

Fundamentos de la Interacción Persona-Ordenador



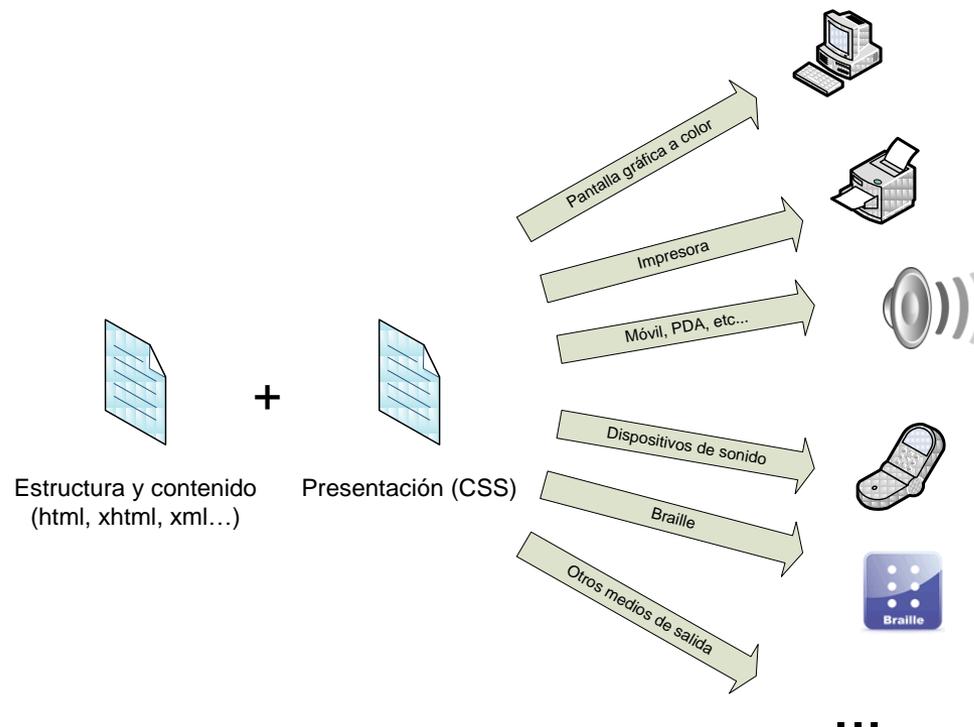
6. CSS

Luis Rodríguez Baena (luis.rodriguez@upsam.es)

Universidad Pontificia de Salamanca
Escuela Superior de Ingeniería y Arquitectura

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 en un medio determinado.



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.
- Se debe procurar que el contenido de la página se pueda mostrar correctamente y en su totalidad sin aplicar las horas de estilo.

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.
 - Publicada como recomendación el 7 de junio de 2011.
 - 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.
 - ✓ En julio de 2012 ha alcanzado el nivel de recomendación el módulo CSS Color Module Level 3 (www.w3.org/TR/2011/REC-css3-color-20110607/) y Selectors Level 3 (<http://www.w3.org/TR/2011/REC-css3-selectors-20110929/>).

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

```
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
18 </head>
19 <body>
20 <h1>Titulo 1</h1>
21 <p style="margin-left:5em;">Lorem ipsum dolor sit amet, consectetur
22 adipiscing
23 elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque
24 penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam
25 felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat
```

Enlace con una hoja de estilos externa

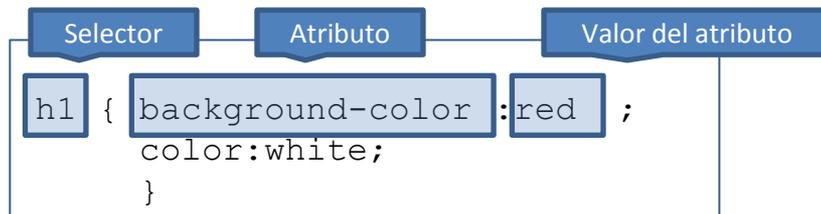
Hoja de estilos incrustada

Estilos en línea

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.
 - ✓ Las hojas de estilo no son sensibles a mayúsculas, excepto aquellas partes que no controla CSS (valores de atributos `id` y `class`, nombres de fuentes, URL...)

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 puede 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 mediante cualquiera de estas dos sintaxis:

```
@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.
- Cuando un mismo elemento se repite en varias especificaciones de estilo hay que determinar cual es el estilo definitivo que va a tener el elemento.
- 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.
- Además, 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.
 - ✓ Un estilo que se aplique al elemento `h1` será más específico que el que se aplique al elemento `body`.
 - Si dos reglas tienen el mismo nivel de importancia, origen y especificidad, la última en ser especificada será la de máxima prioridad.
 - ✓ Las hojas de estilo importadas se consideran que están especificadas antes que 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 el valor de una propiedad no heredada.
 - ✓ Cuando se asigna una propiedad a un elemento, su valor sustituye al valor heredado.
 - ✓ Los elemento 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 pixel, 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.
- ❑ Valor real.
 - Puede que el valor resultante no pueda ser utilizado por limitaciones técnicas del navegador.
 - ✓ El valor real se ajustará los 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.
 - ✓ Es más recomendable utilizar medidas 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 `%`.

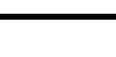
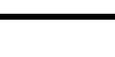
❑ URL.

- `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. Los colores básicos son:
 - aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, 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, vvv, aaa)`.
 - ✓ Mediante porcentajes sobre la cantidad de rojo, verde y azul.
 - `rgb(rrr%, vvv%, aaa%)`.
 - ✓ La especificación CSS 2.1 también recoge identificadores para los colores del sistema utilizados en la interfaz de usuario (`ActiveBorder`, `ButtonFace`, `CaptionText`, etc.) que en CSS 3 están desaconsejados.

Named	Numeric	Color name	Hex rgb	Decimal
		<i>black</i>	#000000	0,0,0
		<i>silver</i>	#C0C0C0	192,192,192
		<i>gray</i>	#808080	128,128,128
		<i>white</i>	#FFFFFF	255,255,255
		<i>maroon</i>	#800000	128,0,0
		<i>red</i>	#FF0000	255,0,0
		<i>purple</i>	#800080	128,0,128
		<i>fuchsia</i>	#FF00FF	255,0,255
		<i>green</i>	#008000	0,128,0
		<i>lime</i>	#00FF00	0,255,0
		<i>olive</i>	#808000	128,128,0
		<i>yellow</i>	#FFFF00	255,255,0
		<i>navy</i>	#000080	0,0,128
		<i>blue</i>	#0000FF	0,0,255
		<i>teal</i>	#008080	0,128,128
		<i>aqua</i>	#00FFFF	0,255,255

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, kilohertzio.
- ❑ **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.

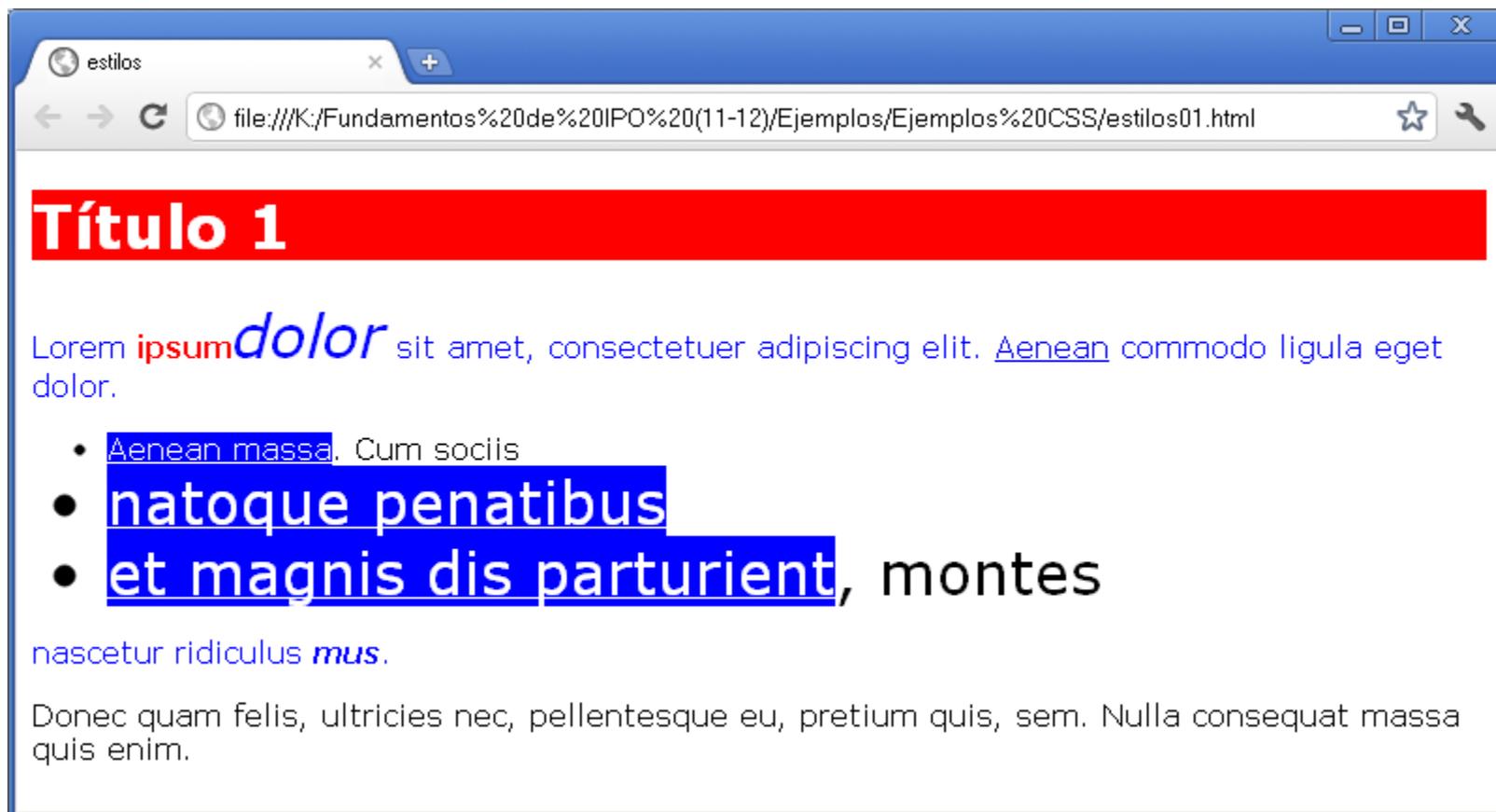
Selectores

- ❑ Se utilizan para identificar el elemento sobre el que se aplicará el estilo.
- ❑ Selector universal.
 - Se utiliza el asterísco (*) 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 esté inmediatamente después de un elemento $E1$.
- ❑ Si existen varios selectores con las mismas declaraciones se pueden agrupar separándolos por comas.
 - La indicación $E1, E2 > E3, E4 E5$ se aplicaría a todos los elementos $E1$, a los elementos $E3$ que sean hijos directos de $E2$ y a los elementos $E5$ que sean descendientes de $E4$.

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 sean hijos directos 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>. Cum sociis</li>
  <li><a href="www.colimbo.net">natoque penatibus</a> </li>
  <li><a href="www.colimbo.net">et magnis dis parturient</a>, montes </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;}`. Selecciona los elementos cuyo atributo `class` sea igual a ejemplo y el lenguaje sea francés.
 - `[atrib ~ =valor]`
 - ✓ Selecciona los elementos que en el atributo `atrib` contengan la palabra `valor`.
 - `[attrib|=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 atributo (II)

```
<style type="text/css">
  * {font-family:Verdana, Arial, Helvetica, sans-serif}
  /* Todos los elementos h1 que tengan el atributo title ... */
  h1[title] {color:blue}
  /* Todos los elementos de la clase ejemplo... */
  [class="ejemplo"] {font-family:"Courier New", Courier, monospace}
  /* Todos los elementos de la clase ejemplo que tengan el atributo lang="en" ... */
  [class="ejemplo"][lang="en"] {background-color:yellow}
  /* Todos los elementos span en los que el contenido de la clase lang empiece por "en-" */
  span[lang|="en"] {color:white;background-color:black}
</style>
</head>

<body>
<h1>Lorem ipsum dolor sit</h1>
<p class="unaclase">amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor.</p>
<h1 title="Aenan massa">Aenan massa</h1>
<p>Cum sociis natoque penatibus et magnis <span class="unaclase" lang="en-US">dis parturient
montes</span>, nascetur ridiculus mus. Donec quam felis, ultricies nec,
<span class="ejemplo">pellentesque eu</span>, pretium quis, sem. Nulla consequat massa quis enim. </p>
<p class="ejemplo" lang="en">Donec pede justo, ...ullamcorper ultricies nisi. Nam eget dui.</p>
```

Selectores

Selectores de atributo (III)



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 (`.`).
 - Selecciona todos los elementos de la clase `centrado`.

```
.centrado {text-align:center;}
```
 - Selecciona los párrafos de la clase `rojo`.

```
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">Ae
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,<
</li>
</ul>
<p>nascetur <strong class="rojo">ridiculus</strong> mus. Donec quam
ultricies nec,
pellentesque eu, pretium quis, sem. Nulla consequat massa quis er
</div>

<p>Donec pede justo, fringilla vel, aliquet nec, vulputate eget,
enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Null
felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. V
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.

```
#menunavegacion {border-color:blue;}  
#menu-izq a {text-decoration: none;}  
h1#titulolibro {font-size: 200%;}
```
- ❑ Aunque CSS no es sensible a mayúsculas, los valores de `id` y de `class` en XHTML **si son sensibles a mayúsculas**.

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 su nombre, atributos o contenidos.
 - ✓ Por ejemplo, el primer hijo de un elemento.
 - ✓ Pseudo-clases dinámicas.
 - Un elemento puede pertenecer o dejar de pertenecer a una pseudo-clase a medida que el usuario interactúa con el documento.
 - Por ejemplo, cuando el ratón pasa por encima de un elemento, cuando un elemento es activo.

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-elementos `:before` y `:after`.
 - Se verán más adelante en el apartado “Contenido generado”.
- ❑ Pseudo-clase `:first-child`.
 - Hace referencia al primer elemento hijo del padre de un nodo 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-clases (III)

```
<style>
  * {font-family: Verdana, Arial, Helvetica, sans-serif;}
  p {text-indent:3em}
  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;
                                padding:0;}

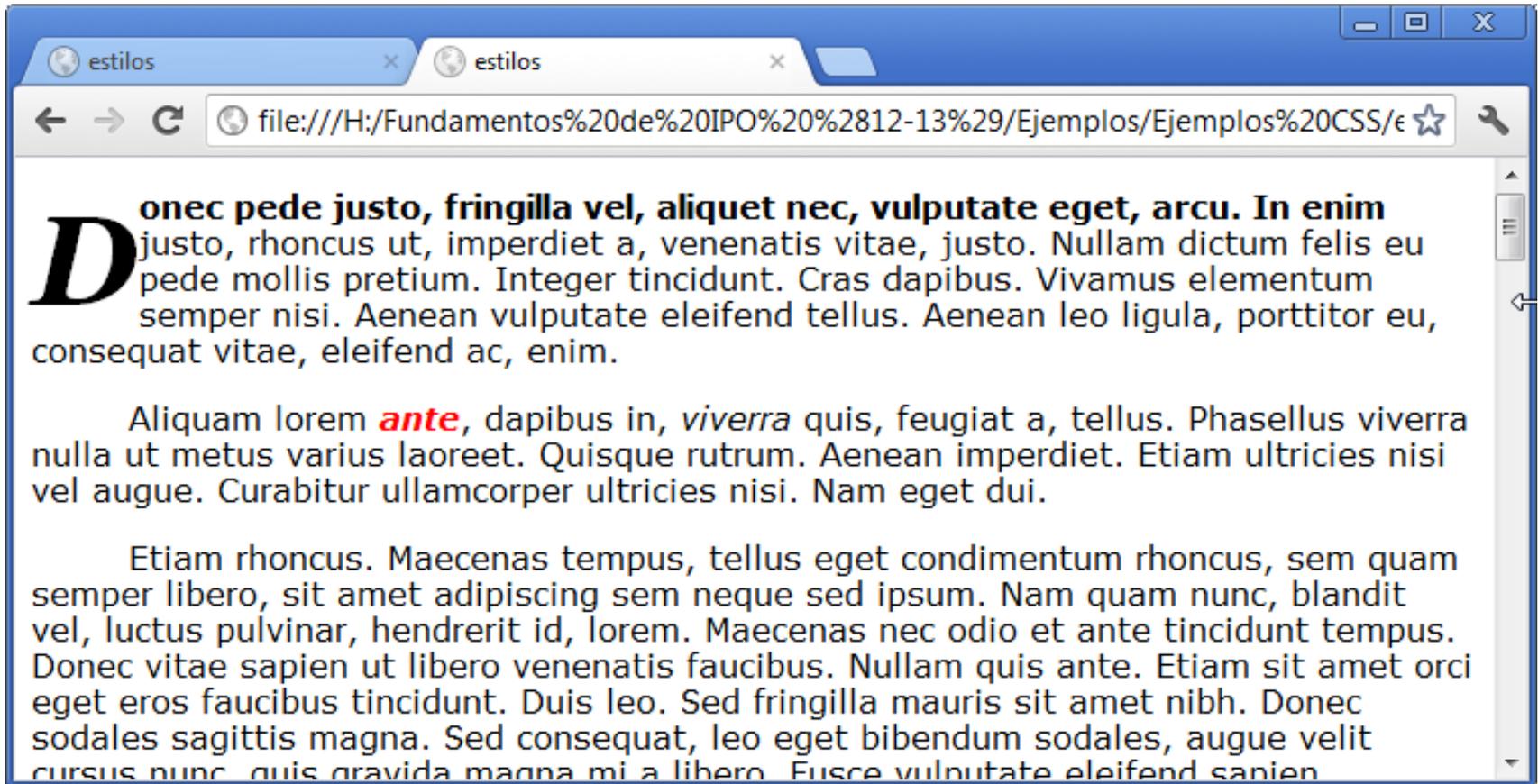
  /*Selecciona el elemento p que sea primer hijo de cualquier elemento:
  el primer párrafo del documento */
  p:first-child {text-indent:0}
  /*Selecciona el elemento em que sea el primer hijo de cualquier elemento:
  en el segundo párrafo la palabra ante.
  em:first-child {color:red;
                  font-weight:bold;}

</style>
</head>
<body>
<p class="primerparrafo">Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim
...
consequat vitae, eleifend ac, enim. </p>

<p>Aliquam lorem <em>ante</em>, dapibus in, <em>viverra</em>
...
ullamcorper ultricies nisi. Nam eget dui.</p>
```

Selectores

Pseudo-elementos y pseudo-clases (IV)



Selectores

Pseudo-elementos y pseudo-clases (V)

❑ 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.
- Aunque se suelen utilizar para enlaces, se pueden utilizar para cualquier otro elemento que pueda detectar esas acciones.
 - ✓ Por ejemplo, cualquier elemento puede detectar si el ratón pasa por encima.

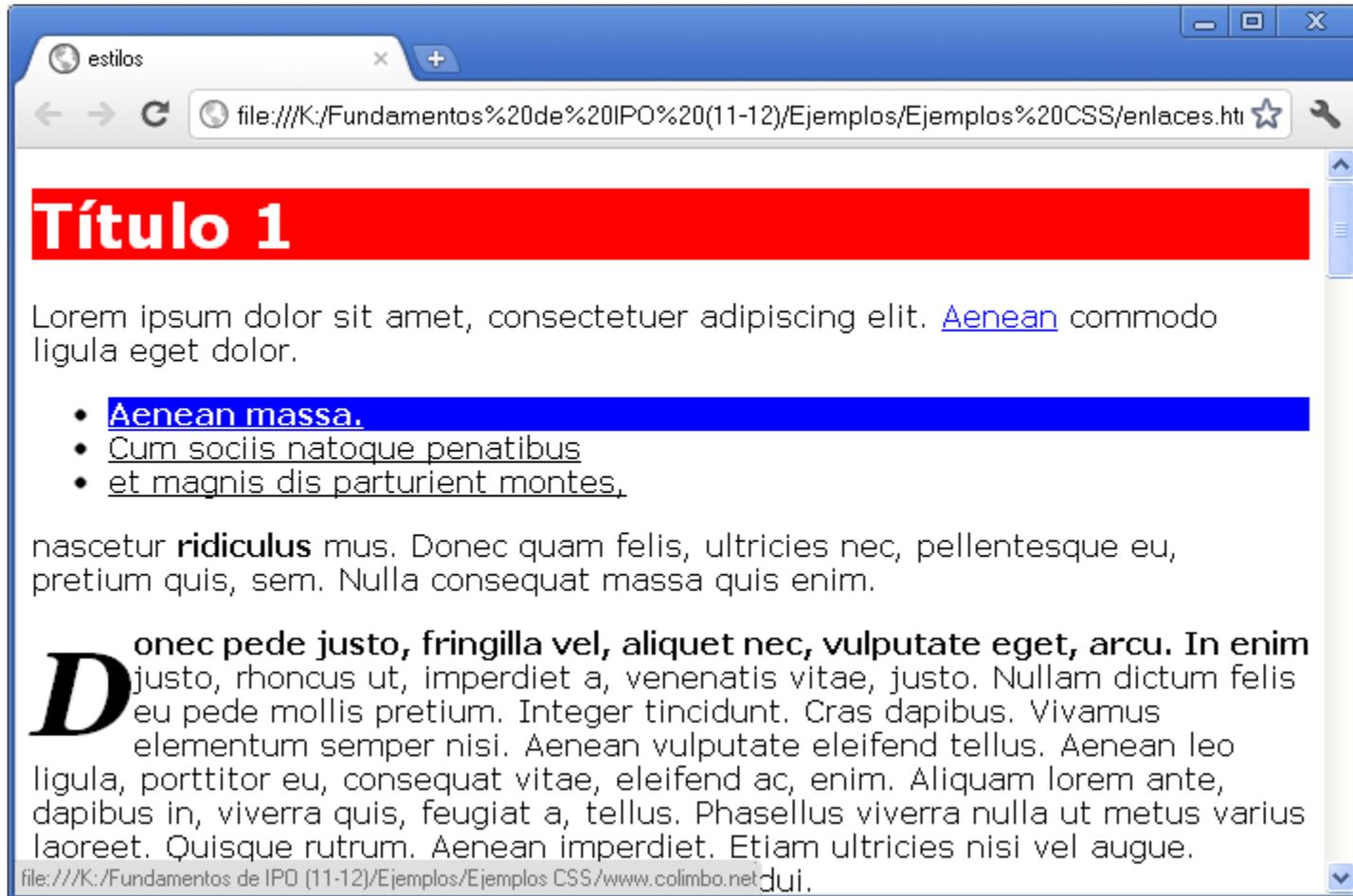
Selectores

Pseudo-elementos y pseudo-clases (VI)

```
<style type="text/css">
...
a:hover { font-weight:bold;}
a:focus {color:red;}
a:active {color:green; font-weight:bold;}
p.resaltar:hover {color:white;
                  background-color:black}
li a {color:black; display:block}
li a:active, li a:hover, li a:focus
    {background-color:blue;
      color:white;}
</head>
<body>
<h1>T&iacute;tulo 1</h1>
<p>Lorem ipsum dolor sit amet, consectetur
  adipiscing</span> elit. <a href="http://www.colimbo.net">Aenean</a> commodo ligula eget dolor.
</p>
<ul id="menunavegacion">
  <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 class="resaltar">nascetur <strong>ridiculus</strong> mus. Donec quam felis, ultricies nec,
pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim.</p>
...
```

Selectores

Pseudo-elementos y pseudo-clases (VII)



Colores y fondos

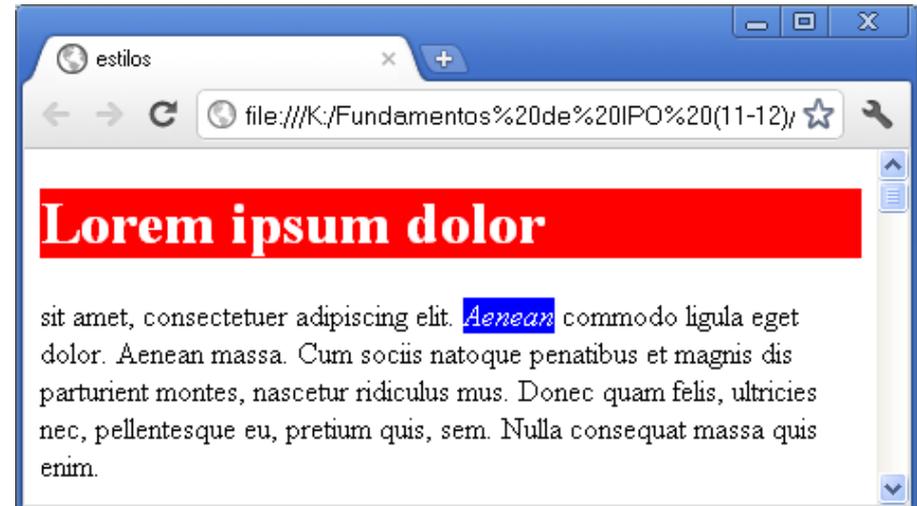
❑ Propiedad `color`.

- Permite establecer el color de primer plano.
- El valor sería una especificación de color o el valor `inherit` (el color de primer plano si 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, o el valor `inherit` (la propiedad no se hereda) 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(0%,0%,100%);
}
```



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 se 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`.

Colores y fondos (III)

- ❑ Inicialmente la imagen se colocará en la esquina superior izquierda del área de padding (relleno, ver más adelante) del bloque dónde está definida.
 - Se cambia con la propiedad `background-position`.
 - La posición se establece por:
 - ✓ Valor `inherit`.
 - ✓ Uno o dos...
 - Porcentajes.
 - Medidas (`em`, `ex`, `px`, `in`, `cm`, `mm`, `pt`, `0 pc`).
 - Si sólo aparece uno, hará referencia a la posición horizontal, la vertical será del 50%.
 - ✓ Los valores...
 - `top left`, `top`, `top center`, `right top`, `left`, `left center`, `center`, `right`, `right center`, `bottom left`, `bottom`, `bottom center`, `right bottom`.
- ❑ La propiedad `background` permite establecer los valores del color de fondo, la imagen, la posición de la imagen, el anclaje y la repetición.

`background: color de fondo imagen repetición attachment posicion`

 - Los valores se meten directamente sin comas entre ellos.

Colores y fondos (IV)

```
.imagenrelleno {background-image:url("lunares.png");
}

.banner {padding-top:90px;
         background: url("banner.png") repeat-x fixed;
}

...
<body class="imagenrelleno">
...
<p class="banner">
...
```



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
```

 - Courier New o Time New Roman van entre comillas porque tienen espacios en blanco.
 - ✓ 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`.

❑ Manejo de los espacios en blanco.

- De forma predeterminada:
 - ✓ Si aparecen varios espacios en blanco, se juntan en un único espacio.
 - ✓ Si aparece uno o más espacios en blanco al final de una línea se produce un salto de línea.
- La propiedad `white-space` permite modificar este comportamiento.
- Valores:
 - ✓ `normal`.
 - ✓ `nowrap`. Suprime los saltos de línea.
 - ✓ `pre`. No junta los espacios en blanco y suprime los saltos de línea.
 - ✓ `pre-wrap`. No junta los espacios en blanco, respetando los saltos de línea.

Listas

❑ Propiedad `list-style-type`.

- 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`.

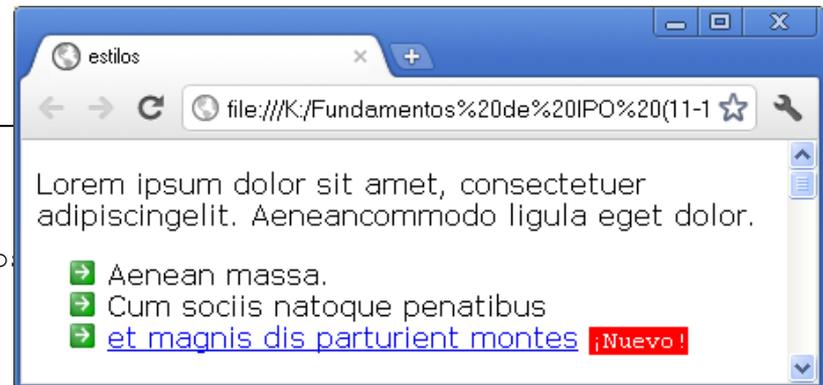
- Especifica la posición de la viñeta.
- Puede tomar los valores `inside`, `outside`, o `inherit`.
 - ✓ Con `outside`, la viñeta queda fuera de la caja de texto.
 - ✓ Con `inside`, la viñeta forma parte de la caja de texto.

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

Contenido generado

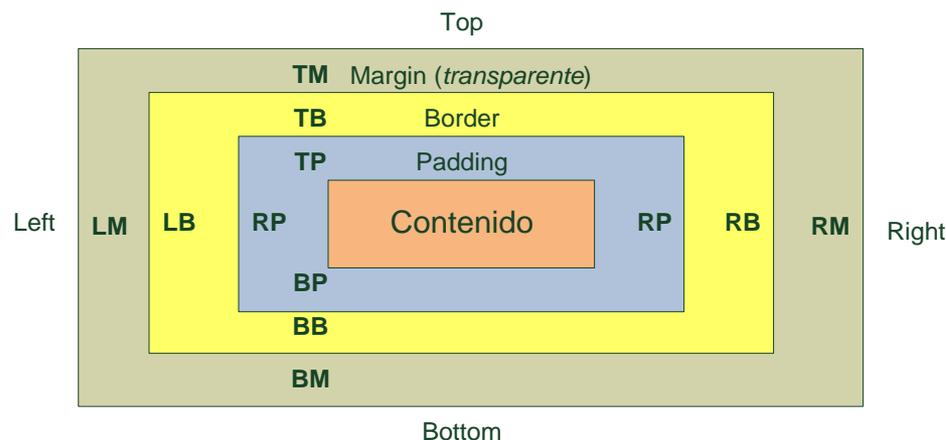
- ❑ CSS permite incluir contenido en los elementos mediante las pseudo-classes `:before` y `:after` y la propiedad `content`.
 - `:before` y `:after` indican la ubicación del contenido (antes o después del elemento seleccionado por el selector).
 - `content` indica el contenido que se va a incluir.
 - ✓ Los valores pueden ser:
 - `normal`. No añade nada.
 - Una cadena (entrecomillada).
 - Un url.

```
.nuevo:after{content:"¡Nuevo!";
background-color:red;
color:white;
font-family:"Courier New", Courier, mono;
font-size:smaller;
}
...
<li class="nuevo">
  <a href="#">
    et magnis dis parturient montes
  </a>
</li>
```



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 *límite*.
 - 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 estilo 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.

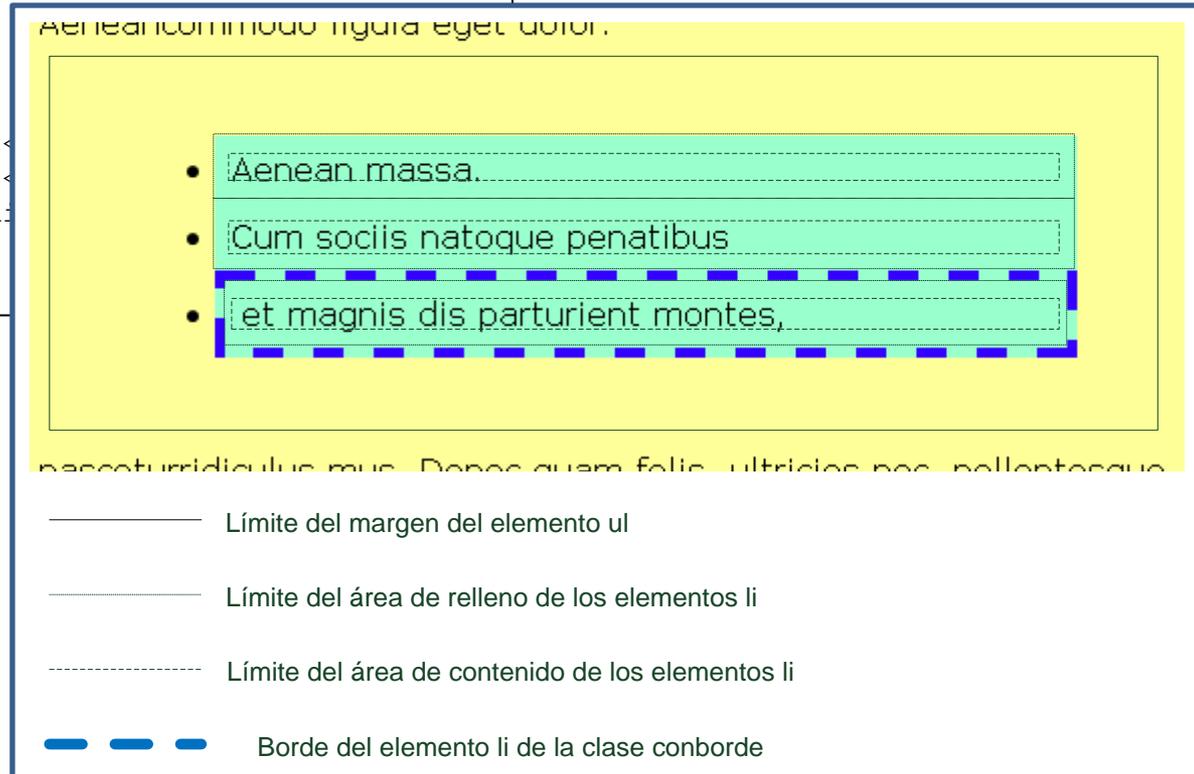
- Propiedades `border-top`, `border-right`, `border-bottom` y `border-left`.
 - ✓ Permiten definir el ancho, el estilo y el color de cada borde.
 - Recibe los valores de ancho, estilo y color separados por un espacio en blanco.
- Propiedad `border`.
 - ✓ Permite definir el ancho, el estilo y el color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

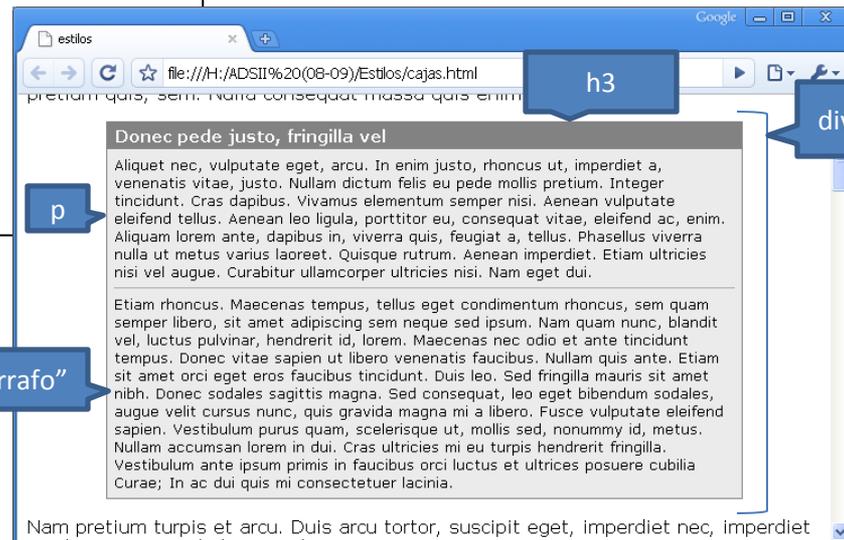
<p>...</p>
<ul>
  <li class="flecha">Aenean massa.</li>
  <li>Cum sociis natoque penatibus</li>
  <li class="conborde">et magnis di</li>
</ul>
<p>...</p>
```



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.
 - Si el acceso visual se redimensiona, las aplicaciones (navegadores) pueden cambiar la composición del documento.
- ❑ 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.
 - El ancho y alto de los bloques de contención se pueden establecer con las propiedades `width` y `height` del elemento.
 - ✓ Puede tomar como valores una unidad de medida, un porcentaje o el valor `auto`.
 - Las aplicaciones deben ofrecer mecanismos de desplazamiento (por ejemplo, barras de scroll) cuando el acceso visual es más pequeño que el área de contención.

Formato visual

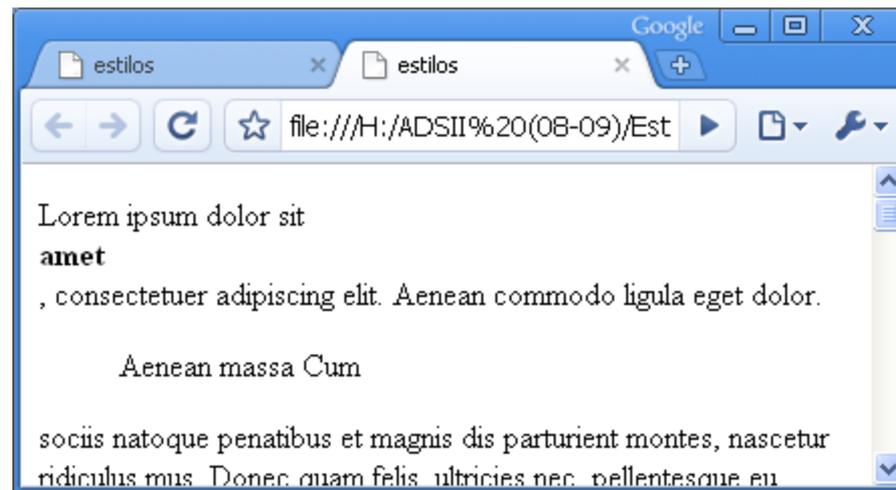
Tipos de bloques y propiedad display

- ❑ 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

Tipos de bloques y propiedad display (III)

❑ Ejemplo: menú de navegación

```
* {font-family:Verdana, Arial, Helvetica, sans-serif;
}
.logo {font-family:Verdana, Arial, Helvetica, sans-serif;
font-size:3em;
font-style:italic;
font-weight:bold;
font-style:italic;
color:#B7B7B7;}
.navegacionprincipal {background-color:#F6F6F6;
border-top:#B7B7B7 thin solid;
border-bottom:#B7B7B7 thin solid;
margin-top:0.5em;
padding:0.5em;}
.navegacionprincipal ul{list-style-type:none;
padding:0.5em;
display:inline;}
.navegacionprincipal li{display:inline;}
.navegacionprincipal li+li:before{content:"| "};
...
<div class="logo">Lorem ipsum </div>
<div class="navegacionprincipal">
<ul>
<li><a href="#">Inicio</a></li>
<li><a href="#">Productos</a></li>
<li><a href="#">Soporte</a></li>
<li><a href="#">Contacto</a></li>
</ul>
</div>
<p>Dolor sit amet, consectetur adipiscing elit. ...
```

Formato visual

Tipos de bloques y propiedad display (IV)



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 elemento flotante.
 - 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`, 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`.
- No tienen efecto con el esquema de posicionamiento `static`.

Formato visual

Flujo normal

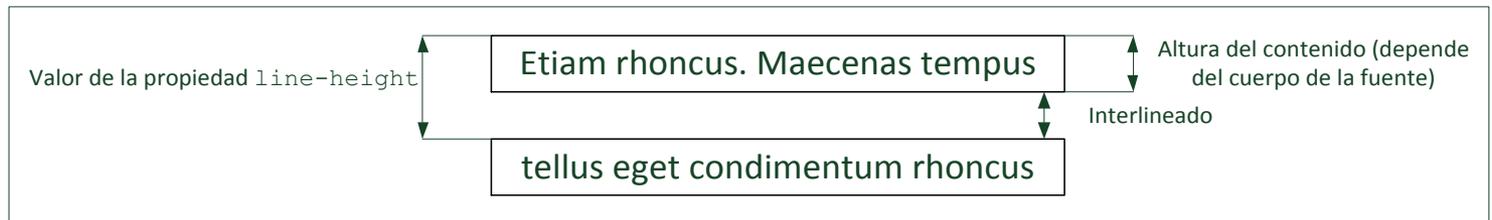
- ❑ Cajas de bloque.
 - Se colocan una después de otra verticalmente.
 - La distancia vertical entre cajas hermanas la determina la propiedad `margin`.
 - ✓ Los márgenes verticales de cajas adyacentes se funden para formar uno sólo.
- ❑ Cajas de línea.
 - Se colocan horizontalmente a partir de la parte 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

Flujo normal (II)

❑ Propiedad `line-height`.

- La altura de una línea se calcula a partir de la altura del contenido de los elementos que la contienen.
 - ✓ La altura de una línea dentro de un elemento de bloque, por ejemplo, una línea de un elemento `p`, será la altura del mayor de los elementos.
 - ✓ La diferencia entre la altura del contenido y el alto de línea será el *interlineado* del elemento.
 - La mitad del interlineado es llamado medio interlineado.
 - Las aplicaciones de usuario centran el contenido de una caja de línea, añadiendo medio interlineado por encima y medio por abajo.
- Valores: `normal`, número, medida, porcentaje o `inherit`.



❑ Propiedad `vertical-align`.

- Indica la alineación del contenido de un elemento dentro de una línea.
- Valores:
 - ✓ Los siguientes valores sólo tienen significado con respecto a un elemento a nivel de línea padre, o a un elemento a nivel de bloque padre:
 - `baseline`, `middle`, `sub`, `super`, `text-top`, `text-bottom`.
 - Medida o porcentaje, positivo (sube sobre la línea base) o negativo (baja sobre la línea base).

Formato visual

Flujo normal (III)

- ❑ Cajas de línea (azul) y de bloque (negro).
 - La palabra "adipiscing" tiene un tamaño de 3em.
 - La palabra "elit" tiene el valor de vertical-align a top.
 - La palabra "Aeneam" tiene un tamaño de 2em.



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 callout boxes are overlaid on the page:

- float: right**: A callout box pointing to a square placeholder on the right side of the first paragraph.
- float: left**: A callout box pointing to a square placeholder on the left side of the second paragraph. The text of this callout is: "Como el tamaño del flotante excede el tamaño del bloque, también flota en el siguiente párrafo".
- float: none (o propiedad no establecida)**: A callout box pointing to a square placeholder on the left side of the third paragraph.

Other callouts include "float: left" pointing to a square placeholder on the left side of the second paragraph, and another "float: left" pointing to a square placeholder on the left side of the third paragraph.

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)

Imagen con float:left

Imagen con float:right

Párrafo con clear:right

Nam pretium turpis et arcu. Duis arcu tortor, suscipit eget, imperdiet nec, imperdiet iaculis, ipsum. Sed aliquam ultrices mauris. Integer ante arcu, accumsan a, consectetur eget, posuere ut, mauris. Praesent adipiscing. Phasellus ullamcorper ipsum rutrum nunc. Nunc nonummy metus. Vestibulum volutpat pretium libero. Cras id dui. Aenean ut eros et nisl sagittis vestibulum. Nullam nulla eros, ultricies sit amet, nonummy id, imperdiet feugiat, pede. Sed lectus. Donec mollis hendrerit risus. Phasellus nec sem in justo pellentesque facilisis. Etiam imperdiet imperdiet orci. Nunc nec neque. Phasellus leo dolor, tempus non, auctor et, hendrerit quis, nisi.

Curabitur ligula sapien, tincidunt non, euismod vitae, posuere imperdiet, leo. Maecenas malesuada. Praesent congue erat at massa. Sed cursus turpis vitae tortor. Donec posuere vulputate arcu. Phasellus accumsan cursus velit. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed aliquam, nisi quis porttitor congue, elit erat euismod orci, ac placerat dolor lectus quis orci. Phasellus consetetur vestibulum elit. Aenean tellus metus, bibendum sed, posuere ac, mattis non, nunc. Vestibulum fringilla pede sit amet augue. In

Párrafo con clear:left

ipsum rutrum nunc. Nunc nonummy metus. Vestibulum volutpat pretium libero. Cras id dui. Aenean ut eros et nisl sagittis vestibulum. Nullam nulla eros, ultricies sit amet, nonummy id, imperdiet feugiat, pede. Sed lectus. Donec mollis hendrerit risus. Phasellus nec sem in justo pellentesque facilisis. Etiam imperdiet imperdiet orci. Nunc nec neque. Phasellus leo dolor, tempus non, auctor et, hendrerit quis, nisi.

Curabitur ligula sapien, tincidunt non, euismod vitae, posuere imperdiet, leo. Maecenas malesuada. Praesent congue erat at massa. Sed cursus turpis vitae tortor. Donec posuere vulputate arcu. Phasellus accumsan cursus velit. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed aliquam, nisi quis porttitor congue, elit erat euismod orci, ac placerat dolor lectus quis orci. Phasellus consetetur vestibulum elit. Aenean tellus metus, bibendum sed, posuere ac, mattis non, nunc. Vestibulum fringilla pede sit amet augue. In turpis. Pellentesque posuere. Praesent turpis.

Párrafo con clear:both

Phasellus nec sem in justo pellentesque facilisis. Etiam imperdiet imperdiet orci. Nunc nec neque. Phasellus leo dolor, tempus non, auctor et, hendrerit quis, nisi.

Curabitur ligula sapien, tincidunt non, euismod vitae, posuere imperdiet, leo. Maecenas malesuada. Praesent congue erat at massa. Sed cursus turpis vitae tortor. Donec posuere vulputate arcu. Phasellus accumsan cursus velit. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed aliquam, nisi quis porttitor congue, elit erat euismod orci, ac placerat dolor lectus quis orci. Phasellus consetetur vestibulum elit. Aenean tellus metus, bibendum sed, posuere ac, mattis non, nunc. Vestibulum fringilla pede sit amet augue. In turpis. Pellentesque posuere. Praesent turpis.

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 (barras de scroll).

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).
 - ✓ No especifican el alto total de la página, sino sólo de la 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.

Formato visual

Capas

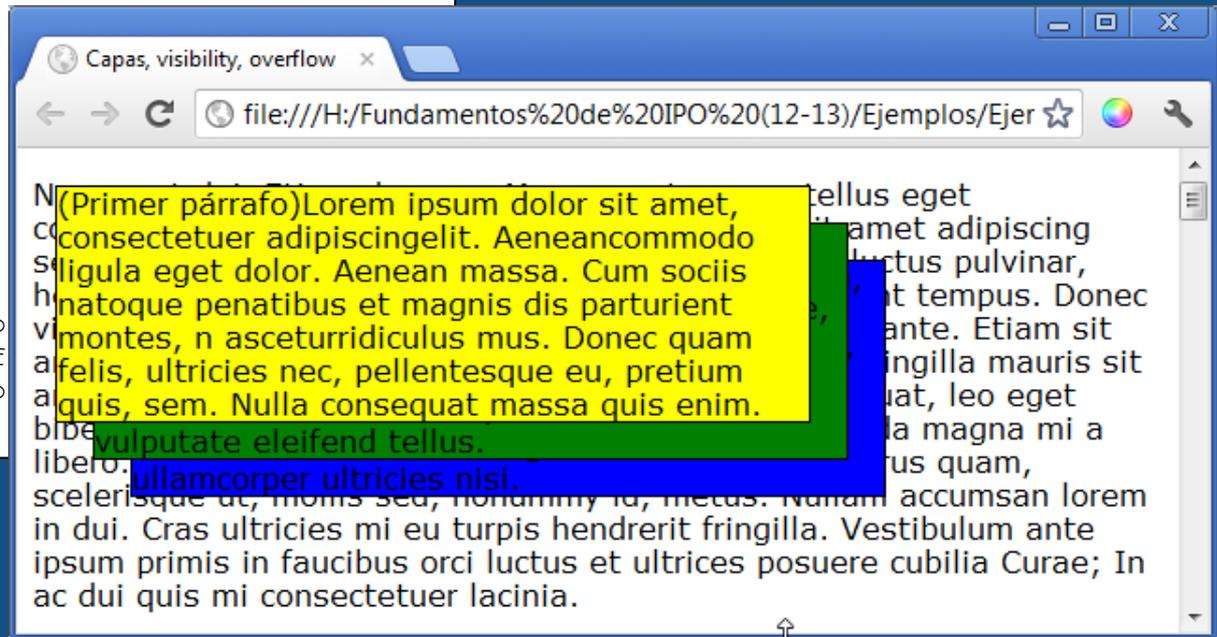
- ❑ Cuando las cajas se posicionan de forma absoluta se pueden superponer unas sobre otras, formando una pila.
- ❑ Normalmente el navegador las coloca según el orden del código:
 - El último elemento del código html estará en la parte superior de la pila.
- ❑ La propiedad `z-index` permite modificar este comportamiento.
 - Admite un número entero.
 - Cuando mayor es el número, más cerca estará la capa del usuario, ocultando aquellas que tengan un número inferior.
 - La propiedad sólo tiene efecto si el elemento está posicionado "posicionado".
 - ✓ La propiedad `position` deberá tener un valor distinto de `static`.

Formato visual

Capas (II)

```
* {font-family: Verdana, Arial, Helvetica, sans-serif;
    background-color:white;}
div {position:absolute;
    border: solid black 1px;
    width:50%;}
.cajainf {top:20px;
    left:20px;
    z-index:15;
    background-color:yellow;}
.cajamed {top:40px;
    left:40px;
    z-index:10;
    background-color:green;}
.cajasup {top:60px;
    left:60px;
    z-index:5;
    background-color:blue;}
...
<body>
```

```
<div class="cajainf">(Primer párrafo)
<div class="cajamed">(Segundo párrafo)
<div class="cajasup">(Tercer párrafo)
...
```



Formato visual

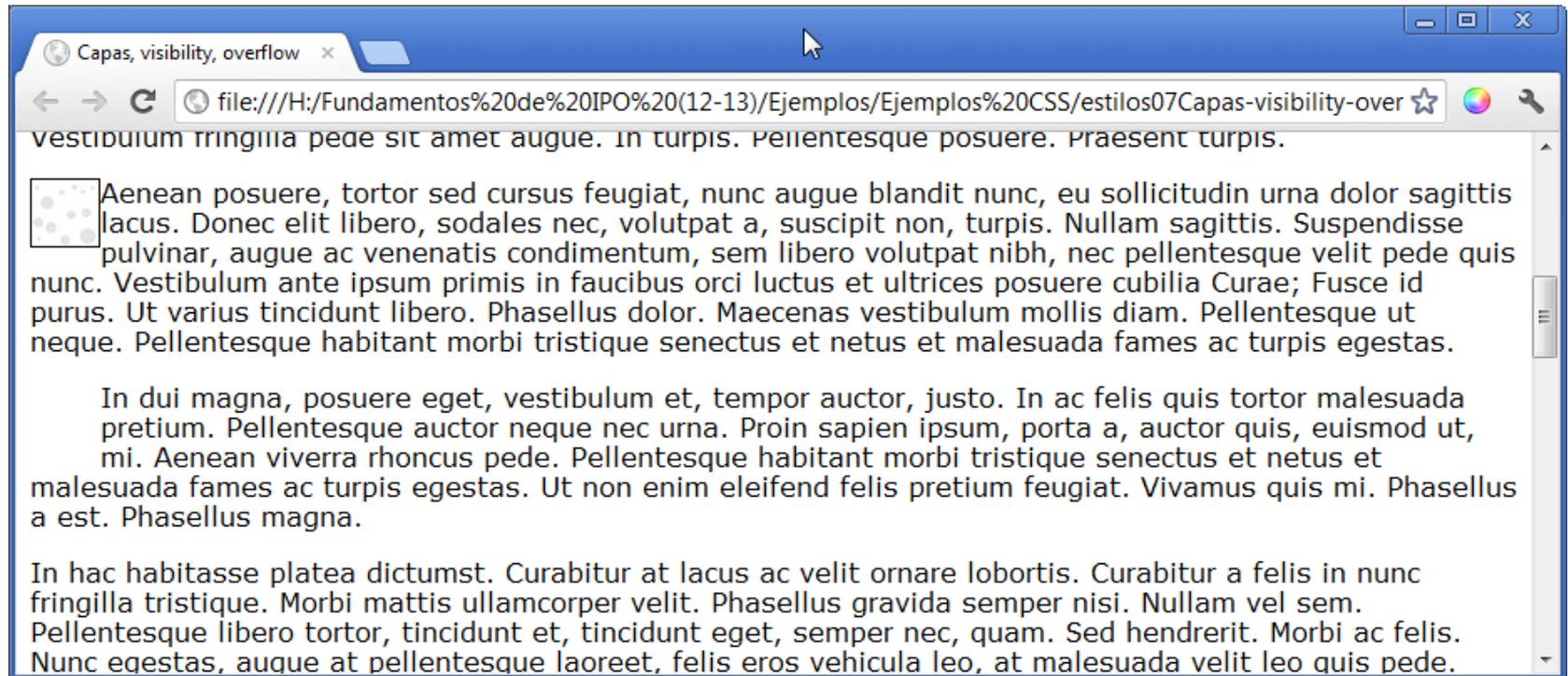
Propiedad visibility

- ❑ Permite controlar la visibilidad del contenido de un bloque.
 - Mientras que la propiedad `display=none`, elimina el bloque del contenido y no ocupa espacio, `visibility` guarda el espacio que ocupa el bloque, pero no muestra su contenido.
- ❑ Puede tomar los valores:
 - `visible`, valor por omisión.
 - `hidden`, oculta el contenido de un bloque.
 - `collapse`, sólo es aplicable a filas, columnas, grupos de filas y grupos de columnas. Oculta los elementos de forma similar a `display=none`.
 - `inherit`, hereda el comportamiento del elemento padre.

Formato visual

Propiedad visibility (II)

```
...  
<p>Aenean posuere, tortor... egestas.</p>  
<p>In dui magna,... magna.</p>  
<p> In hac habitasse ... vulputate lorem.</p>  
...
```



Formato visual

Propiedad overflow

- ❑ En ocasiones, el contenido de un elemento no cabe en el bloque designado para él.
- ❑ Cuando esto ocurre, el contenido se desborda, ocupando saliendo de su bloque y ocupando las cajas adyacentes.
- ❑ La propiedad `overflow` controla el comportamiento del contenido cuando sobrepasa su bloque de contención.
- ❑ Puede tomar los valores:
 - `visible`, valor por omisión. El contenido sobrepasa su bloque de contención.
 - `hidden`, el contenido se corta y no se visualiza el contenido que esté más allá de su bloque.
 - `scroll`, si el contenido se desborda, aparecen mecanismos para desplazarlo.
 - `auto`, se deja que el navegador determine el comportamiento del desbordamiento.

Formato visual

Propiedad overflow (II)

```
.colizq, .colder, .colcen {width:200px;
margin:0 3em 20em 3em;
height:300px;
float:left;
border:1px solid black}
...
<p class="colizq" style="overflow:visible">Morbi nec metus. Phasellus... laoreet nibh.</p>
<p class="colcen" style="overflow:hidden">Fusce convallis metus id ... vel, lacus.</p>
<p class="colder" style="overflow:scroll">Sed augue ipsum, egestas nec, ... non, velit.</p>
```

