

# Hojas de estilo en cascada (CSS)

---

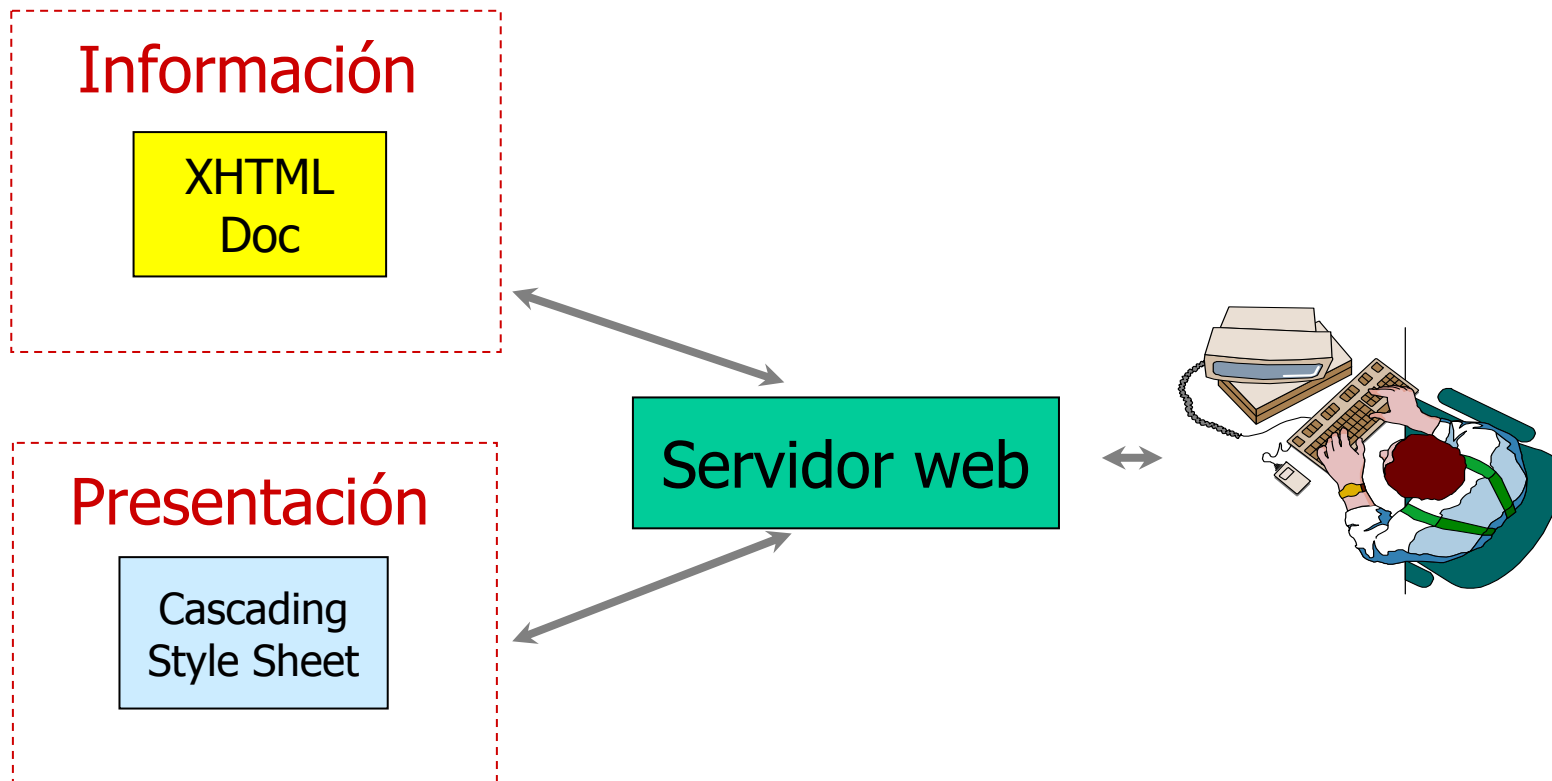
Tecnologías Web



- ❑ El objetivo fundamental de las hojas de estilo CSS es separar el contenido de la apariencia
- ❑ La información de nuestros sitios WEB tienen que estar almacenadas en páginas (X)HTML sin incluir ningún tipo de información a cerca de su estilo
- ❑ Permite administrar el estilo de sitios WEB que sean grandes y que requieran ser sofisticados.
  - ❑ Creando las hojas CSS una vez, podemos compartirlas entre varios desarrolladores WEB.
  - ❑ Si necesitamos realizar un cambio de estilo simplemente tenemos que hacerlo en la hoja CSS pertinente.

- ❑ Las hojas de estilo en cascada es una forma de asociar información de presentación a documentos HTML y XML.
- ❑ Permite definir estilos para la presentación de los distintos elementos de un documento
- ❑ Los estilos normalmente se almacenan en una hoja de estilo (un fichero de texto con la extensión .css)
- ❑ Tipos de hojas de estilo CSS
  - ❑ Recomendación Cascading Style Sheets 1
    - ❑ <http://www.w3.org/TR/CSS1>
  - ❑ Recomendación Cascading Style Sheets 2 (actualmente 2.1)
    - ❑ <http://www.w3.org/TR/CSS21>
  - ❑ Draft Cascading Style Sheets 3 (en desarrollo)
- ❑ Algunos navegadores tienen extensiones propietarias
  - ❑ IEExplorer → Permite definir mediante una hoja CSS el color de las barras de "scroll".

- Cuando asociamos una hoja CSS a una página WEB "sugerimos" al navegador como debe presentar dicha página.



# ¿Cómo son las hojas de estilo CSS?

Documento .css

```
body{
  font-family: Verdana;
  font-size: 1em;
  text-align: justify}

h1{
  font-family: Verdana, sans-serif;
  font-size: 1.3em}

code{
  font-family: Courier, sans-serif;
  font-size: 1em
}

.note{
  background-color: #003333;
  border-width: thin;
  border-color: black;
  border-style: ridge;
  color: white;
  font-family: Geneva;
  font-size: .9em;
  vertical-align: text-bottom
}
```

Hoja de estilos

Regla

Regla

Regla

Regla

Selector

Atributos

Regla

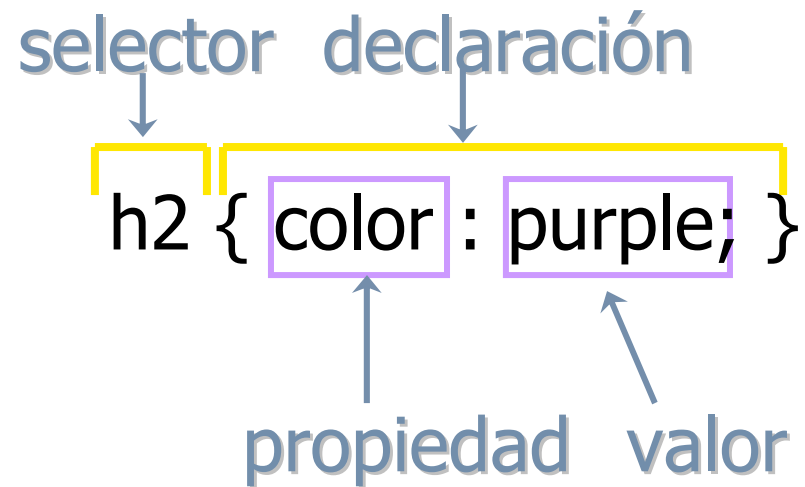
Regla

## Hojas de estilo en ¿cascada?

- ❑ La definición de los estilos para un elemento se puede realizar:
  - ❑ Dentro del propio elemento (internas al elemento)
  - ❑ Con una hoja de estilo interna
  - ❑ Con una hoja de estilo externa
  - ❑ Comportamiento por defecto del visualizador (p.e. IExplorer)
- ❑ Todas las definiciones se combinan en *cascada* para producir una única hoja de estilo
  - ❑ Las definiciones mas locales tienen prioridad

## □ Sintaxis

- selector {propiedad: valor}
- Normalmente el selector es el elemento al que se aplica el estilo



- ❑ A menos que haya una regla mas especifica, los elementos comprendidos dentro de una etiqueta, heredan el estilo definido para esta etiqueta
- ❑ Regla
  - ❑ `h1 { color: red; }`
- ❑ Fragmento de documento
  - ❑ `<h1>un documento con titulo con palabra <em>enfaticada</em><h1>`
- ❑ La palabra enfaticada aparecerá enfaticada y en rojo
- ❑ Si existe una regla del tipo `em { color: blue; }` entonces se mostrara en azul ya que es una regla mas especifica que prevalece sobre la herencia



- ❑ Algunas propiedades no son heredadas
  - ❑ E.j propiedad `background`
- ❑ Valores de atributos porcentuales
  - ❑ Para los valores de algunas propiedades se admite un valor en forma de porcentaje.

```
p { font-size: 10pt }  
/* ' line-height' Hace referencia a 'font-size', i.e. 12pt */  
p { line-height: 120% }
```

- ❑ En estos casos se hereda el "valor efectivo" (el valor calculado) para dicha propiedad.

- ❑ Los tipos son
  - ❑ Bloque
  - ❑ En línea
  - ❑ Listas
  
- ❑ Cada etiqueta HTML pertenece a uno de estos tipos y cada tipo de etiqueta se mostrará de una manera distinta
  - ❑ Bloque: El contenido está separado del resto de elementos que lo rodean, normalmente se deja un margen de espacio encima y debajo de la etiqueta.
  - ❑ En línea: Los elementos no son separados del resto de elementos que lo rodean.
  - ❑ Listas: Se comportan de manera similar al tipo bloque, pero además se muestra una marca a su izquierda, para identificar cada elemento de la lista
  
- ❑ El tipo al que asociamos la etiqueta se controla con el atributo display
  - ❑ `display : box` → Convierte la caja en tipo bloque
  - ❑ `display: inline` → Convierte la caja en tipo en línea
  - ❑ `display: list-item` → Convierte la caja en tipo lista

- ❑ Una etiqueta

```
p {text-align: center; color: red}
```

- ❑ Se pueden agrupar varias etiquetas a las que se les aplica el mismo estilo

```
h1, h2, h3, h4, h5, h6 { color: green }
```

- ❑ Clase

- ❑ ID

- ❑ Pseudo clase

- ❑ Pseudo elemento

- ❑ Contextual

## Clase como selector de regla

- ❑ A una etiqueta HTML se le puede asignar una o varias clases

```
<h2 class="cabecera2">
```

- ❑ Cuando se especifican varias clases van separadas por un espacio en blanco

```
<etiqueta class="clase1 clase2">
```

- ❑ La clase especifica un selector particular en la hoja de estilo

```
h2.cabecera2 { text-align: center; }
```

- ❑ Un especificador de clase puede aplicarse a diversos elementos

```
.cabecera2 { text-align: center; }
```

```
<h2 class="cabecera2">...</h2>  
...  
<h1 class="cabecera2">...</h1>
```

## ID Como selector de regla

- ❑ La regla se aplica al contenido HTML en función de su identificador "id"

```
#identificadorUnico { color : silver; }
```

- ❑ Documento HTML

```
<h1 id="identificadorUnico" >Este texto aparecera en color plateado.</h1>
```

- ❑ Usando este tipo de selector se puede establecer el etiquetas particulares de nuestro documento.
- ❑ El objetivo de las hojas de estilo CSS es que la apariencia sea homogénea con lo que hay que usar este tipo de selectores con mesura.

## Pseudo clases como selectores de reglas

- ❑ Son clases "virtuales" que nosotros no insertamos de manera explícita en nuestras páginas
  - ❑ El navegador dependiendo de ciertas situaciones "crea" estas clases virtuales.
  
- ❑ Enlaces
  - ❑ `a:link`
    - ❑ Enlace no visitado o activado por el usuario
  - ❑ `a:hover`
    - ❑ Cuando se coloca el raton sobre el enlace
  - ❑ `a:active`
    - ❑ Cuando se hace click sobre el enlace
  - ❑ `a:visited`
    - ❑ Enlace ya visitado por el usuario

## Pseudo clases como selectores de reglas (cont.)

- ❑ Los selectores de pseudo clases con "encajan" con las clases que incluimos dentro del atributo "class"
  - ❑ El estilo no tiene efecto sobre el enlace

```
a:link { color: red }  
<a class="link"> ... </a>c
```

- ❑ Las pseudo clases también pueden combinarse con otras clases

```
a.external:visited { color: blue }  
<a class="external" href="http://out.side/">external link</a>
```

- ❑ También pueden usarse con selectores contextuales

```
a:link img { border: solid blue }
```

# Pseudo elementos como selectores de reglas

- ❑ Algunos elementos tipográficos no están asociados con la estructura sino como se presentará dicha información.
- ❑ `:first-letter`
  - ❑ Primera letra en un elemento de bloque como `<p>` o `<h1>`
- ❑ `:first-line`
  - ❑ Primera línea en un elemento de bloque como `<p>` o `<h1>`

## Ejemplo de pseudo-elemento

```
<html>
  <head>
    <title>Probando pseudo elementos</title>
    <style type="text/css">
      p:first-line { font-variant: small-caps; font-weight: bold };
      p:first-letter { font-size: 200%;}
    </style>
  </head>
  <body>
    <p>Esta es una línea de prueba</p>
  </body>
</html>
```



## Pseudo elementos como selectores de reglas (cont)

- ❑ Cuando usamos selectores contextuales, los pseudo elementos solo pueden aparecer al final

```
body p:first-letter { color: purple }
```

- ❑ Pueden combinarse con selectores de clase

```
p.inicial:first-letter { color: red }  
<p class="inicial">Primer párrafo</A>
```

**No todos los atributos de las CSS afectarán a las reglas para pseudo elementos.**  
**<http://www.w3.org/TR/CSS1#typographical-pseudo-elements>**

- ❑ Simplifican el tratamiento de las "excepciones"

## Ejemplo

```
<html>
  <head><title>Selectores contextuales</title>
  <style type="text/css">
    h1 { color: blue }
    em { color: red }
  </style>
  <body>
    <h1>Titulo con palabra <em>enfaticada</em></h1>
    <p>Parrafo en el que necesitamos <em>enfaticar</em>
    parte del mismo</p>
  </body>
</html>
```

- ❑ Solución del problema usando selectores contextuales

```
<style type="text/css">
  h1 { color: blue }
  h1 em { color: red }
</style>
```

# Como asociar una hoja de estilos a un fichero XHTML

## ❑ Dentro del propio elemento (no aconsejable)

```
<p style="color: green; margin-left: 20px">esto es un párrafo </p>
```

## ❑ Hoja de estilo interna

- ❑ Definición de los estilos dentro de la sección <head> del propio documento marcado

```
<head>
  <style type="text/css">
    hr {color: blue};
    p {margin-left: 20px}
    body {background-image: url("fondo.jpg")}
  </style>
</head>
```

## ❑ Hoja de estilo externa

- ❑ Fichero de texto (.css) con todas las definiciones de estilo

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

- <http://www.w3schools.com/css>
- [http://www.devguru.com/Technologies/css/quickref/css\\_intro.html](http://www.devguru.com/Technologies/css/quickref/css_intro.html)
- <http://www.brainjar.com/css/positioning/default.asp>
- <http://www.htmlhelp.com/reference/css/>
- [http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/dhtml\\_node\\_entry.asp](http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/dhtml_node_entry.asp)
- [http://www.westciv.com/style\\_master/academy/browser\\_support/index.html](http://www.westciv.com/style_master/academy/browser_support/index.html)
- <http://www.w3.org/TR/CSS1>
- <http://www.w3.org/TR/CSS21>

## Las etiquetas <div> y <span>

- ❑ Son elementos genéricos que no tienen por sí mismos ningún significado que afecte a la presentación.
  - ❑ El navegador no presenta su contenido de ninguna manera especial.
- ❑ La etiqueta <div> sirve para organizar el contenido a nivel de bloque
- ❑ La etiqueta <span> sirve para organizar el contenido a nivel de línea
- ❑ Los elementos <div> y <span> se usan mucho para organizar la presentación de los documentos HTML mediante el uso de selectores de clase.

### Ejemplo de uso de <div> y <span>

```
div{ font-family: Comic Sans MS; font-size: 14 pt }
span.cursiva { font-style: oblique }

<div>Este es un elemento de <span class="cursiva">nivel de
bloque</span>.</div>
```

## ❑ Motivación

- ❑ Permitir modularidad en las hojas de estilo
- ❑ Equilibrio entre el autor del documento y el lector del mismo
  - ❑ Se da preferencia al autor sobre el lector CSS 1.0

## ❑ Regla @import

- ❑ Permite importar una hoja .css dentro de otra hoja de estilos
- ❑ Syntaxis: @import url( **http://host/path\_to\_css** )

```
<style type="text/css">
  @import url(http://style.com/basic);
  H1 { color: blue }
  ...
</style>
```

- ❑ Las hojas de estilo importadas, también se ven afectadas por las reglas de la cascada.
- ❑ Las reglas @import deben ir al comienzo de la hoja de estilo, antes de definir ninguna regla con atributos

- ❑ Resolución de conflictos
  - ❑ Si tenemos varias reglas que afectan a los mismos elementos, para resolver los conflictos asignamos un peso a dichas reglas → gana la que tiene mayor peso.
  
- ❑ Algoritmo de resolución del valor de una propiedad
  - ❑ Buscar todas las reglas que apliquen sobre una etiqueta, casos:
    - ❑ No existen reglas: Se usa el valor heredado y si no existe, se usa el valor por defecto.
    - ❑ Si existen reglas: Criterios de ordenación
      - ❑ Se asigna un peso según donde provenga la regla → Prioridad: navegador, usuario, autor (prioridad ascendente)
      - ❑ Se asigna mayor peso a las reglas marcadas con `!important`
      - ❑ Se asigna mayor peso a las reglas que tengan selectores sean más específicos.
      - ❑ Si dos reglas tienen el mismo peso, al última en definirse gana.
  - ❑ Nos quedamos con la regla de mayor peso

### ❑ Prioridad de selectores. Calcular:

- ❑ a = Número de atributos "id"
- ❑ b = Número de atributos "class" en el selector
- ❑ c = Número de etiquetas en el selector
- ❑ Concatenar estos 3 valores y ésta es la prioridad → "abc"

```
li           {...} /* a=0 b=0 c=1 -> specificity = 1 */
ul li       {...} /* a=0 b=0 c=2 -> specificity = 2 */
ul ol li    {...} /* a=0 b=0 c=3 -> specificity = 3 */
li.red      {...} /* a=0 b=1 c=1 -> specificity = 11 */
ul ol li.red {...} /* a=0 b=1 c=3 -> specificity = 13 */
#x34y      {...} /* a=1 b=0 c=0 -> specificity = 100 */
```

- ❑ Las reglas definidas en atributo style tienen la misma prioridad que reglas que usan selectores ID

### ❑ Declaraciones !important

- ❑ Como hemos visto, sirven para aumentar la prioridad de una regla sobre otra para asignar el valor de un atributo particular.
- ❑ Sintaxis

```
selector { atributo:valor ! important ; atributo: valor ; }
```



## ❑ Propiedades "resumidas"

### Propiedades de texto

```
h1 {  
  font-weight: bold;  
  font-size: 12pt;  
  line-height: 14pt;  
  font-family: Helvetica;  
  font-variant: normal;  
  font-style: normal;  
}
```

### Propiedad resumido

```
h1 { font: bold 12pt/14pt Helvetica }
```

- ❑ A cada propiedad se le puede asignar un valor dentro de su conjunto de valores válidos
  - ❑ Identificadores específicos: `blue`, `auto`, `Arial`, `center`
  - ❑ Números (enteros y reales): `10`, `12.1`
  - ❑ URLs: `url( http://www.bg.com/pin.gif )`

## Valores de las unidades de medida

- ❑ Se representa con un número (entero o real) seguido por la unidad de medida.
  - ❑ Ej: 10px, 12pt, 1.5mm
- ❑ Existen dos tipos de unidades de medidas
  - ❑ Absolutas
  - ❑ y relativas
- ❑ Absolutas
  - ❑ Son útiles sólo cuando se conocen propiedades del medio (i.e navegador)
  - ❑ Valores: in (pulgadas), cm, mm, pt (puntos: 1/72 de pulgada), pc (picas: 12 puntos)

### Ejemplo unidades absolutas

```
H1 { margin: 0.5in }      /* pulgadas, in = 2.54cm */
H2 { line-height: 3cm }  /* centímetros */
H3 { word-spacing: 4mm } /* milímetros */
H4 { font-size: 12pt }   /* puntos, 1pt = 1/72 in */
H4 { font-size: 1pc }    /* picas, 1pc = 12pt */
```

## Valores de las unidades de medida (cont.)

### ❑ Relativas

- ❑ Representan una longitud respecto a otra medida de longitud.
  - ❑ El valor del que se depende, es particular a cada propiedad
- ❑ Valores permitidos
  - ❑ em: Relativo al tamaño de fuente
  - ❑ ex: relativo a la altura de la letra x minúscula
  - ❑ px: relativo al tamaño de pixel

```
h1 { margin: 0.5em }      /* ems, the height of the element's font */
h1 { margin: 1ex }       /* x-height, ~ the height of the letter 'x' */
p  { font-size: 12px }   /* pixels, relative to canvas */
```

- ❑ Las unidades 'em' y 'ex' son relativas al tamaño de letra dentro del propio elemento, excepto para la propiedad "font-size" para la que se toma el tamaño de letra del padre.
- ❑ Las unidades relativas se heredan como valores calculados

```
body {
  font-size: 12pt;
  text-indent: 3em; /* i.e. 36pt */
  h1 { font-size: 15pt }
```

¿Para h1 text-indent vale: 36pt, 45pt?

## ❑ Porcentajes

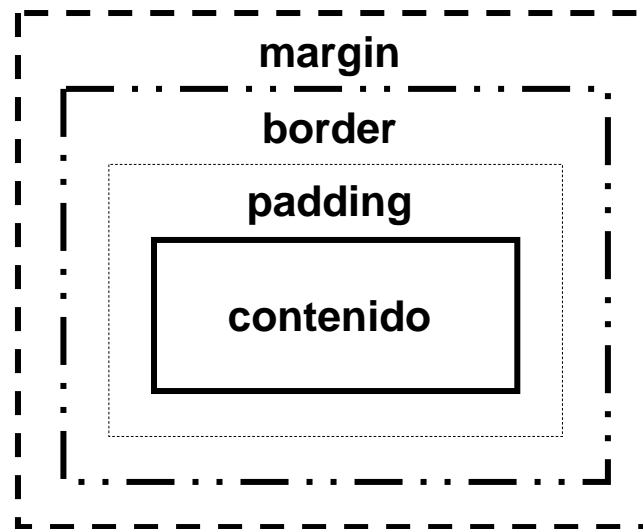
- ❑ Sintaxis: signo opcional, un número y un signo de porcentaje (%)
- ❑ Los porcentajes siempre son valores relativos a otro valor.
  - ❑ Cada propiedad establece el valor al que se refiere el porcentaje
- ❑ Los porcentajes se heredan como valores calculados

```
p { line-height: 120% } /* 120% del valor de 'font-size' */
```

## ❑ Colores

- ❑ Se pueden especificar mediante un identificador de color concreto
  - ❑ E.j: aqua, black, blue, fuchsia, gray
- ❑ Mediante una especificación RGB
  - ❑ Notación 1: #rrggbb
  - ❑ Notación 2: rgb ( r, g, b )
  - ❑ Notación 3: rgb ( r%, g%, b% )

- ❑ La recomendación CSS contempla un modelo de caja con el que cada elemento se visualiza en una caja rectangular



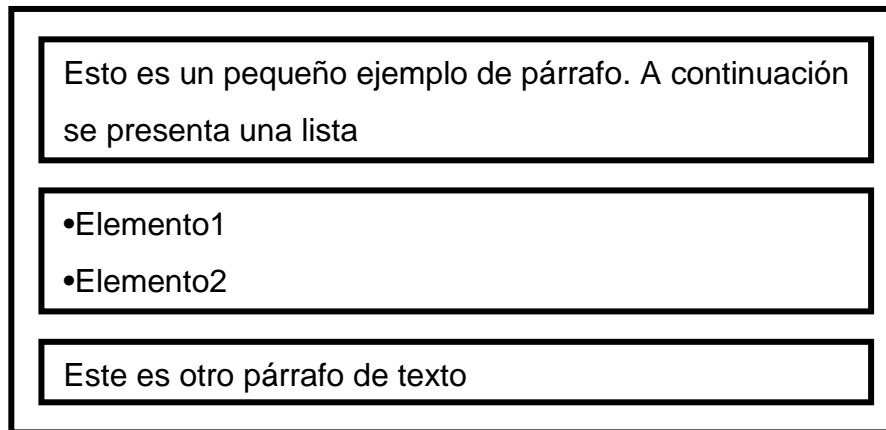
- ❑ El tamaño de cada área (*border*, *margin*, *padding*) se definen con propiedades relativas a las cuatro direcciones: *top*, *bottom*, *left*, *right*

```
margin: 10px /* Propiedad resumida */  
margin-top: 20px;  
margin-right: 20px;
```

- Para cada elemento se genera una caja
- Dentro de la caja de un elemento se encuentran las cajas de los otros elementos que contiene.
- El elemento raíz del documento genera una caja que sirve como bloque contenedor inicial para el resto de elementos
- La ubicación de una caja respecto de su bloque contenedor se puede realizar básicamente de dos formas: CSS 1.0
  - Normal
  - Elementos Flotante

## Flujo normal de Ubicación

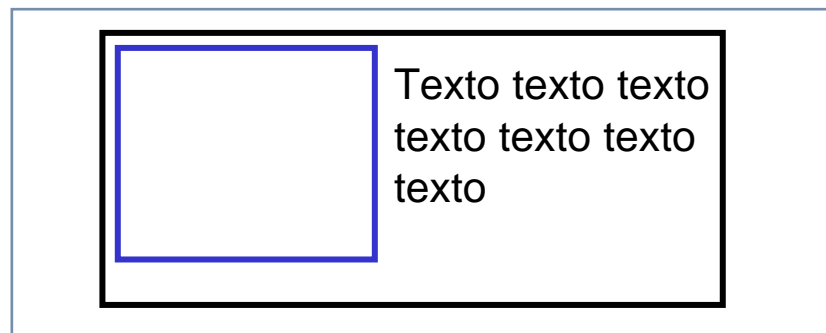
- ❑ Es el método que se aplica por defecto
- ❑ Consiste en colocar las cajas una tras otra verticalmente comenzando en la parte superior del bloque contenedor y siguiendo el orden en el que aparecen los elementos correspondientes en el documento.



## El modelo de flujo de elementos flotantes

- ❑ Una caja de tipo float se desplaza a la izquierda o a la derecha de la línea actual.
- ❑ La característica más interesante es que el resto de contenido puede colocarse a lo largo de su lado (o prohibirse).
- ❑ Si una caja flota hacia la izquierda (derecha), el contenido fluye hacia abajo en su parte derecha (izquierda).

Ejemplo float a la izquierda





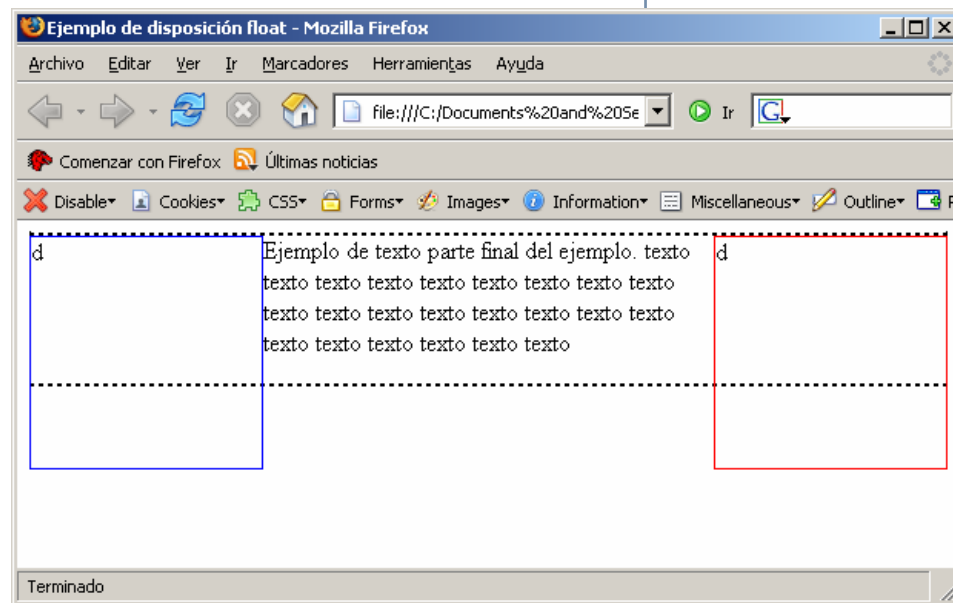
## Control de elementos flotantes

- ❑ Para controlar los elementos flotantes se utiliza la propiedad float
  - ❑ `float: left` → La caja flota a la izquierda
  - ❑ `float: right` → La caja flota a la derecha
  - ❑ `float: none` → La caja no flota
  
- ❑ También podemos controlar como se comportarán las cajas que están cerca de una o varias cajas que flotan
  - ❑ Podemos controlar si el borde de una caja puede estar cerca del borde de una caja flotante
  - ❑ `clear: left` → La caja tiene que estar debajo de cualquier caja que flote a la izquierda que aparezca antes en el documento
  - ❑ `clear: right` → La caja tiene que estar debajo de cualquier caja que flote a la derecha que aparezca antes en el documento
  - ❑ `clear: both` → La caja tiene que estar debajo de cualquier caja que flote a que aparezca antes en el documento
  - ❑ `clear: none` → No hay ningún tipo de restricción

# Ejemplo de cajas float

## Ejemplo 2 cajas float

```
<html>
  <head>
    <title>Ejemplo de disposición float</title>
    <style type="text/css">
      #float1{
        float:left;
        width: 150px;
        height: 150px;
        border: 1px solid blue;
      }
      #float2{
        float:right;
        width: 150px;
        height: 150px;
        border: 1px solid red;
      }
      p {
        border: 1px dashed black;
      }
    </style>
  </head>
  <body>
    <p><div id="float1">d</div><div id="float2">d</div> texto ...</p>
  </body>
</html>
```





- ❑ Las recomendación CSS 2 es una extensión de las CSS1, siendo por tanto compatibles en "casi todos" los aspectos
- ❑ Cambios
  - ❑ Concepto de "tipos de dispositivos" ó tipos de medio
    - ❑ Ej: `screen`, `aural`, `print`
  - ❑ Más tipos de selectores
  - ❑ Más tipos de etiquetas
  - ❑ Más tipos de posicionamientos
  - ❑ z-index
  - ❑ ...

- ❑ Nos sirven para especificar como presentar como se presentará un documento en distintos tipos de dispositivo.
- ❑ Tipos de dispositivos

all	Todos los dispositivos
braille	Dispositivos braille táctiles
embossed	Dispositivos braille impresión
handheld	Dispositivos móviles
print	Impresión
Projection	Dispositivos de presentación
screen	Dispositivos de pantalla
speech	Sintetizadores de voz
tty	Terminales, teletipos
tv	Televisiones (baja resolución)

## Especificando tipos de medio en las CSS

### ❑ Dentro de una hoja CSS

```
@import url("estilosPantalla.css") screen;  
@media print {  
    /* Hoja de estilo para impresión */  
}
```

### ❑ Al referenciar una hoja de estilo externa

```
<link rel="stylesheet" type="text/css" media="print, handheld" href="estilos.css" />
```

### ❑ Podemos especificar varios tipos de medio en una regla @media

```
@media print {  
    body { font-size: 10pt }  
}  
@media screen {  
    body { font-size: 13px }  
}  
@media screen, print {  
    body { line-height: 1.2 }  
}
```

- ❑ Selectores de elementos hijo
  - ❑ Sintaxis: `etiquetaPadre>etiquetaHija { atributos }`
- ❑ Selectores de elementos hermanos
  - ❑ Sintaxis: `etiqueta1+etiqueta2 { atributos }`
- ❑ Estos selectores se pueden combinar con selectores contextuales

```
div ol>li p{  
  font-size: 12pt;  
  font-weight: bold;  
}
```

¿Qué etiquetas selecciona esta regla?

- ❑ Selectores por atributo
  - ❑ `[atributo]` → Encaja si se ha establecido (cualquier valor) de atributo
  - ❑ `[atrib=val]` → Encaja si se ha establecido el atributo `atrib` el valor `val`

# Más tipos de selectores

## Ejemplo de selectores

```
<html>
  <head>
    <title>Prueba de selectores hijo</p>
    <style type="text/css">
      body > P { color: red }
      p { color: green }
      div + p { color: blue }
    </style>
  </head>
  <body>
    <p>Parrafo 1</p>
    <p>Parrafo 2 </p>
    </p>
    <div><p>Parrafo dentro de etiqueta</p></div>
    <p>Párrafo 3</p>
  </body>
</html>
```

## Resultado en el navegador

**Parrafo 1**  
**Parrafo 2**  
Parrafo dentro de etiqueta  
**Párrafo 3**



## Resumen de selectores

Patrón	Significado	Ejemplo
*	Encaja con cualquier elemento	* { }
E	Encaja cualquier etiqueta E	p { }
E F	Encaja cualquier etiqueta F descendiente de E	div p { }
E > F	Encaja cualquier etiqueta F hija de E	body > h1 { }
E:first-child	Encaja la etiqueta E que sea el primer hijo de su padre	div:first-child { }
E:link E:visited	Encaja cualquier enlace que aún no ha sido visitado, o que ya lo ha sido	a:link{ }
E:active E:hover E:focus	Encaja cuando la etiqueta E está siendo activada, el curso pasar encima de ella o ha adquirido el foco	div:active{ }
E + F	Encaja cualquier etiqueta F que tiene como hermana a una etiqueta E que aparece justo antes	div + h1 { }
E[foo]	Encaja cualquier elemento E que se ha establecido el atributo foo	div[name]
E[foo="warning"]	Encaja cualquier etiqueta E que se ha establecido el atributo foo a warning	span[class="warning"]
DIV.warning	Cualquier elemento con valor warning en el atributo class	.titulo{ }
E#mi_id	Encaja con cualquier elemento con id mi_id	#float{ }

## Nuevos sistemas de posicionamiento

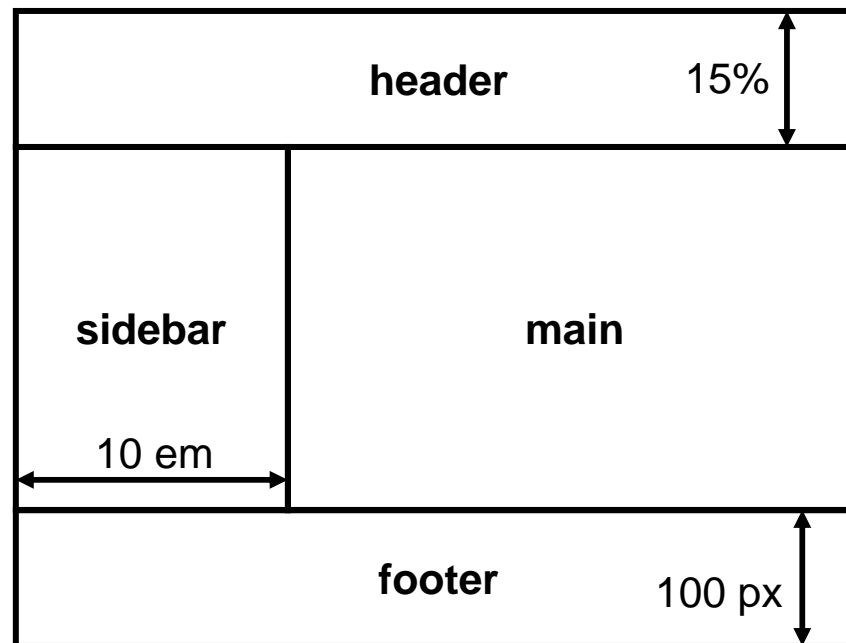
- ❑ `position: static`
  - ❑ Este es el modo de posicionamiento habitual
- ❑ `position: relative`
  - ❑ La caja se coloca con el posicionamiento habitual pero se desplaza según los atributos `left`, `right`, `top`, `bottom`
- ❑ `position: absolute`
  - ❑ La caja se coloca de manera absoluta con respecto a la caja que la contiene según los valores de `left`, `right`, `top`, `bottom`. La caja puede quedar oculta u ocultar otras cajas.
  - ❑ El atributo `z-index` permite controlar como se apilan las cajas. Las cajas que queden debajo se ocultarán
- ❑ `position: fixed`
  - ❑ Es un tipo de posicionamiento absoluto, pero se utiliza como referencia el “puerto de vista”.
  - ❑ Cuando utilizamos este modo de posicionamiento, la caja no se mueve cuando utilizamos las barras de desplazamiento.

## Ejemplo de posicionamiento fixed

```
<html>
  <head>
    <title>Ejemplo de disposición float</title>
    <style type="text/css" media="screen">
      body { height: 600 px; /* Necesario para los porcentajes */}
      #header{
        position: fixed;
        width: 100%; height: 15%;
        top:0; right: 0;
        bottom:auto; left: 0
      }
      #sidebar{
        position: fixed;
        width: 10em; height: auto;
        top:15%; right: auto;
        bottom:100px; left: 0;
      }
      #main{
        position: fixed;
        width: auto; height: auto;
        top:15%; right: 0;
        bottom:auto; left: 10em;
      }
      #footer{
        position: fixed;
        width: 100%; height: 100px;
        top:auto; right: 0;
        bottom:0; left: 0;
      }
    </style>
  </head>
</html>
```

## Ejemplo de posicionamiento

```
</style>
</head>
<body>
  <div id="header">...</div>
  <div id="sidebar">...</div>
  <div id="main">...</div>
  <div id="footer">...</div>
</body>
</html>
```



## Críticas, dudas y sugerencias...

Federico Peinado  
[www.federicopeinado.es](http://www.federicopeinado.es)