

Análisis y Diseño de Sistemas de Información para Internet



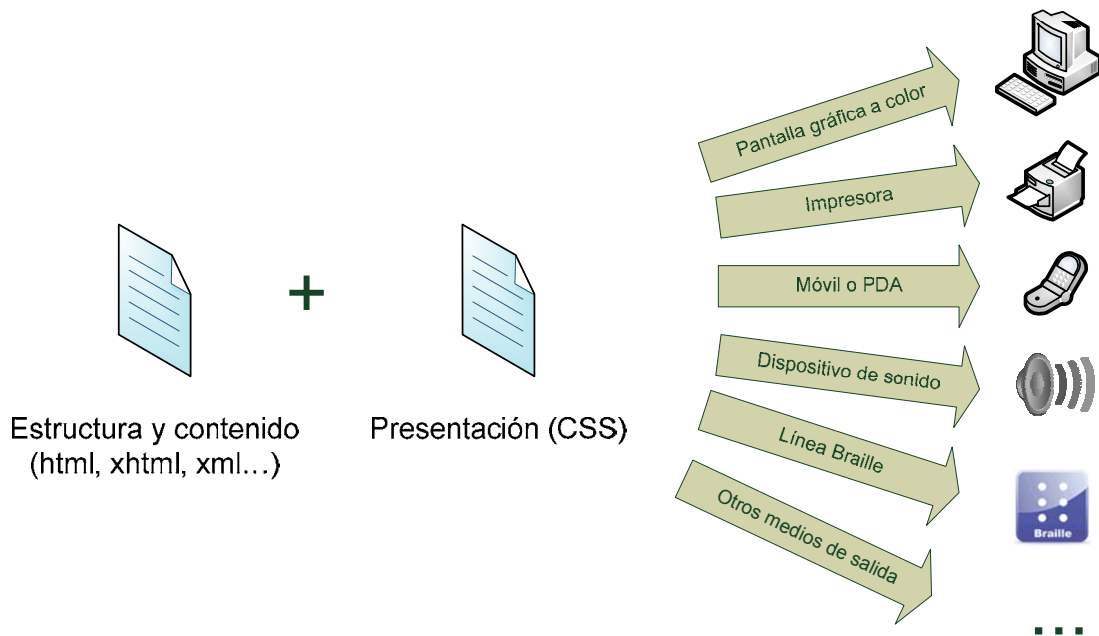
2. CSS

Luís Rodríguez Baena (luis.rodriguez@upsam.net)

Universidad Pontificia de Salamanca (campus Madrid)
Facultad de Informática

Introducción a CSS

- ❑ CSS (*Cascade Style Sheets*) es un estándar del W3C que define las reglas para la presentación del contenido de un documento html o xhtml.
 - Permite definir el aspecto de un elemento o un grupo de elementos dentro de un documento.



Introducción a CSS (II)

❑ Ventajas:

- Simplifica el código de las páginas web.
- Optimiza el rendimiento de los navegadores.
- Permite dar una apariencia homogénea a un sitio web al aplicar los mismos estilos a todas sus páginas.
- Amplía las posibilidades de presentación de HTML al permitir mucho más control.
- Permite presentar el mismo contenido de diferentes formas sin tener que modificarlo.
- Permite que los usuarios con necesidades especiales creen sus propias hojas de estilo para ver el contenido según sus preferencias.

❑ Desventajas:

- No todos los navegadores soportan por igual todas las características de CSS.

Introducción a CSS (III)

□ El estándar CSS.

- En 1996, el W3C aprueba la primera versión del estándar, conocida como CSS nivel 1.
- En mayo de 1998, se aprueba la segunda versión CSS nivel 2.
 - ✓ Las especificaciones están recogidas en www.w3.org/TR/1998/REC-CSS2-19980512/.
 - Existe una traducción al español en www.sidar.org/recur/desdi/traduc/es/css/cover.html.
- La recomendación actual del W3C es CSS nivel 2, revisión 1.
 - ✓ Presenta pocos cambios frente a la versión anterior.
 - Corrige algunas erratas de la especificación, elimina propiedades no implementadas por los navegadores, elimina propiedades no incluidas en CSS 3, etc.
 - ✓ Las especificaciones están recogidas en www.w3.org/TR/CSS21/.
 - Existe una traducción al español en www.w3.org/Style/css21-updates/css2.1_spa.pdf.
- Actualmente está bajo desarrollo CSS nivel 3.

Introducción a CSS

Modo de funcionamiento

- ❑ Cuando una aplicación de usuario procesa un documento fuente:
 - Analiza el documento y crea su estructura.
 - Identifica el tipo de medios (pantalla, impresora, sonido) al que va dirigido.
 - Recupera todas las hojas de estilo asociadas al documento que estén especificadas al tipo de medio.
 - Toma nota de cada elemento de la estructura, asignando un valor individual a cada propiedad según el tipo de medio.
 - ✓ Estos valores se asignan según los mecanismos de cascada y herencia.
 - A partir de la estructura del documento genera una estructura de formato.
 - Transfiere la estructura del formato al medio al que está dirigido.

Introducción a CSS

Añadir estilos a un documento

□ Existen tres formas de añadir estilos a un documento:

- Estilos en línea.
 - ✓ El atributo `style` de todos elementos html permite añadir estilos particulares a cada elemento.
 - ✓ Se aplica al elemento que contiene el atributo.
 - ✓ No es recomendable su utilización.
 - No separa la presentación del contenido.
 - Dificulta la modificación de la presentación.
- Hojas de estilo incrustadas.
 - ✓ Se incluyen en el elemento `head`, mediante la etiqueta `style`.
 - Se utiliza con el atributo `type` con el valor `"text/css"`.
 - ✓ Se aplica a todos los elementos del documento.
- Hojas de estilo externas.
 - ✓ Las definiciones de los estilos se almacenan en un archivo de texto independiente.
 - ✓ Se aplica a todos los documentos que enlacen con el archivo mediante...

```
<link href="nombre_archivo" rel="stylesheet" type="text/css" />
```

Introducción a CSS

Añadir estilos a un documento

Enlace con una hoja de estilos externa

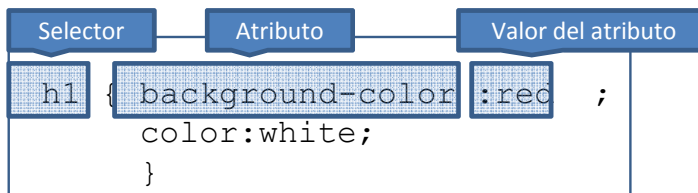
Hoja de estilos incrustada

Estilos en línea

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
2   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4 <head>
5   <meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
6   <title>estilos</title>
7   <meta name="generator" content="Amaya, see http://www.w3.org/Amaya/" />
8   <link href="estilos.css" rel="stylesheet" type="text/css" />
9
10  <style type="text/css">
11    <!--
12      h1 {background-color:red;
13         color:white;
14        }
15    -->
16  </style>
17 </head>
18 <body>
19 <h1>Titulo 1</h1>
20 <p style="margin-left:5em;">Lorem ipsum dolor sit amet, consectetur
21 adipiscing
22 elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque
23 penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam
24 felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat
```

Sintaxis

- ❑ Tanto las hojas de estilo externas, como las hojas de estilo incrustadas están formadas por una serie de declaraciones.
 - Estos estamentos pueden ser de dos clases:
 - ✓ Reglas.
 - ✓ Reglas arroba.
- ❑ Reglas:



- Están formadas por:
 - ✓ Un selector.
 - ✓ Una o varias parejas formadas por:
 - Un atributo.
 - Valor del atributo.
 - ✓ El atributo y su valor están separados por dos puntos (:).
 - ✓ Cada pareja está separada por punto y coma (;).
 - ✓ Todas las parejas están encerradas entre llaves.

Sintaxis (II)

❑ Reglas arroba.

- Comienzan por el carácter arroba (@).
- A continuación aparece un identificador.
- Pueden llevar un conjunto de declaraciones.
- Permiten dar información adicional al navegador.
- Ejemplos:

```
@import "otrahoja.css"
```

- ✓ Añade en el lugar que aparece la hoja de estilos otrahoja.css.

```
@media print{  
    * {font-size: xx-large;}  
}
```

- ✓ Incorpora estilos especiales cuando el medio de salida es la impresora.

❑ Comentarios.

- Tanto las hojas de estilo incrustadas como las hojas de estilo en línea pueden incluir comentarios.
 - ✓ Los comentarios se hacen al modo de C:
 - `/* comentario */`

Cascada

□ Cascada.

- El estilo de un elemento puede repetirse tanto en una hoja de estilo externa, una hoja incrustada o en un estilo en línea.
- Además, mediante la regla arroba `@import`, permite importar estilos desde otras hojas de estilo.

```
@import "hoja_de_estilo.css";
```

```
@import url("hoja_de_estilo.css");
```

- ✓ Estas declaraciones deberían ir antes que cualquier otra regla en la hoja de estilo.
- En todas estas definiciones se ha establecido un orden de prioridades en la cascada para determinar el estilo del elemento.

Cascada (II)

□ El orden de la cascada.

- Una hoja de estilo puede tener tres orígenes.
 - ✓ El autor, a través de la definición de estilos.
 - ✓ El usuario.
 - También puede especificar su propia hoja de estilo.
 - Esto es especialmente útil para aquellas personas con necesidades especiales.
 - ✓ El navegador, que también tiene una hoja de estilo por omisión.
- Estos tres orígenes tienen un orden de cascada que determina las características de presentación de un elemento, aunque éste aparezca en varios lugares.
- Algunas reglas pueden tener la declaración `!important`.
 - ✓ Una regla declarada como `!important`, tendrá más prioridad que una regla normal.

```
p {font-size: 2em !important}
```
 - ✓ Esta declaración permite a los usuarios con necesidades especiales hacer que prevalezcan sus estilos frente a los estilos del autor.

Cascada (III)

- ❑ Las aplicaciones de usuario deben aplicar la siguiente orden de disposición.
 - Encontrar todas las declaraciones que se aplican a la propiedad de un elemento concreto para el tipo de medio al que está dirigido.
 - Clasificar por la importancia y origen las distintas declaraciones. De menos a más prioridad:
 1. Hojas de estilo del navegador.
 2. Hojas de estilo normales del usuario.
 3. Hojas de estilo normales del autor.
 4. Hojas de estilo importantes del autor.
 5. Hojas de estilo importantes del usuario.
 - Clasificar por lo específico del selector: los más específicos sustituyen a los más generales.
 - Si dos reglas tienen el mismo nivel de importancia, origen y especificad, la última en ser especificada será la de máxima prioridad.
 - ✓ Las hojas de estilo importadas se consideran que están antes de cualquier otra regla.
 - ✓ En los estilos generados por el autor, la máxima prioridad estará en el último declarado, es decir los estilos en línea.
 - A continuación las hojas de estilo incrustadas y las hojas de estilo externas.

Herencia

□ Herencia.

- Algunos valores son heredados por los elementos hijo de un elemento.
 - ✓ Por ejemplo, en el código...

```
<p>Este es un <em>trozo</em> de c&oacute;digo</p>
```

 - ... el contenido del elemento `em` heredará las características asignadas al elemento `p`.
- Hay que tener en cuenta que:
 - ✓ No se heredan todas las propiedades. Cada una define si se hereda o no.
 - ✓ A todos los elementos se les puede aplicar el valor `inherit` en cualquier propiedad.
 - Este valor se utiliza para conseguir que un elemento herede del padre una propiedad no heredada.
 - ✓ Cuando se asigna una propiedad a un elemento, su valor sustituye al valor heredado.
 - ✓ Los elementos heredan los valores computados del padre, no los especificados.

Valores de las propiedades

- ❑ Cuando la aplicación ha construido el documento debe asignar un valor a cada propiedad.
- ❑ El valor final se resuelve...
 - Se determina por medio del valor especificado en la propiedad (valor especificado).
 - Se resuelve a partir de un valor calculado a partir de la herencia (valor computado).
 - Si es necesario, se convierte en el valor usado absoluto, en el caso de que sea relativo a otros (valor usado).
 - Se transforma según las limitaciones técnicas del entorno (valor real).
- ❑ Valor especificado.
 - Valor asignado a la propiedad en la declaración o heredado.
 - ✓ Si no aparecen estos valores se utiliza el valor predeterminado de la propiedad.
- ❑ Valor computado.
 - Cuando se trata de un valor relativo que necesita de un cálculo para obtener su valor absoluto.
 - ✓ Si el elemento `body` se ha especificado con un tamaño de letra de 10 pixels, y el elemento `p` con un tamaño de 125%, el contenido de `p` tendrá un tamaño de letra calculado a partir del valor especificado heredado: 12,5 pixels.
- ❑ Valor usado.
 - Cuando se trata de un valor relativo a otros.
 - ✓ Por ejemplo, si se trata de un porcentaje sobre la anchura del bloque donde está situado (la anchura puede depender del medio donde se ejecute).
- ❑ Valor real.
 - Puede que el valor resultante no pueda ser utilizado por limitaciones técnicas del navegador.
 - ✓ El valor real se ajustará lo máximo posible al valor computado o al valor usado.

Valores de las propiedades (II)

❑ Medidas.

- Están formadas por un número (real o entero) y un identificador que indica la unidad de medida.
- El identificador puede indicar medidas absolutas o relativas.
 - ✓ Medidas relativas.
 - em. Valor computado del cuerpo de letra del elemento en que se usa.
 - Cuando se usa en la propiedad `font-size`, hace referencia al cuerpo de letra del elemento padre.
 - ex. Altura de la letra x minúscula de la fuente relevante.
 - px. Pixels.
 - ✓ Medidas absolutas.
 - in. Pulgadas (2,54 centímetros).
 - cm. Centímetros.
 - mm. Milímetros.
 - pt. Puntos (1/72 pulgadas).
 - pc. Picas (12 puntos).

❑ Porcentajes.

- Número seguido del símbolo %.

❑ URL.

- `url(especificaciónDeURL)`.
 - ✓ La especificación puede ir encerrada o no entre comillas simples o dobles.

Valores de las propiedades (III)

□ Color.

- Puede especificarse:
 - ✓ Mediante un identificador de color.
 - aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, orange, purple, red, silver, teal, white, y yellow.
 - ✓ Mediante la cantidad de rojo, verde y azul en hexadecimal.
 - #RRVVA.
 - ✓ Mediante la cantidad de rojo, verde y azul en decimal.
 - `rgb(rrr, vv, aaa)`.
 - ✓ Mediante porcentajes sobre la cantidad de rojo, verde y azul.
 - `rgb(rrr%, vv%, aaa%)`.
 - ✓ La especificación también recoge identificadores para los colores del sistema utilizados en la interfaz de usuario (`ActiveBorder`, `ButtonFace`, `CaptionText`, etc.).
 - En CSS3 la asignación de colores del sistema parece que estarán desaconsejados.

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

Valores de las propiedades (IV)

Ángulos.

- Un número seguido de un identificador de la unidad de medida.
 - ✓ deg, grados.
 - ✓ rad, radianes.
 - ✓ grad, gradianes.

Tiempos.

- Un número seguido de un identificador de la unidad de medida.
 - ✓ ms, milisegundos.
 - ✓ s, segundos.

Frecuencias.

- Un número seguido de un identificador de la unidad de medida.
 - ✓ Hz, herzio.
 - ✓ kHz, kiloherzio.

Cadenas.

- Se pueden encerrar entre comillas simples o dobles.

Identificadores.

- Sólo pueden estar formados por caracteres alfanuméricos y el guión. Deben comenzar por un carácter alfabético.

`inherit`.

- Indica que se trata de un valor heredado.

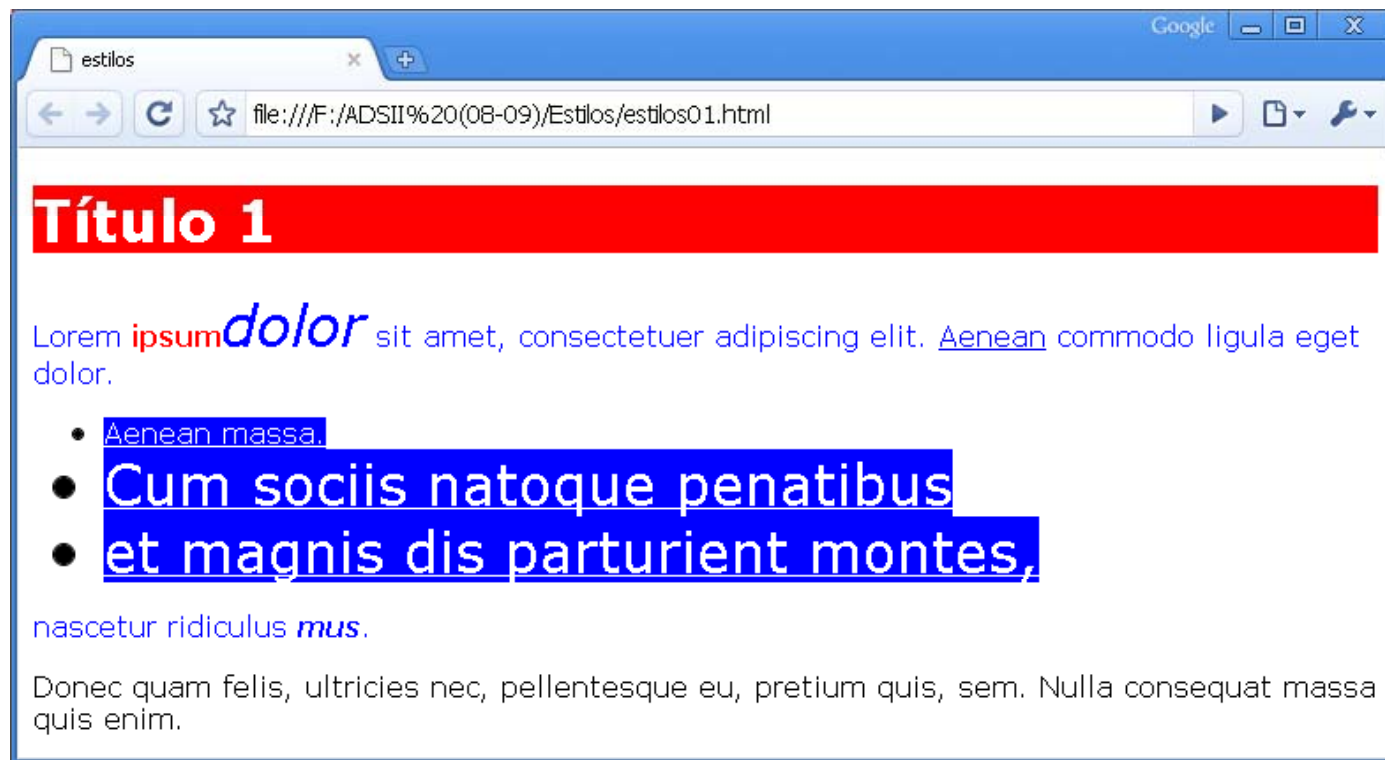
Selectores

- Se utilizan para identificar el elemento sobre el que se aplicará el estilo.
- Selector universal.
 - Se utiliza el asterisco (*) para indicar que el estilo se aplicará a todos los elementos.
- Selector de elemento.
 - Se utiliza el nombre del elemento.
- Selectores de descendiente.
 - La indicación E1 E2 en un selector indicará que el estilo se aplicará a los elementos E2 que sean descendientes de E1.
- Selectores de hijos.
 - La indicación E1 > E2 en un selector indicará que el estilo se aplica a los elementos E2 que sean hijos directos de E1.
- Selectores de hermanos adyacentes.
 - La indicación E1 + E2 en un selector indicará que el estilo se aplica al elemento E2 que este inmediatamente después de un elemento E1.

Selectores (II)

```
<style type="text/css">
  /*Selectores universales */
  /* Todos los elementos tienen un tipo de letra Verdana */
  * {font-family: Verdana, Arial, Helvetica, sans-serif;}
  /*Selectores de elemento */
  /*Todas las cabeceras h1 tienen el fondo rojo y la letra blanca */
  h1 {background-color:red; color:white;}
  /*Todos los párrafos tienen la letra azul */
  p {color:blue;}
  /*Selectores de descendiente */
  /* Todos los enlaces que estén dentro de un elemento li tienen el fondo azul y la letra blanca */
  li a {color:white; background-color:blue;}
  /*Selectores de hijo*/
  /*Todos los elementos strong que estén dentro de un elemento p son de color rojo */
  p>strong {color:red;}
  /*Selectores de hermanos */
  /* Todos los elementos li que estén junto a otro elemento li tienen el cuerpo de letra de dos em */
  li+li{font-size: 2em;}
  /* Los elementos se pueden combinar...
  Todos los elementos em que sean adyacentes a un elemento strong que sea hijo directo
  de un elemento p tiene el cuerpo de la letra mayor */
  p>strong+em {font-size:2em;}
</style>
</head>
<body>
<h1>T&iacute;tulo 1</h1>
<p>Lorem <strong>ipsum</strong><em> dolor</em> sit amet, consectetur adipiscing elit.
<a href="www.colimbo.net">Aenean</a> commodo ligula eget dolor. </p>
<ul>
  <li><a href="www.colimbo.net">Aenean massa.</a></li>
  <li><a href="www.colimbo.net">Cum sociis natoque penatibus</a> </li>
  <li><a href="www.colimbo.net">et magnis dis parturient montes,</a> </li>
</ul>
<p>nascetur ridiculus <em><strong>mus</strong></em>.</p>
<div>Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. </div>
</body>
```

Selectores (III)



Selectores

Selectores de atributo

□ Permiten seleccionar elementos en función de los atributos que contengan.

- Se pueden establecer de cuatro formas distintas.

✓ [atrib]

- Selecciona a los elementos que contengan el atributo `atrib`, independiente de su valor.

- `h1[title] {color:red;}`

Selecciona los elementos `h1` que contengan el atributo `title`.

✓ [atrib=valor]

- Selecciona a los elementos que contengan el atributo `atrib` con el valor especificado.

- `[class="ejemplo"] {font-family:Verdana;}`

Selecciona los elementos cuyo atributo `class` sea igual a "ejemplo".

- `div[class="ejemplo"][lang="fr"] {display:none;}`

- [atrib ~ =valor]

✓ Selecciona los elementos que en el atributo `atrib` contengan la palabra `valor`.

- [atrib|=valor]

✓ Selecciona los elementos en los que el atributo `attrib` comience con la palabra `valor`, separada por un guión.

- Está pensada para elegir los códigos de lenguaje-sublenguaje del atributo `lang` ("es-ES", "en-US", etc.).

Selectores

Selectores de clase

- ❑ Una de las formas más comunes de dar estilos a determinados elementos es mediante el atributo `class`.

```
[class="centrado"] {text-align:center}
```

...

```
<h1 class="centrado">Encabezado</h1>
```

```
<p class="centrado">bla, bla, bla...</p>
```

✓ Estos dos elementos aparecerían centrados.

```
<div class="centrado">...</div>
```

✓ Todos los elementos contenidos en la etiqueta `div` aparecerían centrados.

- ❑ El asignar clases a elementos `div` y `span` es una forma muy habitual para dar estilo a varias etiquetas de forma conjunta o a fragmentos de contenido que no están en ninguna etiqueta.
- ❑ CSS permite declarar el estilo de una clase utilizando simplemente el carácter punto (`.`).

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

```
p.rojo {color:red;}
```

Selectores

Selectores de clase (II)

```
<style type="text/css">
  <!--
  * {font-family: Verdana, Arial, Helvetica, sans-serif;}
  h1 {background-color:red;
      color:white;
      }
  .sangrado {margin-left:5em;}
  .rojo {color:red;}
  -->
</style>
</head>
<body>
<div class="sangrado">
  <h1>T&iacute;tulo 1</h1>

<p>Lorem <strong>ipsum</strong> <em>dolor</em> sit amet, <span class="rojo">
Consectetuer adipiscing</span> elit. <a href="www.colimbo.net">Aenean</a>
commodo ligula eget dolor. </p>
<ul>
  <li><a href="www.colimbo.net">Aenean massa.</a></li>
  <li><a href="www.colimbo.net"><span class="rojo">Cum sociis natoque
penatibus</span></a> </li>
  <li><a href="www.colimbo.net">et magnis dis parturient montes,</a>
</li>
</ul>
<p>nascetur <strong>ridiculus</strong> mus. Donec quam felis, ultric
pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim.
</div>

<p>Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu
enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam
felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivam
elementum semper
```



Selectores

Selectores de id

- ❑ Permite seleccionar los elementos que tengan un valor determinado en el atributo `id`.
 - Si un elemento tiene un atributo `id`, su valor debe ser único en todo el documento.
- ❑ El selector de id utiliza el formato `#valorId`.
 - Por su exclusividad, se suele utilizar para especificar la estructura visual del documento.
 - ✓ Establece el estilo del elemento identificado como `menunavegacion`.

```
#menunavegacion {border-color:blue;}
```
 - ✓ Establece el estilo de los enlaces que estén dentro del elemento con `id` `menu-izq`.

```
#menu-izq a {text-decoration: none;}
```
 - ✓ Establece el estilo de la cabeceras que tenga el `id` `titulolibro`.

```
h1#titulolibro {font-size: 200%;}
```


Selectores

Pseudo-elementos y pseudo-clases

- ❑ Todos los selectores anteriores pueden deducirse de la estructura del documento.
- ❑ Los pseudo-elementos y las pseudo-clases permiten aplicar estilos a partir de información que no tiene que ver con la estructura del documento:
 - Pseudo-elementos.
 - ✓ Permiten crear una abstracción sobre la estructura del documento, pero no especificada por el código fuente.
 - Por ejemplo, la primera letra o la primera línea de un elemento.
 - Pseudo-clases.
 - ✓ Clasifican un elemento mediante propiedades no basadas en la estructura del documento.
 - Pseudo-clases dinámicas.
 - Un elemento puede pertenecer o dejar de pertenecer a una pseudo-clase a medida que el usuario interactúa con el documento.
 - Pseudo-clases estáticas.
 - En ocasiones sus propiedades se pueden deducir de la estructura del documento.

Selectores

Pseudo-elementos y pseudo-clases (II)

❑ Pseudo-elemento `:first-line`.

- Aplica los estilos a la primera línea de un párrafo.
- El concepto de "primera línea" dependerá del entorno en el que se representa el documento.

❑ Pseudo-elemento `:first-letter`.

- Aplica los estilos al primer carácter de un párrafo.
- Se puede utilizar para letras capitulares.

❑ Pseudo-clase `:first-child`.

- Hace referencia al primer elemento hijo de uno dado.

```
div> p:first-child {text-indent: 0}
```

- ✓ El primer párrafo de un elemento hijo dentro de un elemento `div` no tendrá sangría.

❑ Pseudo-clase `:lang(codigoIdioma)`.

- Permite seleccionar los elementos que contengan un atributo `lang` en un idioma determinado.

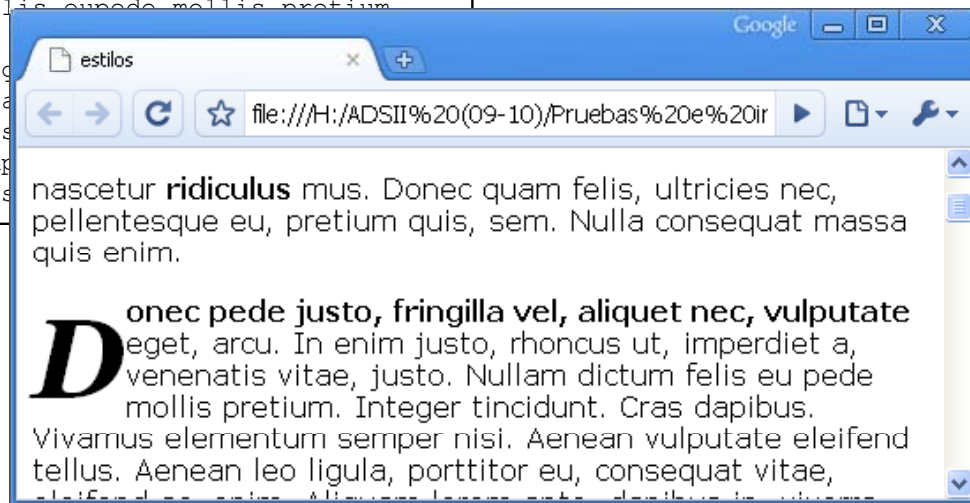
Selectores

Pseudo-elementos y pseudo-classes (III)

```
p.primerparrafo:first-line {font-weight:bold;}
p.primerparrafo:first-letter {font-size:400%;
    font-style:italic; font-family:Georgia,
    "Times New Roman", Times, serif; float:left;}
```

```
<p>nascetur <strong>ridiculus</strong> mus. Donec quam felis,
ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat
massa quis enim.</p> </div>
```

```
<p class="primerparrafo">Donec pede justo, fringilla vel, aliquet
nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a,
venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium
Integer tincidunt. Cras dapibus. Vivamus
vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae,
eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus
varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies
```



Selectores

Pseudo-elementos y pseudo-clases (IV)

❑ Pseudo-clases para enlaces.

- Permiten modificar el formato estándar de los enlaces.
 - ✓ `:link`, hace referencia a un enlace no visitado.
 - ✓ `:visited`, hace referencia a un enlace visitado.
- Pseudo-clases dinámicas.
 - ✓ Permiten actuar por acciones del usuario.
 - `:hover`, el formato se aplica cuando el dispositivo apuntador pasa por encima del enlace.
 - `:active`, el formato se aplica en el momento en que el usuario activa el enlace (por ejemplo, al pulsar sobre él).
 - `:focus`, el formato se aplica cuando es el elemento activo de la página.

```
a:hover { font-weight:bold;}
a:focus {color:red;}
a:active {color:red;
          font-weight:bold;}
}

li a {text-decoration:none;
      color:black; display:block}
}
li a:active, li a:hover, li a:focus
{background-color:blue;
  color:white;
}
```



Colores y fondos

- ❑ Propiedad `color`.
 - Permite establecer el color de primer plano.
 - El valor sería una especificación de color o el valor `inherit` (el color de primer plano no se hereda).
- ❑ Propiedad `background-color`.
 - Establece el color de fondo.
 - ✓ En los elementos de bloque, el color de fondo será toda la caja del bloque.
 - ✓ En los elementos en línea, el color de fondo será únicamente el área del contenido.
 - El valor sería una especificación de color, el valor `inherit` (la propiedad no se hereda) o el valor `transparent` (valor por omisión).
 - Cada caja (de bloque o en línea) tiene un margen y un área de relleno.
 - ✓ El margen siempre es transparente.
 - ✓ El área de relleno toma el color de fondo.

```
h1 {color:white;
     background-color:rgb(255,0,0);
}
em {color:#ffffff;
    background:rgb(100%,0,0);
}
```



Colores y fondos (II)

- ❑ También es posible agregar una imagen de fondo:
 - La propiedad `background-image` permite agregar una imagen de fondo.
 - Su valor será el URL del recurso, el valor `inherit` o el valor `none`.
 - La imagen se repite a lo largo de todo el fondo.
 - ✓ Con la propiedad `background-repeat` es posible modificar eso:
 - Valor `repeat` (valor por omisión), la imagen se repite a lo largo de todo el fondo.
 - Valor `no-repeat`, la imagen aparece una sola vez.
 - Valor `repeat-x`, la imagen se repite a lo largo de todo el eje de las x.
 - Valor `repeat-y`, la imagen ser repite a lo largo de todo el eje de las y.
 - Valor `inherit`.
 - La imagen se irá desplazando con el scroll del documento.
 - ✓ La propiedad `background-attachment` permite modificar ese comportamiento.
 - Valor `scroll` (por omisión).
 - Valor `fixed`, la imagen se mantiene fija.
 - Valor `inherit`.

Fuentes

❑ Propiedad `font-family`.

- Permite establecer una lista con la familia de fuentes preferidas que se utilizarán para presentar el texto.
 - ✓ Una familia de fuentes puede estar formada por varios tipos de letra.
 - Por ejemplo, en la definición de la letra Verdana, se define un tipo normal, otro en negrita, otro en cursiva, en ocasiones otro condensada, etc.
 - ✓ Existen tres tipos de familias genéricas: `serif` (tipo Times), `sans-serif` (tipo Arial), `mono` (tipo courier), `cursive` (tipo script), o `fantasy`.
 - ✓ En la lista se pondrán por orden de preferencia.
 - Si la primera fuente no está presente, se pasará a la siguiente.
 - Es conveniente acabar la lista con una familia genérica por si no está disponible ninguna de las otras.
 - Ejemplos de valores de la propiedad:
Verdana, Arial, Helvetica, sans-serif
Georgia, "Times New Roman", Times, serif
"Courier New", Courier, mono
 - ✓ El valor de la propiedad se hereda.

❑ Propiedad `font-style`.

- Estilo del tipo de letra.
- Puede tomar los valores `normal`, `italic`, `oblique` o `inherit`.
 - ✓ La letra `oblique`, es una letra normal inclinada y será la que se utiliza cuando no hay disponible un tipo en itálica en la familia de fuentes.

Fuentes (II)

❑ Propiedad `font-variant`.

- Puede tomar los valores `normal` o `small-caps` (versalita).

❑ Propiedad `font-weight`.

- Indica el peso de la fuente.
- Puede tomar los valores `normal`, `bold`, `bolder`, `lighter`.
- También puede tomar valores de centenas entre 100 y 900.
 - ✓ La letra `normal` equivale al valor 400.
 - ✓ La letra `bold` equivale al 700.
 - ✓ `bolder` y `lighter` equivalen un peso mayor o menor que el peso de la letra heredada.

❑ Propiedad `font-size`.

- Establece el tamaño de la letra.
- Puede tomar como valor:
 - ✓ Un tamaño absoluto (`xx-small`, `x-small`, `small`, `medium`, `large`, `x-large`, `xx-large`).
 - ✓ Un tamaño relativo (`larger` o `smaller`).
 - ✓ Una medida o un porcentaje.

Texto

- ❑ Propiedad `text-indent`.
 - Indica la sangría de la primera línea de un párrafo.
 - Puede tomar como valor una medida, un porcentaje o `inherit`.
- ❑ Propiedad `text-align`.
 - Permite especificar la alineación de las líneas de un bloque de texto.
 - Puede tomar los valores `left`, `right`, `center`, `justify` o `inherit`.
 - ✓ No todos los navegadores manejan bien el texto justificado.
- ❑ Propiedad `text-decoration`.
 - Añade decoración al texto.
 - Puede tomar los valores `none`, `underline`, `overline`, `line-through`, `blink` o `inherit`.
 - ✓ Pueden aparecer varios de estos valores: `underline`, `overline`, `line-through` o `blink`.
- ❑ Propiedad `text-transform`.
 - Permite cambiar los caracteres a mayúsculas o minúsculas.
 - Puede tomar los valores `capitalize` (pone el primer carácter en mayúsculas), `uppercase`, `lowercase`, `none` o `inherit`.

Texto (II)

□ Espaciado entre palabras y caracteres.

- La propiedad `word-spacing` permite modificar la separación entre palabras.
 - ✓ Puede tomar como valor una medida, la palabra `normal` o `inherit`.
- La propiedad `letter-spacing` especifica el espacio entre caracteres.
 - ✓ Puede tomar como valor una medida, la palabra `normal` o `inherit`.

Listas

❑ Propiedad `list-style`.

- Permite cambiar el tipo de viñeta (para listas no ordenadas) o de numeración (para listas ordenadas).
 - ✓ En listas no ordenadas puede tomar los valores `disc`, `circle`, `square`, `none` o `inherit`.
 - ✓ En listas ordenadas puede tomar los valores `decimal`, `decimal-leading-zero`, `lower-roman`, `upper-roman`, `lower-alpha` o `lower-latin`, `upper-alpha` o `upper-latin`, `none` o `inherit`.

❑ Propiedad `list-style-image`.

- Permite sustituir la viñeta por una imagen.
- Puede tomar como valor un `url`, `none` o `inherit`.

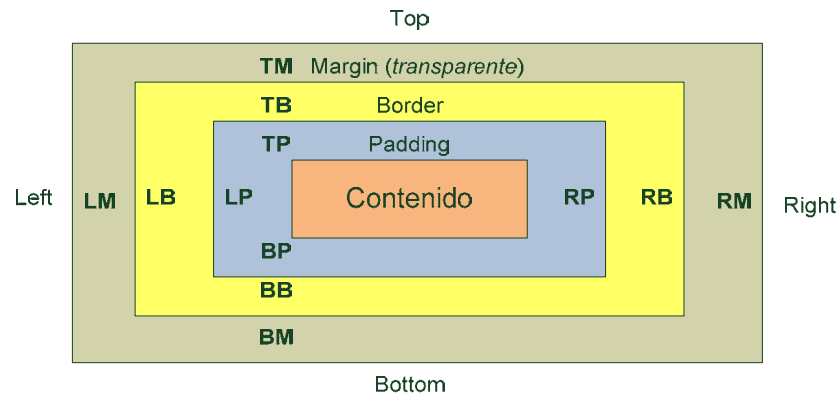
❑ Propiedad `list-style-position`.

- Especifica la posición de la viñeta.
- Puede tomar los valores `inside`, `outside`, o `inherit`.
 - ✓ Con `outside`, la viñeta queda fuera de la caja de texto.
 - ✓ Con `inside`, la viñeta forma parte de la caja de texto.

- Con la propiedad `outside` la viñeta queda así
- Con la propiedad `inside` la viñeta queda así

Modelo de caja

- ❑ Cada elemento html genera una caja.
 - El elemento `body` genera una *caja de contención* que incluye todas las cajas de los elemento.
 - Cuando existen elementos anidados, cada elemento padre también actúa de caja de contención de los elementos hijos.
 - El modelo de cajas permite tener control no sólo sobre el contenido del documento, sino también sobre las cajas, pudiendo especificar para cada elemento su margen, su fondo, sus dimensiones, su color, su posición, etc.
- ❑ Cada caja tiene un área de contenido y áreas circundantes opcionales de relleno (`padding`), borde (`border`) y margen (`margin`).
 - El tamaño de cada área se puede definir por propiedades CSS.



Modelo de caja (II)

- ❑ El margen, el borde y el relleno se pueden dividir en los segmentos superior (*top*), derecho (*right*), inferior (*bottom*) e izquierdo (*left*).
 - En el gráfico aparecen como LM (margen izquierdo), RB (para el borde derecho), TP (para el relleno superior), etc.
- ❑ El perímetro de cada una de las cuatro áreas es el *limite*.
 - Límite del contenido.
 - ✓ Viene dado por la anchura y la altura del contenido procesado del elemento.
 - Se modifica mediante la propiedad `width` y la propiedad `height` de la caja.
 - Límite de padding.
 - ✓ Si tiene anchura 0, su límite será el mismo que el del contenido.
 - Límite del borde.
 - ✓ Si tiene anchura 0, su límite será el mismo que el del área de padding.
 - Límite del margen.
 - ✓ Si tiene anchura 0, su límite será el mismo que el del área de padding.
- ❑ El tamaño total de la caja será el del límite del margen.
- ❑ Estilo del fondo de las distintas áreas.
 - Área de contenido y de padding: propiedad `background` del elemento.
 - Borde: propiedades del borde del elemento.
 - Área de margen: siempre es transparente.

Modelo de caja:

Propiedades del margen

- ❑ Propiedades `margin-top`, `margin-right`, `margin-bottom` y `margin-left`.
 - Especifica el tamaño del margen.
 - Admite como valores una unidad de medida, un porcentaje, `inherit` (la propiedad no hereda) o `auto`.
 - ✓ El valor `auto` hace que el valor lo determina el navegador.
- ❑ Propiedad `margin`.
 - Permite establecer los distintos valores del margen.
 - ✓ Si sólo tiene un valor se aplica a todos los márgenes.
 - ✓ Si tiene dos valores, el primer valor fija los márgenes superior e inferior, mientras que el segundo los márgenes izquierdo y derecho.
 - ✓ Si hay tres valores, el primer valor define el margen superior, el segundo los márgenes izquierdo y derecho y el tercero el inferior.
 - ✓ Si hay cuatro valores se aplican a los márgenes superior, derecho, inferior e izquierdo respectivamente.

Modelo de caja:

Propiedades del relleno

- ❑ Propiedades `padding-top`, `padding-right`, `padding-bottom` y `padding-left`.
 - Especifica el tamaño del área de relleno.
 - Admite como valores una unidad de medida, un porcentaje o `inherit` (la propiedad no hereda).
- ❑ Propiedad `padding`.
 - Permite establecer los distintos valores del relleno.
 - ✓ Si sólo tiene un valor se aplica a todos los valores del relleno.
 - ✓ Si tiene dos valores, el primer valor fija el relleno superior e inferior, mientras que el segundo el relleno izquierdo y derecho.
 - ✓ Si hay tres valores, el primer valor define el relleno superior, el segundo el relleno izquierdo y derecho y el tercero el inferior.
 - ✓ Si hay cuatro valores se aplican al relleno superior, derecho, inferior e izquierdo respectivamente.

Modelo de caja:

Propiedades del borde

- ❑ Propiedades `border-top-width`, `border-right-width`, `border-bottom-width` y `border-left-width`.
 - Especifica el grosor del borde.
 - Admite como valores una unidad de medida, `inherit` (la propiedad no hereda) o uno de los valores `thin`, `thick` o `medium`.
- ❑ Propiedad `border-width`.
 - Permite establecer los distintos valores del relleno.
 - ✓ Si sólo tiene un valor se aplica a todos los valores borde.
 - ✓ Si tiene dos valores, el primer valor fija el borde superior e inferior, mientras que el segundo el borde izquierdo y derecho.
 - ✓ Si hay tres valores, el primer valor define el borde superior, el segundo el borde izquierdo y derecho y el tercero el inferior.
 - ✓ Si hay cuatro valores se aplican al borde superior, derecho, inferior e izquierdo respectivamente.
- ❑ Propiedades `border-top-color`, `border-right-color`, `border-bottom-color` y `border-left-color`.
 - Especifica el color del borde.
 - Admite como valores un color, `inherit` (la propiedad no hereda) o `transparent`.
- ❑ Propiedad `border-color`.
 - Especifica con 1, 2, 3, o cuatro valores el color de cada uno de los bordes.

Modelo de caja:

Propiedades del borde (II)

- ❑ Propiedades `border-top-style`, `border-right-style`, `border-bottom-style` y `border-left-style`.
 - Especifica el aspecto del borde.
 - Admite como valores:
 - ✓ `none`, ningún borde.
 - ✓ `hidden`, igual a `none`.
 - ✓ `dotted`, el borde es una serie de puntos.
 - ✓ `dashed`, el borde es una serie de pequeños segmentos de línea.
 - ✓ `solid`, el borde es un único segmento de línea.
 - ✓ `double`, el borde son dos líneas sólidas.
 - ✓ `groove`, el borde luce como si estuviese tallado en la página.
 - ✓ `ridge`, el borde parece que estuviera sobresaliendo de la página.
 - ✓ `inset`, el borde hace que toda la caja luzca como si estuviera empotrada en la página.
 - ✓ `outset`, el borde hace que toda la caja parezca sobresalir de la página.
- ❑ Propiedad `border-style`.
 - Especifica con 1, 2, 3, o cuatro valores el estilo de cada uno de los bordes.

Modelo de caja:

Propiedades del borde (III)

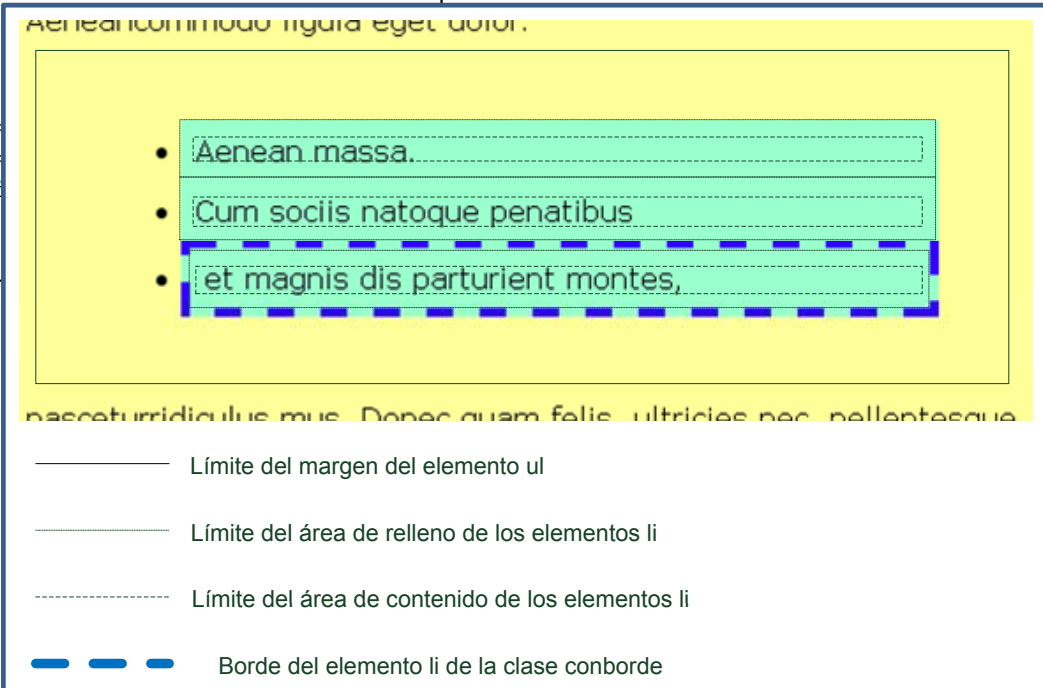
□ Propiedades resumidas del borde.

- Propiedades `border-top`, `border-right`, `border-bottom` y `border-left`.
 - ✓ Permiten definir el ancho, el estilo y el color de todos los bordes.
 - Recibe los valores de ancho, estilo y color separados por un espacio en blanco.
- Propiedad `border`.
 - ✓ Permite definir el ancho, el estilo y el color de los cuatro bordes.

Modelo de caja: Ejemplo

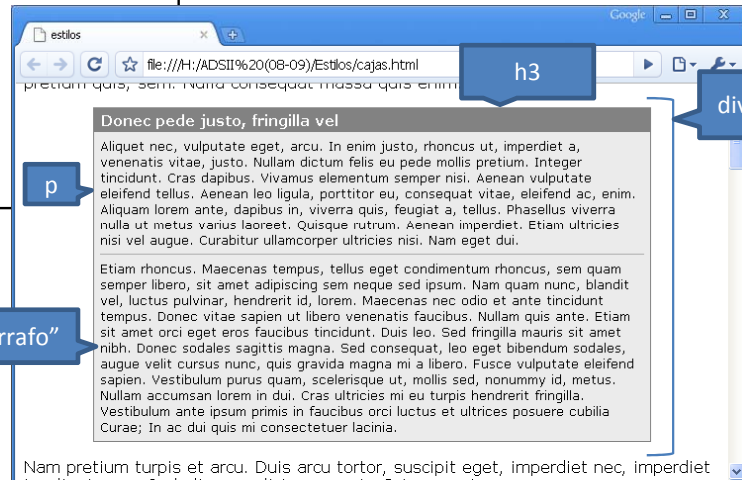
```
* {font-family: Verdana, Arial, Helvetica, sans-serif;
  background-color:#FFFF99;}
ul {margin: 3em;}
li {background-color:#99FFCC;
    padding: 0.5em;}
li.conborde {border:thin dashed #3300FF;}
...

<p>...</p>
<ul>
  <li class="flecha">Aenean massa.</li>
  <li>Cum sociis natoque penatibus</li>
  <li class="conborde">et magnis dis</li>
</ul>
<p>...
```



Modelo de caja: Ejemplo (II)

```
* {font-family:Verdana, Arial, Helvetica, sans-serif}
.nota {font-size:80%;
margin: 0 10%;
padding: 0;
background-color:#EBEBEB;
border: 1px solid #828282;
}
.nota p{margin:0.5em;
padding:0 0 0.5em 0;
border-bottom:thin solid #ACACAC;
}
.nota h3 {margin:0;
padding:0.2em 0.5em;
background-color:#828282;
color:#ffffff;
}
.nota p.ultimoparrafo{border-bottom:none;
padding-bottom:0;
}
```



Formato visual

- ❑ El formato visual de CSS permite definir el tipo y dimensiones de la caja, su posición, comportamiento y relación con otras cajas de la estructura del documento.
- ❑ Las aplicaciones para *medios continuos* (por ejemplo, navegadores gráficos) ofrecen un *acceso visual* (una ventana).
 - El contenido del documento se visualiza a través del acceso visual.
- ❑ Bloques de contención.
 - Un bloque de contención es una caja rectangular que delimita el contenido del mismo y de los elementos descendientes.
 - ✓ Cada caja generada actúa como bloque de contención de las cajas descendientes.
 - El elemento raíz de un documento forma el bloque de contención inicial.

Formato visual

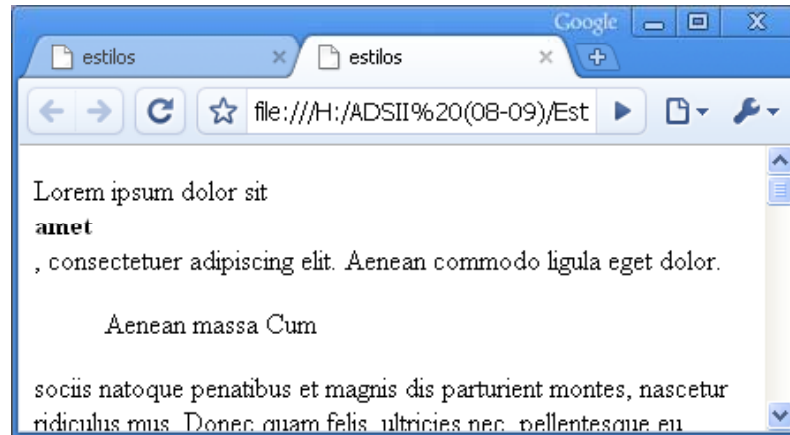
Tipos de bloques y propiedad display

- ❑ Elementos a nivel de bloque y cajas de bloque.
 - Son aquellos elementos del documento fuente tratados como bloques (por ejemplo las cabeceras o los párrafos).
 - Generan una caja de bloque principal.
 - ✓ La caja de bloque principal actúa de bloque de contención para las cajas de los elementos descendientes.
 - Algunos elementos generan cajas adicionales.
 - ✓ Por ejemplo, los elementos `list-item`, generan cajas fuera del bloque para contener viñetas o datos alfanuméricos.
- ❑ Elementos a nivel de línea y cajas en línea.
 - No forman nuevos bloques.
 - ✓ Texto enfatizado, elementos `span`, imágenes...
 - Su contenido es distribuido en líneas.
- ❑ Propiedad `display`.
 - Permite determinar el tipo de caja generado por el elemento.
 - Puede tomar los valores:
 - ✓ `block`, el elemento generará una caja de bloque.
 - ✓ `list-item`, genera una caja principal y una cada adicional para viñetas o datos alfanuméricos.
 - ✓ `inline`, el elemento generará una caja en línea.
 - ✓ `none`, el elemento no genera ninguna caja.
 - El elemento no se visualizará.

Formato visual

Tipos de bloques y propiedad display (II)

```
<p>Lorem ipsum dolor sit <strong style="display:block">amet</strong>, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. </p>
<ul>
  <li style="display:inline">Aenean</li>
  <li style="display:inline"> massa</li>
  <li style="display:inline">Cum</li>
</ul>
<p>sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. </p>
```



Formato visual

Esquemas de posicionamiento

- Una caja se presenta en el acceso visual en alguno de estos tres esquemas de posicionamiento:
 - Flujo normal.
 - ✓ Los elementos se muestran uno detrás de otro según la posición en la que aparecen en el documento.
 - Flotantes.
 - ✓ La caja se presente de acuerdo al flujo normal, y después se mueve a la izquierda o derecha tanto como sea posible.
 - El contenido del elemento siguiente fluye a lo largo del costado del siguiente.
 - Posicionamiento absoluto.
 - ✓ La caja sale del flujo normal y se le asigna una posición respecto al bloque de contención.

Formato visual

Esquemas de posicionamiento (II)

❑ Propiedad `position`.

- Determina el esquema de posicionamiento.
 - ✓ Puede tomar los valores:
 - `static`, (propiedad por omisión) la caja se presenta de acuerdo al flujo normal.
 - `relative`, la caja se presenta respecto al flujo normal y luego se desplaza a una posición relativa respecto a la posición normal.
 - La posición de las cajas siguientes se calculan como si la caja no se hubiera desplazado.
 - `absolute`, la posición y tamaño de la caja se fijan con las propiedades `top`, `right`, `bottom` y `left`, respecto al bloque de contención.
 - Estas cajas se sacan del flujo normal.
 - `fixed`, se posicionan de forma absoluta, pero su posición se fija respecto al acceso visual (ventana del navegador).

❑ Propiedades `top`, `right`, `bottom`, `left`.

- Especifican el desplazamiento de la caja respecto al bloque de contención.
- Pueden tomar como valores una medida, un porcentaje, el valor `auto` o `inherit`.

Formato visual

Flujo normal

❑ Cajas de bloque.

- Se colocan una después de otra verticalmente.
- La distancia vertical entre cajas hermanas la determina la propiedad `margin`.
 - ✓ Los márgenes verticales de cajas adyacentes se funden para formar uno sólo.

❑ Cajas de línea.

- Se colocan horizontalmente a partir de la parte superior del bloque de contención.
- El ancho de la caja lo determina el bloque de contención y la presencia de flotantes.
- Cuando la altura de una caja en línea es menor que la caja que la contiene, su alineación vertical se determina por la propiedad `vertical-align`.
- Cuando el ancho de las cajas en línea es menor que el ancho de la caja que la contiene su distribución vertical la establece la propiedad `text-align`.
- Si una caja de línea excede el ancho de la caja que la contiene, se divide en varias cajas que son distribuidas en varias cajas en línea.

Formato visual

Posicionamiento relativo y absoluto

❑ Posicionamiento relativo.

- Una caja de posicionamiento relativo se sitúa inicialmente respecto al flujo normal.
- Las propiedades `top`, `right`, `bottom` y `left` permiten su desplazamiento.
 - ✓ No modifican el tamaño, sólo desplazan.
 - Para el desplazamiento horizontal basta especificar `right` o `left`.
 - Para el desplazamiento vertical basta especificar `top` o `bottom`.
- Las cajas siguientes se posicionan como si la caja con posicionamiento relativo no se hubiera desplazado.
 - ✓ Puede haber desbordamiento.

❑ Posicionamiento absoluto.

- La caja se sale del flujo normal.
 - ✓ Su posición no interviene en la posición de los siguientes elemento.
 - ✓ Los hijos de la caja seguirán el flujo normal.
- Las propiedades `top`, `right`, `bottom` y `left`, establecen su posición.
- La propiedad `position` puede tener los valores `absolute` o `fixed`.
- Posicionamiento fijo.
 - ✓ La posición se establece respecto al acceso visual.

Formato visual

Flotantes

- ❑ Cajas que se desplazan a la izquierda o derecha de la línea actual.
 - El contenido de la siguiente caja fluye a lo largo del costado de la caja.
- ❑ La propiedad `float` permite especificar hacia dónde se desplazará el contenido de la caja.
 - `float:left`, el contenido flota a la izquierda.
 - ✓ El contenido de la siguiente caja fluye por el costado derecho.
 - `float:right`, el contenido flota a la derecha.
 - ✓ El contenido de la siguiente caja fluye por el costado izquierdo.
 - `float:none`, el contenido no fluye.
- ❑ Si la siguiente caja no cabe en el bloque de contención se coloca según el flujo normal.
- ❑ Si el tamaño de la caja flotante sobrepasa el límite inferior de la siguiente caja, también flotará en el siguiente bloque.

Formato visual

Flotantes (II)

```
<style type="text/css">
  <!--
    .flotarizquierda {float:left;}
    .flotarderecha {float:right;}
    .sinflotar {float:none;}
  -->
</style>

...
<p>Lorem
ipsum dolor sit ... </p>

<p>Donec pede
justo, fringilla vel...</p>

<p class="ultimoparrafo">Etiam rhoncus. Maecenas tempus, tellus
eget condimentum
Rhoncus,...</p>
```

Formato visual

Flotantes (III)

The screenshot shows a web browser window with the address bar displaying `file:///H:/ADSII%20(08-09)/Estilos/flotantes.html`. The page content consists of several paragraphs of Lorem Ipsum text. Three images, represented by boxes with an 'X' inside, are used to illustrate different float properties:

- float: left:** A callout box points to a large image on the left side of the page. The text to its right wraps around it. A second callout box explains: "Como el tamaño del flotante excede el tamaño del bloque, también flota en el siguiente párrafo".
- float: right:** A callout box points to a large image on the right side of the page. The text to its left wraps around it.
- float: none (o propiedad no establecida):** A callout box points to a large image centered at the bottom of the page. The text continues below it without wrapping.

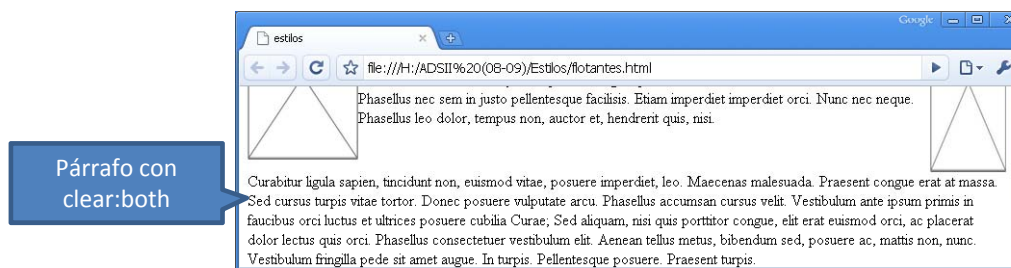
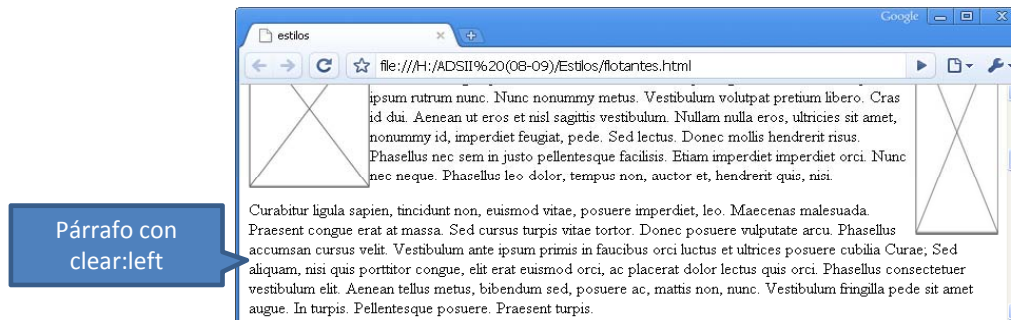
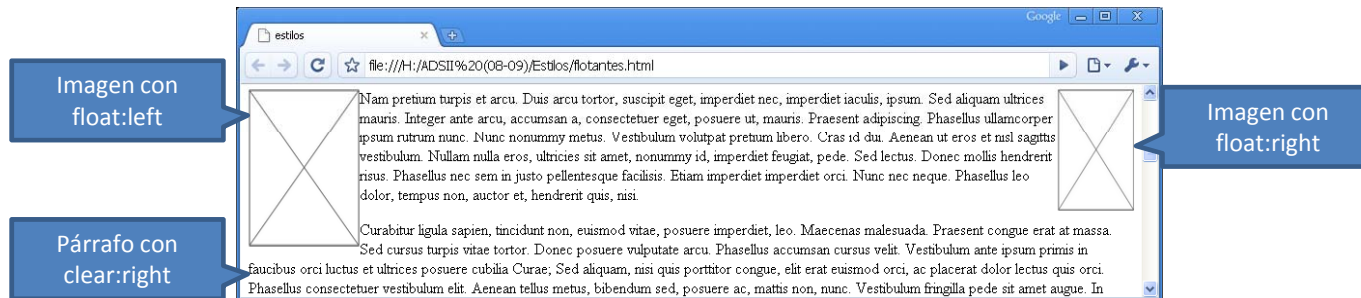
Formato visual

Flotantes (IV)

- ❑ Los elementos siguientes a un flotante fluyen por uno de sus márgenes hasta quedar alineados.
- ❑ Este comportamiento se puede controlar con la propiedad `clear`.
 - Indica cual de los lados no puede quedar adyacente al elemento flotante anterior.
 - Puede tomar los valores: `both`, `right`, `left` o `none`.

Formato visual

Flotantes (V)



Formato visual:

Ancho del contenido

- ❑ La propiedad `width` especifica el ancho del contenido de una caja definida a nivel de bloque.
 - Acepta como valores una medida, un porcentaje, `auto` o `inherit`.
 - Normalmente el ancho total de la caja será la suma de la propiedad `width`, los márgenes izquierdo y derecho, los rellenos izquierdo y derecho y los bordes izquierdo y derecho.
- ❑ Las propiedades `max-width` y `min-width` permiten delimitar un ancho máximo y mínimo de la caja.
 - Si el tamaño del acceso visual es menor que la anchura mínima, la aplicación de usuario deberá mostrar algún sistema de desplazamiento.

Formato visual: Alto del contenido

- ❑ La propiedad `height` especifica el alto del contenido de una caja definida a nivel de bloque.
 - Acepta como valores una medida, un porcentaje, `auto` o `inherit`.
 - Los valores de alto `auto` o `100%` especificarán se establecerán a la altura total del acceso visual (ventana del navegador).
- ❑ Las propiedades `max-height` y `min-height` permiten delimitar un ancho máximo y mínimo de la caja.
 - Si el tamaño del acceso visual es menor que la anchura mínima, la aplicación de usuario deberá mostrar algún sistema de desplazamiento.