

Las cajas de elementos están hechas de cuatro componentes principales. En el núcleo de la caja está el contenido del elemento. El contenido está rodeado por cierta cantidad de relleno, sigue el borde rodeado por el margen como puede verse en las figuras 1 y 2.



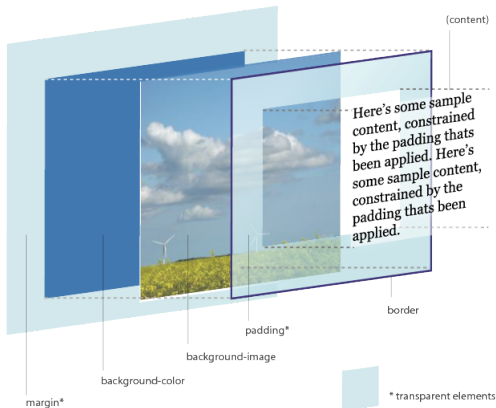
## Introducción al modelo de cajas II

Figura: Modelo de Cajas



## Introducción al modelo de cajas III

**Figura:** Modelo de Cajas 3D



## Introducción al modelo de cajas IV

**Características fundamentales del modelo de caja que merece la pena destacar:**

1. Relleno, bordes y márgenes son opcionales. Si ajusta a cero sus valores se eliminan de la caja.
2. El área de relleno es el espacio entre el borde del área de contenido y el borde (si lo hay). Cualquier color o imagen de fondo aplicados al elemento se extenderá por toda el área de relleno.
3. Los bordes se generan con propiedades de estilo, que especifican su estilo, ancho y color. Cuando un borde tiene huecos, el color o imagen de fondo aparece a través de esos huecos. En otras palabras, los fondos se extienden tras el borde hasta el borde exterior.
4. Los márgenes siempre son transparentes, lo que significa que el color de fondo o el patrón del elemento padre se mostrará a través de ellos. El límite del margen (el borde exterior del elemento) no es visible, pero es una cantidad calculada.

## Introducción al modelo de cajas V

5. El ancho de un documento solo se aplica al ancho del área de contenido. Esto significa que si especifica que un elemento debería tener una anchura de 200px, los contenidos reales se mostrarían con una anchura de 200px, y los anchos acumulativos del relleno, el borde y los márgenes, se sumarían a esa cantidad.
6. Se puede cambiar el estilo de los lados superior, derecho, inferior, e izquierdo de una caja de elemento por separado. Por ejemplo, puede añadir un borde a la parte inferior de un elemento o solo a los lados derecho e izquierdo.

---

<sup>1</sup>Esta imagen se utiliza con autorización de

<http://www.sidar.org/recur/desdi/traduc/es/css/box.html>

<sup>2</sup>Esta imagen se utiliza con autorización de <http://www.hicksdesign.co.uk/boxmodel/>