

# Introducción a CSS

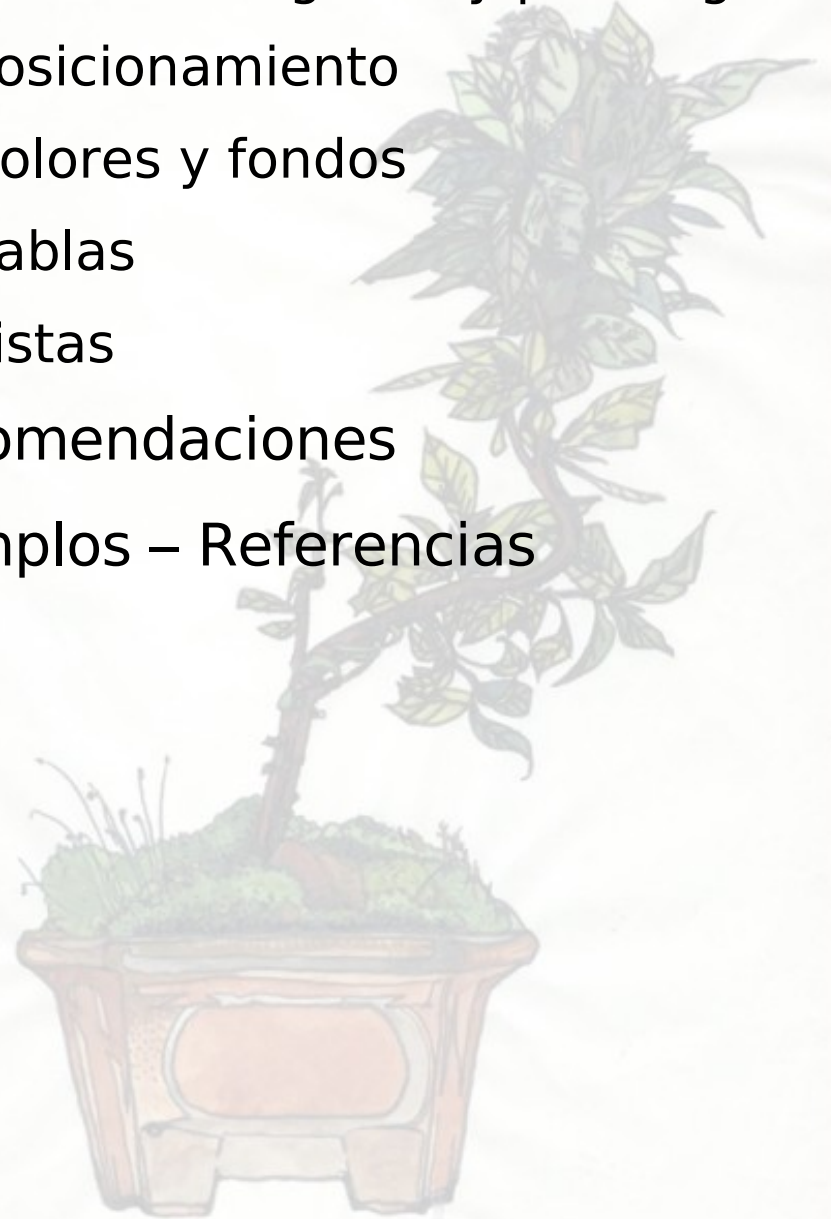
Ana Saiz García  
<ana{arroba}gpul{punto}org>

I Jornada sobre Tecnologías Web y Software Libre  
A Coruña – 12 Julio 2006



# Guión

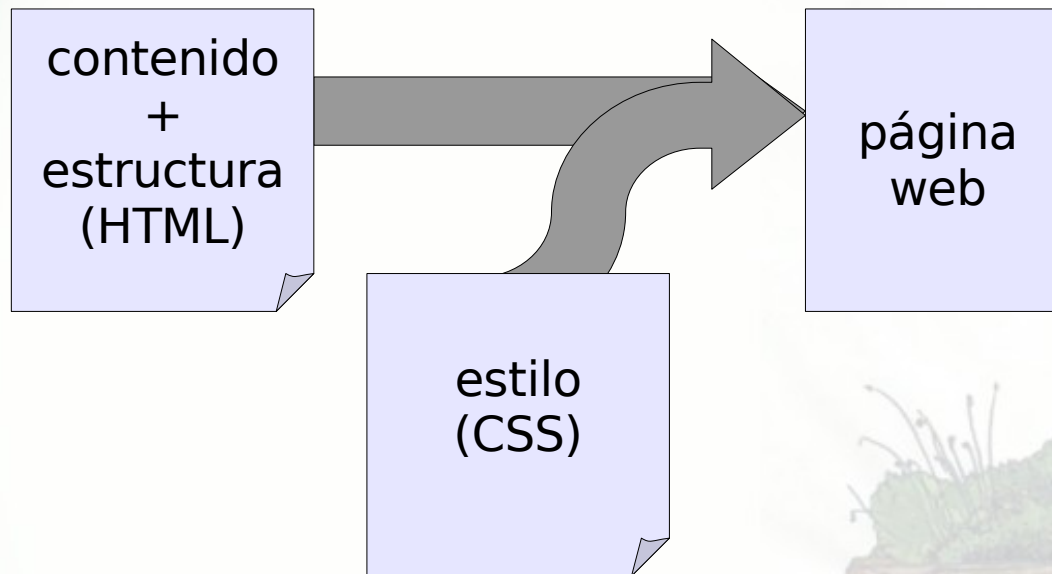
- Introducción
  - ¿Qué es CSS?
  - Ventajas
  - ¿Cómo conjugarlo con HTML?
- Sintaxis básica
  - Elementos
  - Selectores
  - Estructura
  - Unidades
  - Fuentes
  - Texto
- Bordes, márgenes y padding
- Posicionamiento
- Colores y fondos
- Tablas
- Listas
- Recomendaciones
- Ejemplos – Referencias



# Introducción

- **¿Qué es CSS?**

- *Cascading Style Sheets – Hojas de estilo en cascada*
- Estilo: definición de colores, posiciones, fuentes...
- Permiten dar estilo a las páginas Web, separándolo del contenido (HTML)



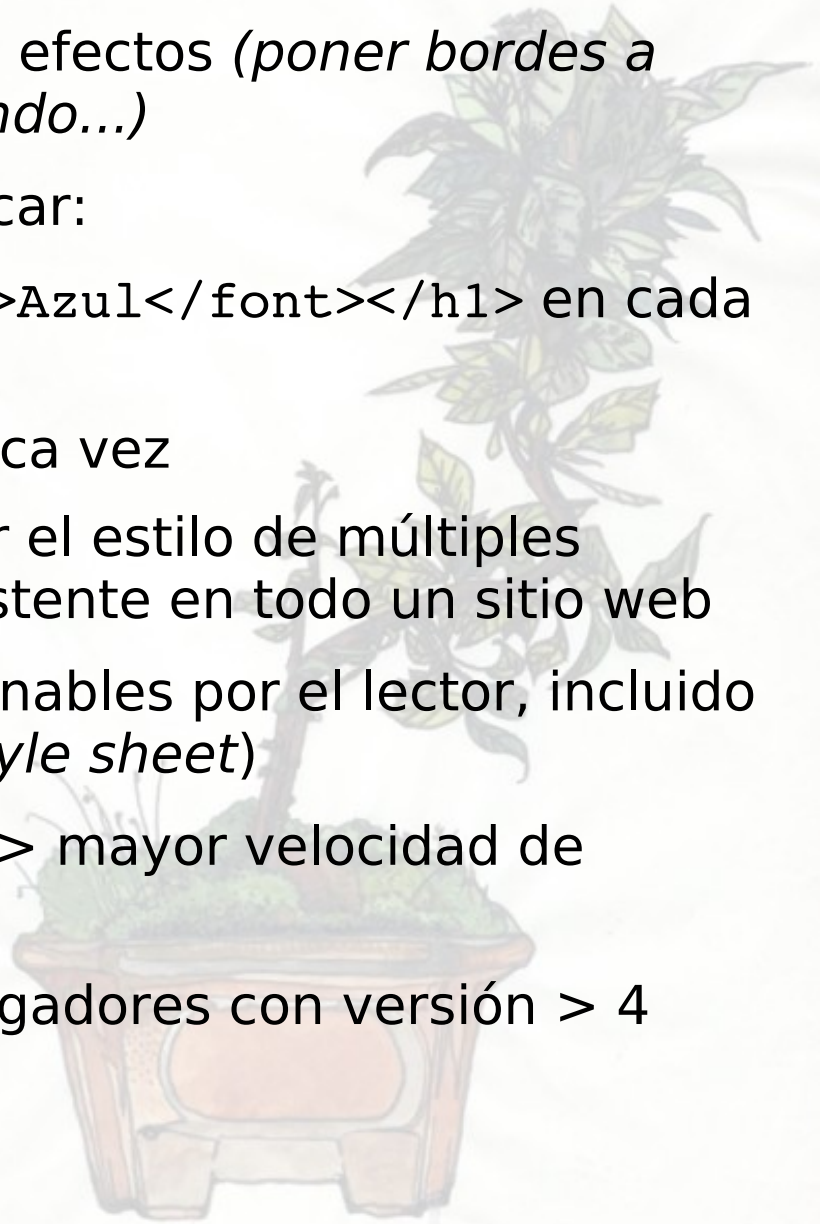
- Estándar del W3C (*Word Wide Web Consortium*)
- Especificaciones: CSS1 -> CSS2 -> CSS2.1 -> CSS3

# Introducción

- **Ventajas**

- Estilos más ricos que HTML, con más efectos (*poner bordes a cualquier elemento, imágenes de fondo...*)
- Más rápido y fácil de definir y modificar:
  - HTML: `<h1><font color="blue">Azul</font></h1>` en cada elemento `<H1>`
  - CSS: `h1 {color: blue;}` una única vez
- Una única hoja de estilos para definir el estilo de múltiples páginas -> mantener un estilo consistente en todo un sitio web
- Flexibilidad: distintos estilos seleccionables por el lector, incluido uno definido por él mismo (*reader style sheet*)
- Reduce tamaño de los documentos -> mayor velocidad de descarga

- **Inconveniente:** sólo lo soportan navegadores con versión > 4



# Introducción

- **¿Cómo conjugarlo con HTML?**

- HTML -> estructura interna, CSS -> estructura visual

```
<html>
  <head>
    <title>Título de la página</title>
    <link rel="stylesheet" type="text/css"
href="estilos1.css" media="all" />
    <style type="text/css">
      @import url(estilos2.css);
      h1 {color: red;}
      body {background: yellow;}
    </style>
  </head>
  <body>
    <h1>Introducción</h1>
    <p style="color: black;">Veamos cómo incluir
estilos con CSS en un documento HTML</p>
  </body>
</html>
```

hoja de estilos externa

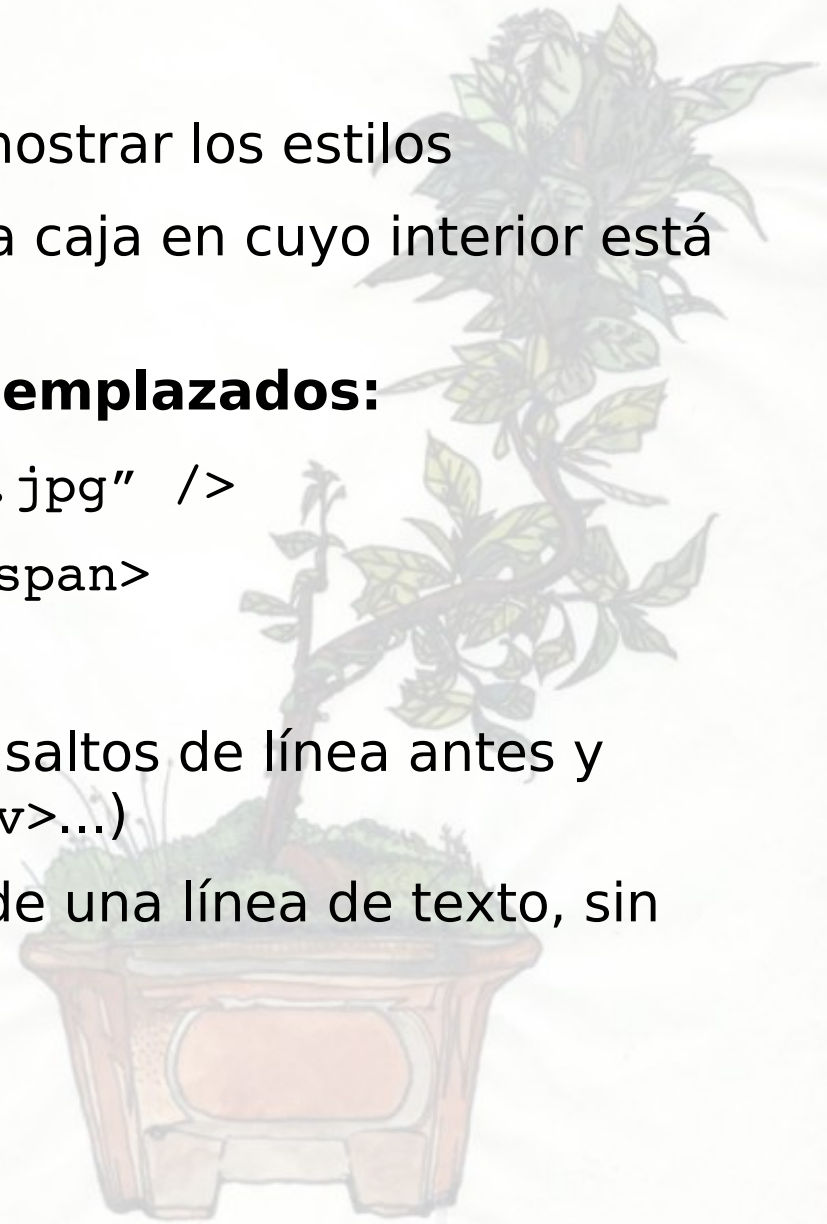
hoja de estilos  
de documento

estilo en línea (*inline*)

# Sintaxis básica

- **Elementos**

- HTML: div, span, p, table...
- CSS se basa en ellos para definir y mostrar los estilos
- Para CSS, cada elemento genera una caja en cuyo interior está el contenido del elemento
- **Elementos reemplazados y no reemplazados:**
  - Reemplazados: ``
  - No reemplazados: `<span>Hola</span>`
- **Elementos de bloque y en línea:**
  - De bloque (*block-level*): caja con saltos de línea antes y después del elemento (`<p>`, `<div>...`)
  - En línea (*line-level*): caja dentro de una línea de texto, sin romper su flujo (`<a>`, `<span>...`)



# Sintaxis básica

- **Selectores**

- Definen qué parte del documento se verá afectada por el estilo

- Reglas:

```
h2 { color: blue; }
```

*selector*                      *declaración*

- Varias declaraciones para un mismo selector: Bloque de declaraciones

```
h2 {color: blue;  
background-color: black;  
font-size: 12pt;}
```

- Algunas propiedades admiten valores formados por varias *keywords* (separadas por espacios)

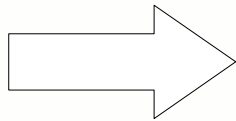
```
h2 {font: italic 12pt sans-serif;}
```

# Sintaxis básica

- **Selectores (II)**

- Se pueden agrupar:

```
h1 {color: blue;}  
h2 {color: blue;}  
h3 {color: blue;}  
h4 {color: blue;}
```



```
h1, h2, h3, h4 {color: blue;}
```

- Selector universal (CSS2): representa a todos los elementos que aparecen en el documento: **\*{color: blue}**

- **Tipos de selectores:**

- Simples -> equivalen al elemento HTML
- Class -> valor del atributo **class** en HTML
- ID -> valor del atributo **id** en HTML
- De atributo (CSS2) -> atributo, sin importar su valor
- Descendientes -> elementos anidados
- Hijos -> elementos descendientes directos de otros

# Sintaxis básica

- **Selectores (y III): tipos de selectores**

Tipo	HTML	CSS
Simple	<code>&lt;p&gt;&lt;/p&gt;</code>	<code>p {color: blue;}</code>
Class	<code>&lt;p class="nota"&gt;&lt;/p&gt;</code>	<code>p.nota {color: blue;}</code>
ID	<code>&lt;p id="nota"&gt;&lt;/p&gt;</code>	<code>#nota {color: blue;}</code>
Atributo	<code>&lt;p class="nota"&gt;&lt;/p&gt;</code>	<code>p[class] {color: blue;}</code>
Descendente	<code>&lt;p&gt;&lt;em&gt;&lt;strong&gt;&lt;/strong&gt;&lt;/em&gt;&lt;/p&gt;</code>	<code>p strong {color: blue;}</code>
Hijo	<code>&lt;p&gt;&lt;em&gt;&lt;strong&gt;&lt;/strong&gt;&lt;/em&gt;&lt;/p&gt;</code>	<code>p &gt; em {color: blue;}</code>

- **Class:**

- en un documento, varios elementos con una misma clase
- todos los elementos con una clase: `.nota {color: blue;}`

- **ID:** un único elemento con una determinada ID en todo el documento

# Sintaxis básica

- **Estructura**

- Documentos (HTML, XML...) -> estructura en la que se basa CSS
- **Herencia** -> los elementos contenidos en otros heredan sus propiedades mientras no se especifique otra cosa

HTML: `<p>Este es un ejemplo de <em>herencias</em></p>`

CSS A: `p {color: blue;}` —————> Este es un ejemplo de herencias

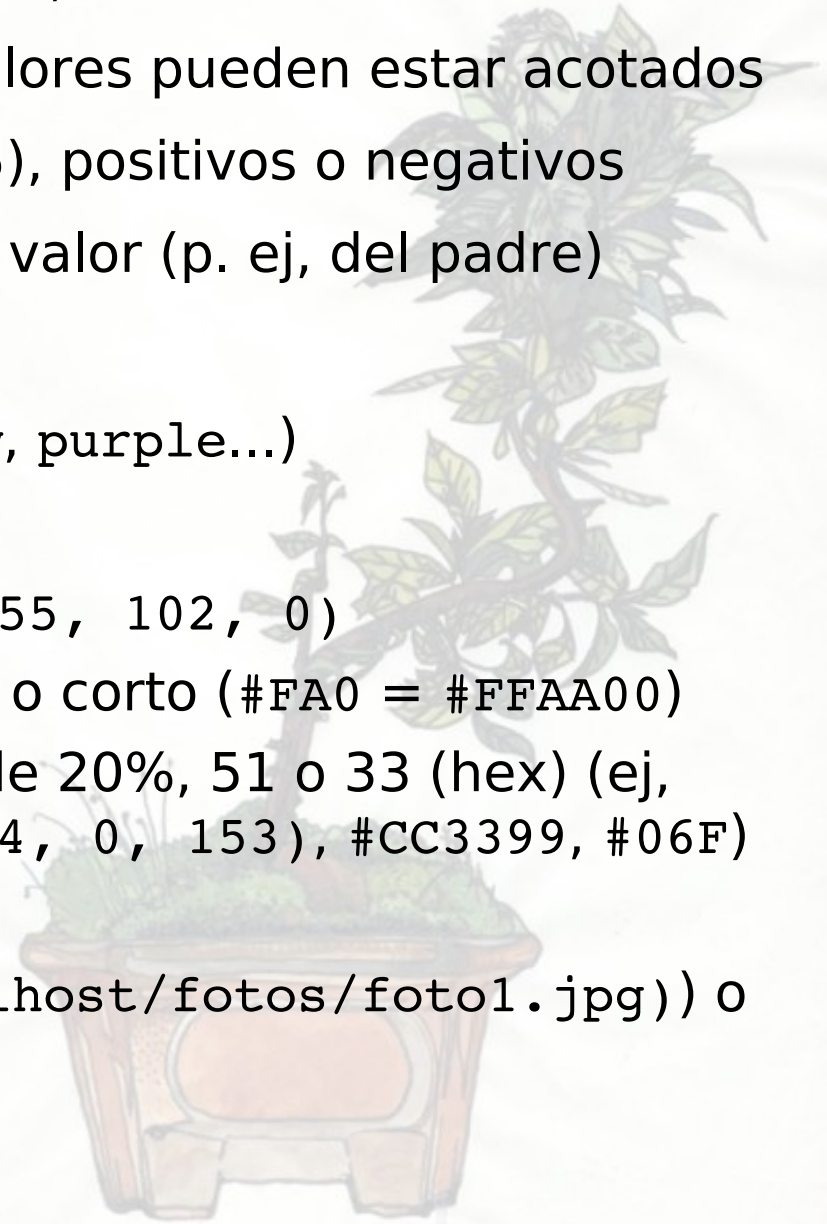
CSS B: `p {color: blue;}` —————> Este es un ejemplo de herencias  
`p em {color: red;}`

- **Cascada** -> Varias reglas que afectan a un mismo elemento: ¿cuál se aplica?
  - La marcada como importante `p {color:red !important;}`
  - La más específica `p.nota {color:red;}` ←  
`p {color: blue;}`
  - La que aparezca más tarde, incluido el documento `p {color:red;}`  
`p {color: blue;}` ←

# Sintaxis básica

- **Unidades**

- Afectan a los valores de color, tamaño, distancias...
- Dependiendo de la propiedad, los valores pueden estar acotados
- **Números:** enteros (2) o reales (12.5), positivos o negativos
- **Porcentajes** (80%), relativos a otro valor (p. ej, del padre)
- **Colores:**
  - Por nombre (blue, green, yellow, purple...)
  - Por RGB:
    - `rgb(100%, 40%, 0%)` o `rgb(255, 102, 0)`
    - hexadecimal: largo (`#FF06CA`) o corto (`#FA0 = #FFAA00`)
    - Colores *Web-Safe*: múltiplos de 20%, 51 o 33 (hex) (ej, `rgb(40%, 80%, 60%)`, `rgb(204, 0, 153)`, `#CC3399`, `#06F`)
- **URLs:** absolutas (`url(http://localhost/fotos/foto1.jpg)`) o relativas (`url(fotos/foto1.jpg)`)



# Sintaxis básica

- **Unidades (y II)**

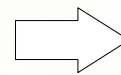
- **Unidades de longitud**

- Absolutas (in, cm, mm, pt, pc)
    - Relativas (px, em, ex)

- **em:** valor del tamaño de fuente para una fuente (ej, tamaño de fuente 12px => 1em=12px)

- Definir tamaños y distancias:

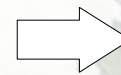
```
h1 {font-size: 12px;  
margin: 1em;}
```



El elemento H1 tendrá un margen de 12px

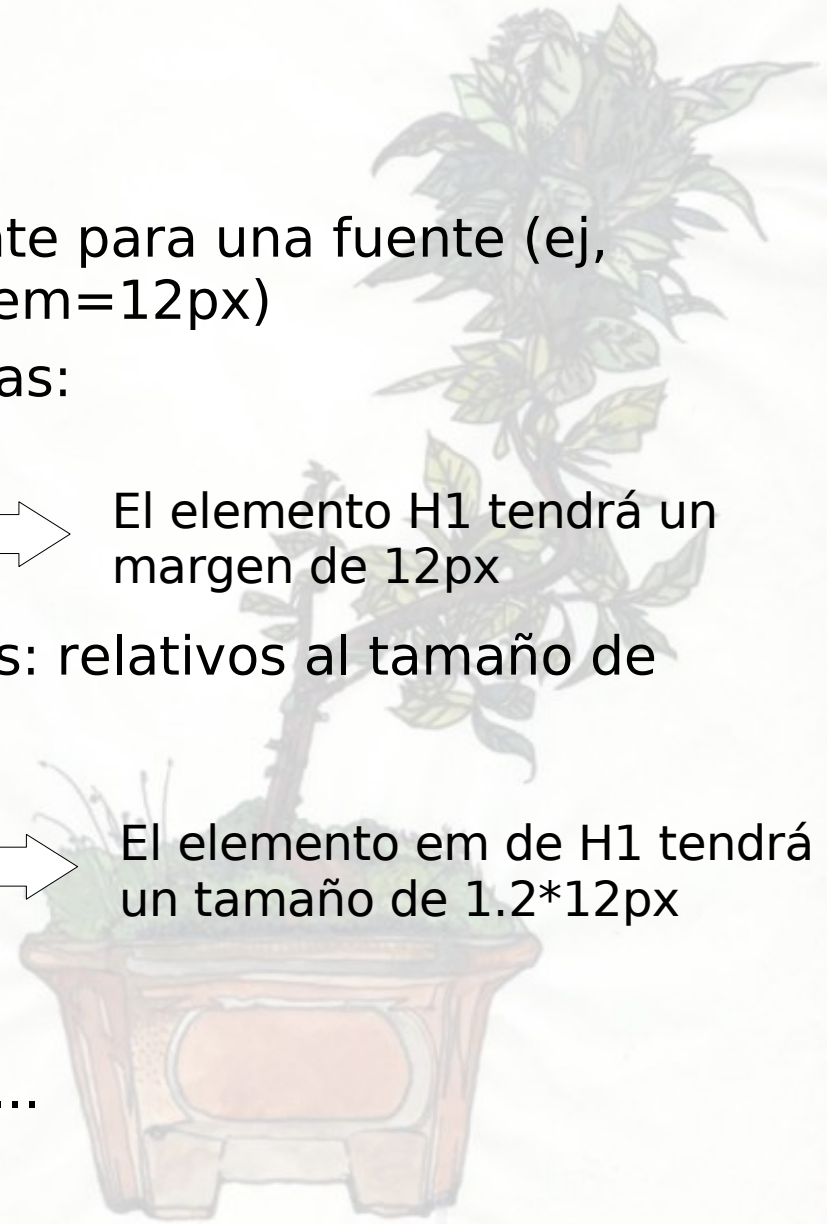
- Definir tamaños de fuentes: relativos al tamaño de fuente padre:

```
h1 {font-size: 12px;}  
h1 strong {font-size: 1.2em;}
```



El elemento em de H1 tendrá un tamaño de 1.2\*12px

- **Palabras especiales:** none, inherit...



# Sintaxis básica

- **Fuentes**

- **font-family:**

- Genéricas: Serif, Sans-serif, Monospace, Cursive, Fantasy
    - Específicas: Times, Verdana, Arial, Helvetica...

- **font-weight:** normal, bold, bolder, lighter, 100, 200...

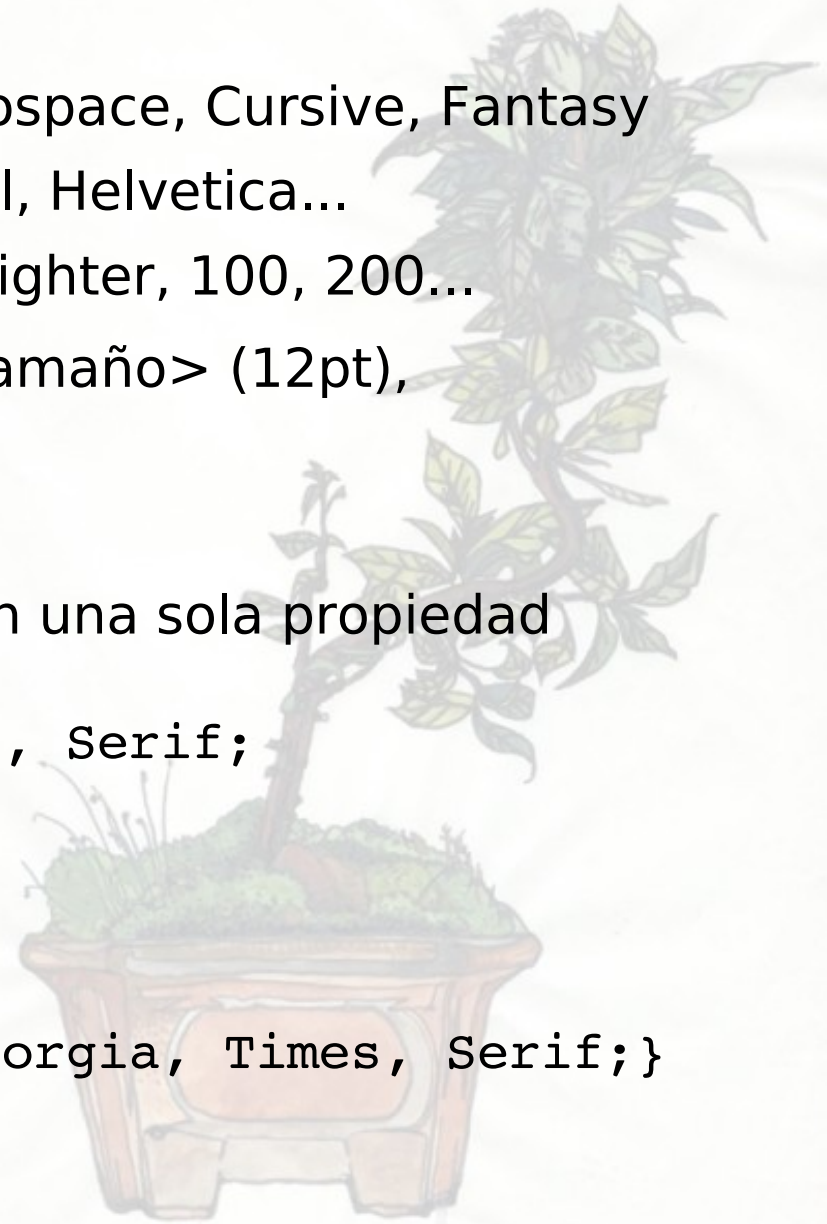
- **font-size:** small, medium, large, <tamaño> (12pt), <porcentaje> (100%)

- **font-style:** italic, oblique, normal

- **font:** recopila todas las anteriores en una sola propiedad

```
p {font-family: Georgia, Times, Serif;  
font-weight: normal;  
font-size: 12px;  
font-style: italic;}
```

```
p {font: normal italic 12px Georgia, Times, Serif;}
```

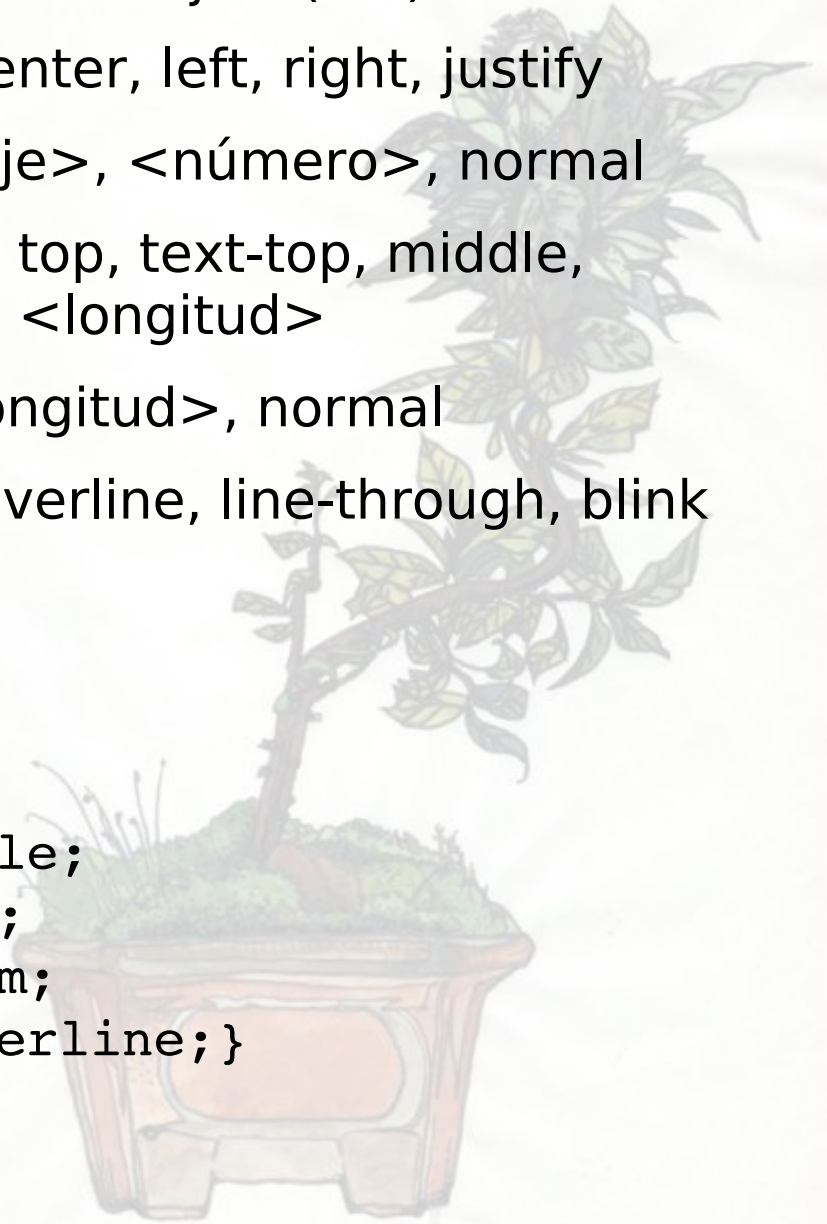


# Sintaxis básica

- **Texto**

- **text-indent:** <longitud> (3em), <porcentaje> (5%)
- **text-align:** (alineación horizontal) center, left, right, justify
- **line-height:** <longitud>, <porcentaje>, <número>, normal
- **vertical-align:** baseline, sub, super, top, text-top, middle, bottom, text-bottom, <porcentaje>, <longitud>
- **word-spacing, letter-spacing:** <longitud>, normal
- **text-decoration:** none, underline, overline, line-through, blink

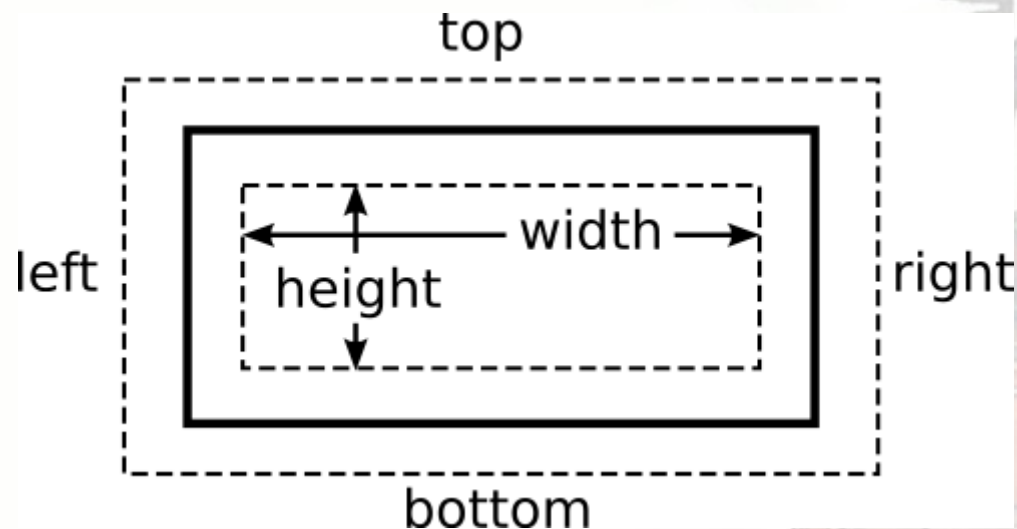
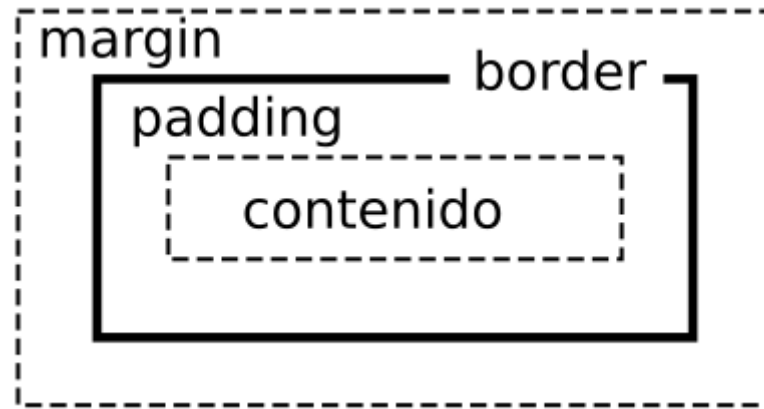
```
p {text-indent: 5em;
    text-align: justify;
    line-height: 2;
    vertical-align: middle;
    word-spacing: normal;
    letter-spacing: 0.2em;
    text-decoration: underline;}
```



# Sintaxis básica

- **Bordes, márgenes y padding**

- Cada elemento genera una caja donde se encuentra el contenido del elemento



# Sintaxis básica

- **Bordes, márgenes y padding (II)**

- **Margin y padding:** <longitud>, <porcentaje>

- Todos iguales: `p{margin: 12px; padding: 5%;}`

- Individualmente:

- margin-top, margin-right, margin-bottom, margin-left

- padding-top, padding-right, padding-bottom, padding-left

```
p{ padding-right: 10px
padding-left: 10px;
margin-top:30px}
```

- Individualmente, agrupados en una sola declaración:

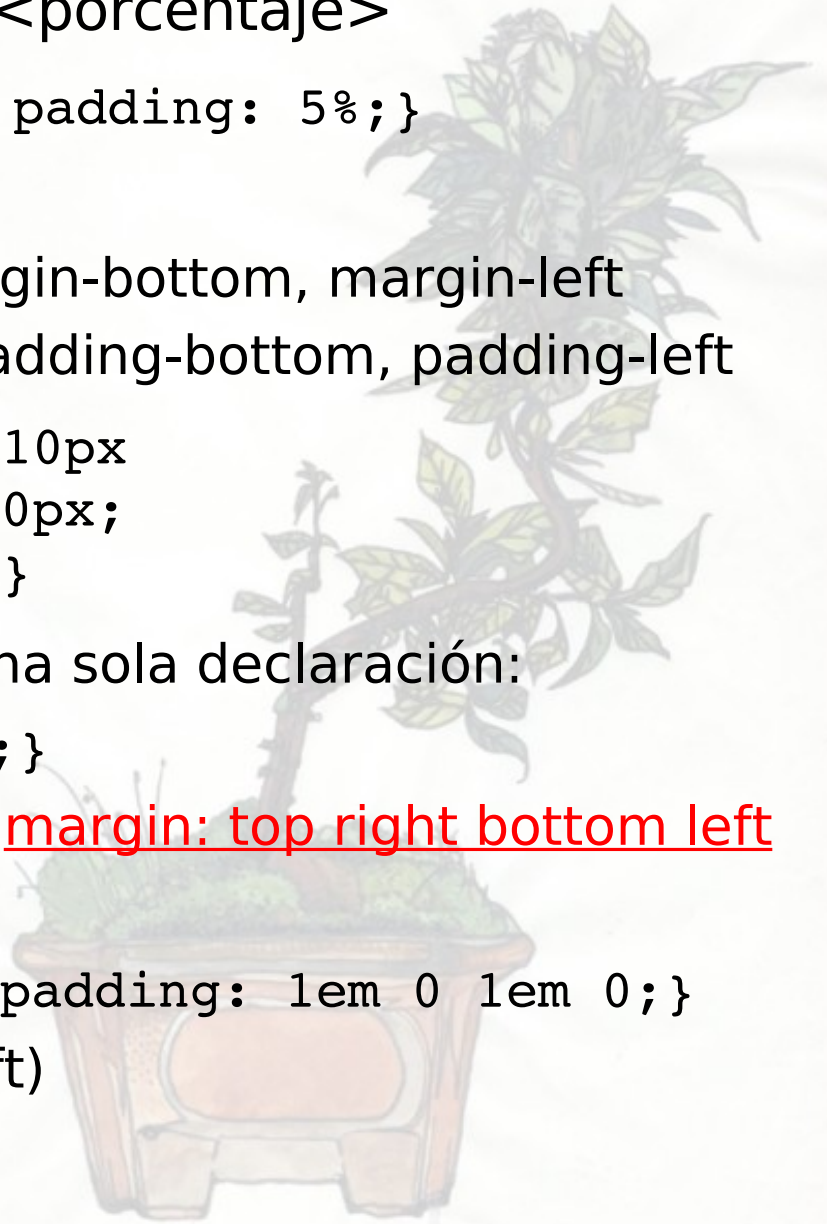
- `p {margin: 20px 0 15px 5%;}`

- desde arriba, sentido horario: margin: top right bottom left

- Agrupados dos a dos:

- `p {padding: 1em 0;} = p {padding: 1em 0 1em 0;}`

- margin: (top,bottom) (right,left)



# Sintaxis básica

- **Bordes, márgenes y padding (y III)**

- **Border:**

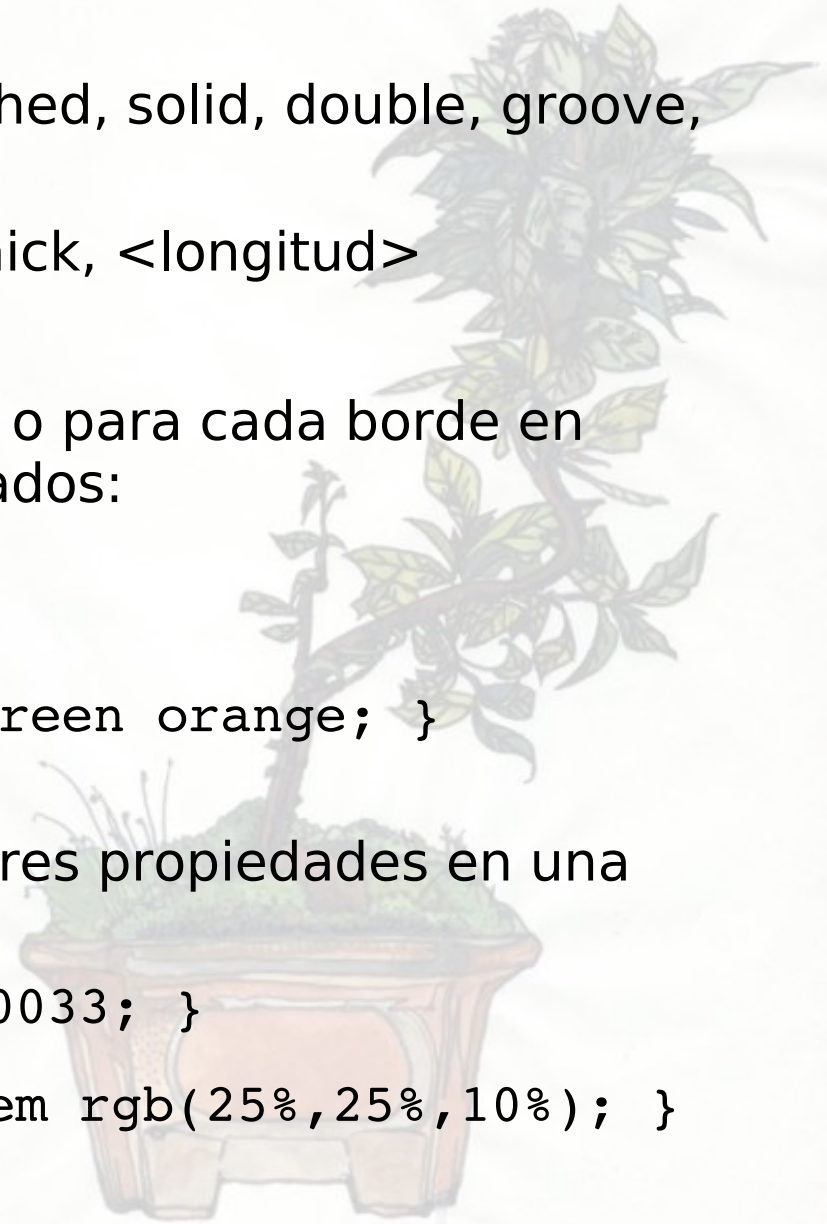
- **border-style:** none, dotted, dashed, solid, double, groove, ridge, inset, outset
    - **border-width:** thin, medium, thick, <longitud>
    - **border-color**
    - Todos se pueden definir globales o para cada borde en particular, por separado o agrupados:

```
p{ border-style: dotted;  
    border-top-width: 2pt;  
    border-color: blue red green orange; }
```

- También se pueden agrupar las tres propiedades en una declaración, global o individual:

```
p{ border: solid thick #FF0033; }
```

```
p{ border-bottom: dashed 1em rgb(25%,25%,10%); }
```

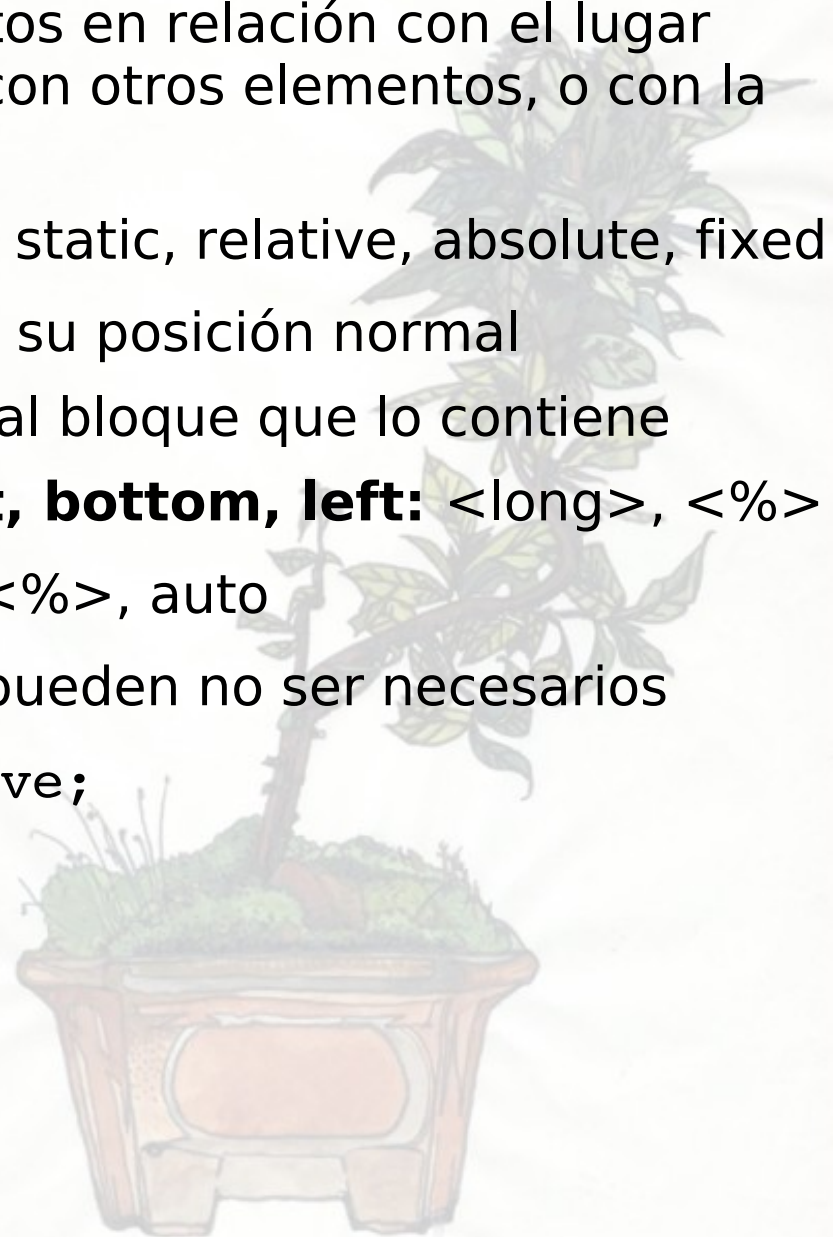


# Sintaxis básica

## • Posicionamiento

- Cómo colocar las cajas de los elementos en relación con el lugar donde aparecen en el flujo de datos, con otros elementos, o con la ventana del navegador
- Tipo de posicionamiento -> **position:** static, relative, absolute, fixed
  - *relative*: desplazamiento relativo a su posición normal
  - *absolute*: desplazamiento relativo al bloque que lo contiene
- Desplazamiento (offset) -> **top, right, bottom, left:** <long>, <%>
- Tamaño -> **width, height:** <long>, <%>, auto
  - si se especifican todos los offsets pueden no ser necesarios

```
p{ position: relative;
  top: 0;
  bottom: 0;
  right: 25%;
  left: 25%;
  width: 50%;
  height: 100%; }
```



# Sintaxis básica

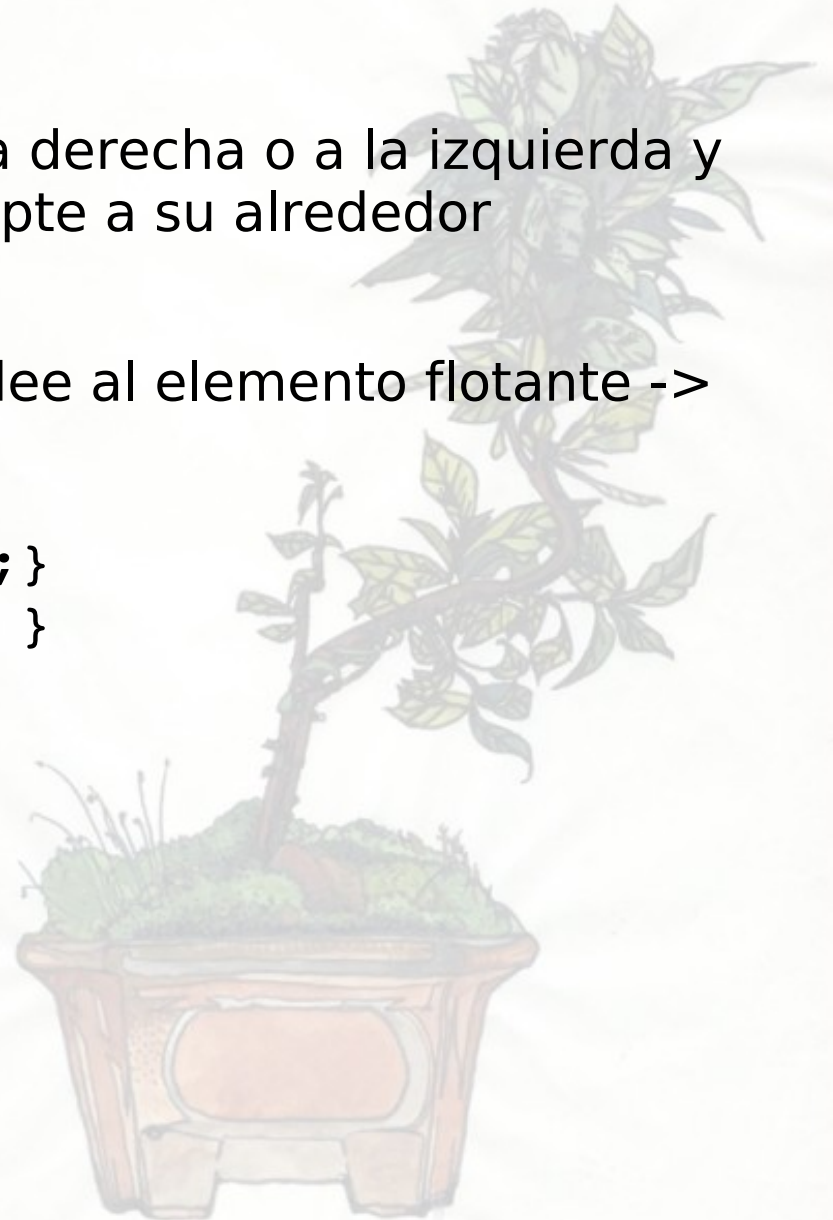
- **Posicionamiento (y II)**

- **Float:**

- permite a un elemento “flotar” a la derecha o a la izquierda y que el resto del documento se adapte a su alrededor
    - **float:** left, right, none
    - para impedir que un elemento rodee al elemento flotante -> **clear:** left, right, none

```
img { float: left; }  
h3 { clear: left; }
```

- Otras: overflow, clip, visibility



# Sintaxis básica

- **Colores y fondos**

- Color de frente -> **color:** <color>

- **Fondos:**

- **background-color:** <color>

- **background-image:** <uri>, none

- **background-repeat:** repeat, repeat-x, repeat-y, no-repeat

- **background-position:** [<%>, <long>, left, center, right]  
[<%>], <long>, top, center, bottom] || [left, center, right] ||  
[top, center, bottom]

- **background-attachment:** scroll, fixed

- Todas juntas -> **background**

```
body { background-color: white;  
        background-image: url(imagenes/fondo.jpg);  
        background-repeat: no-repeat;  
        background-position: bottom right;  
        background-attachment: fixed;}
```

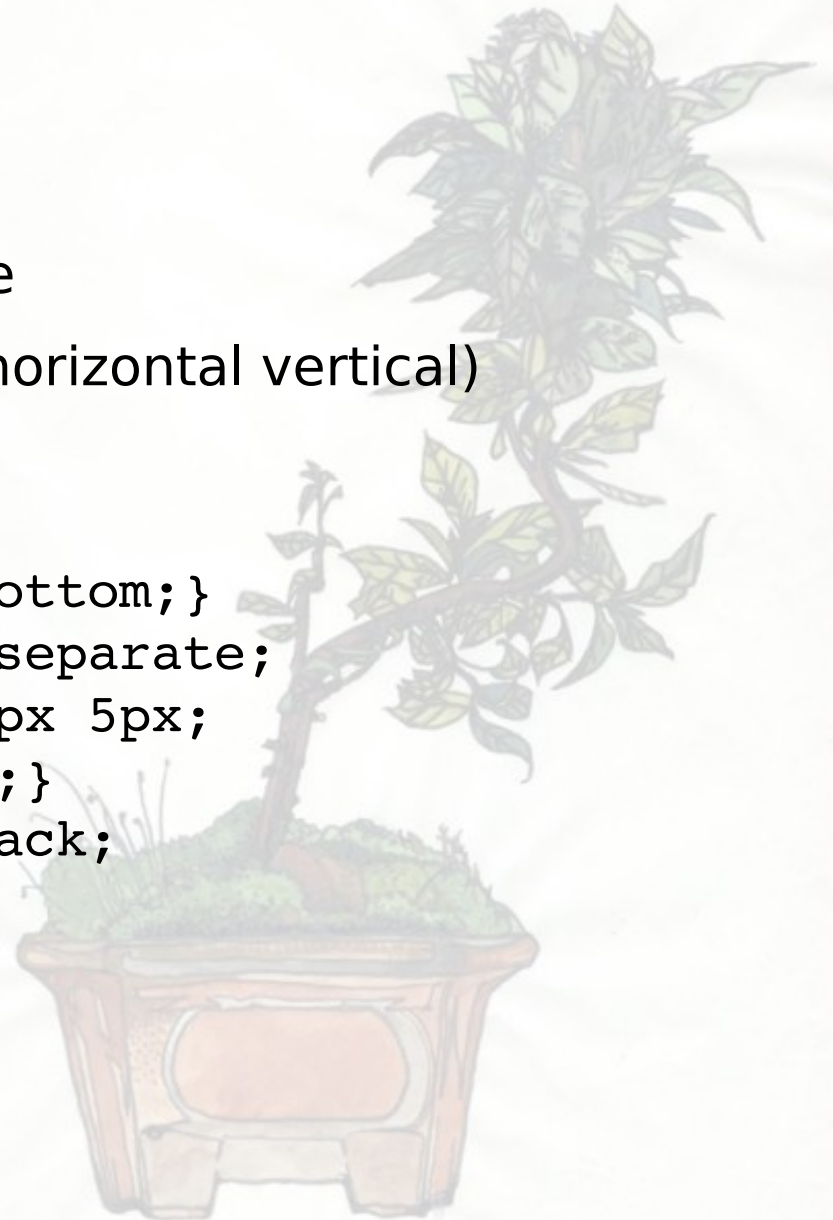
```
body { background: white url(imagenes/fondo.jpg)  
        no-repeat bottom right fixed;}
```

# Sintaxis básica

- **Tablas**

- Para contenido, no para layouts
- **caption-side:** top, bottom
- **border-collapse:** collapse, separate
- **border-spacing:** <long> <long> (horizontal vertical)
- **empty-cells:** show, hide

```
caption {caption-side: bottom;}
table {border-collapse: separate;
       border-spacing: 3px 5px;
       empty-cells: hide;}
td {border: 2pt solid black;
    padding: 3px;}
```

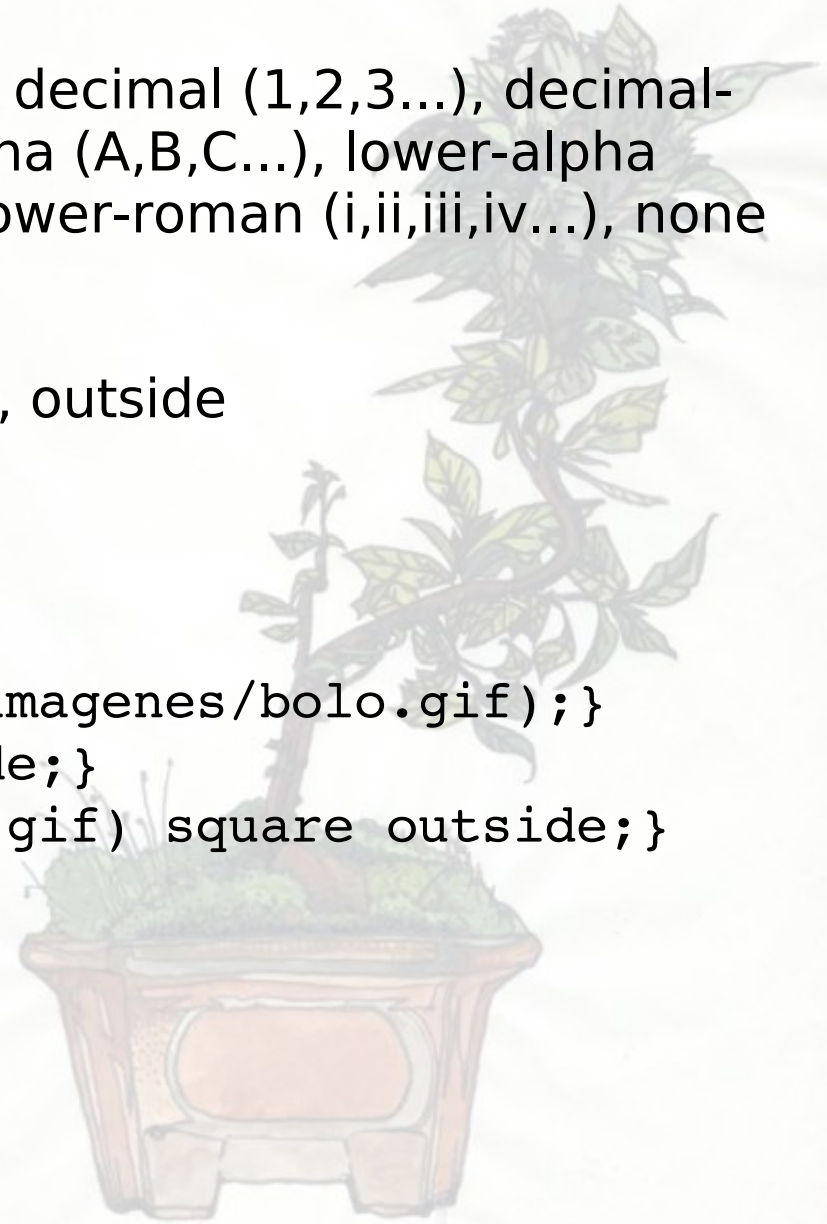


# Sintaxis básica

- **Listas**

- **list-style-type:** disc, circle, square, decimal (1,2,3...), decimal-leading-zero (01,02,03...), upper-alpha (A,B,C...), lower-alpha (a,b,c...), upper-roman (I,II,III,IV...), lower-roman (i,ii,iii,iv...), none
- **list-style-image:** <uri>, none
- **list-style-position** (CSS2.1): inside, outside
- Todas juntas -> **list-style**

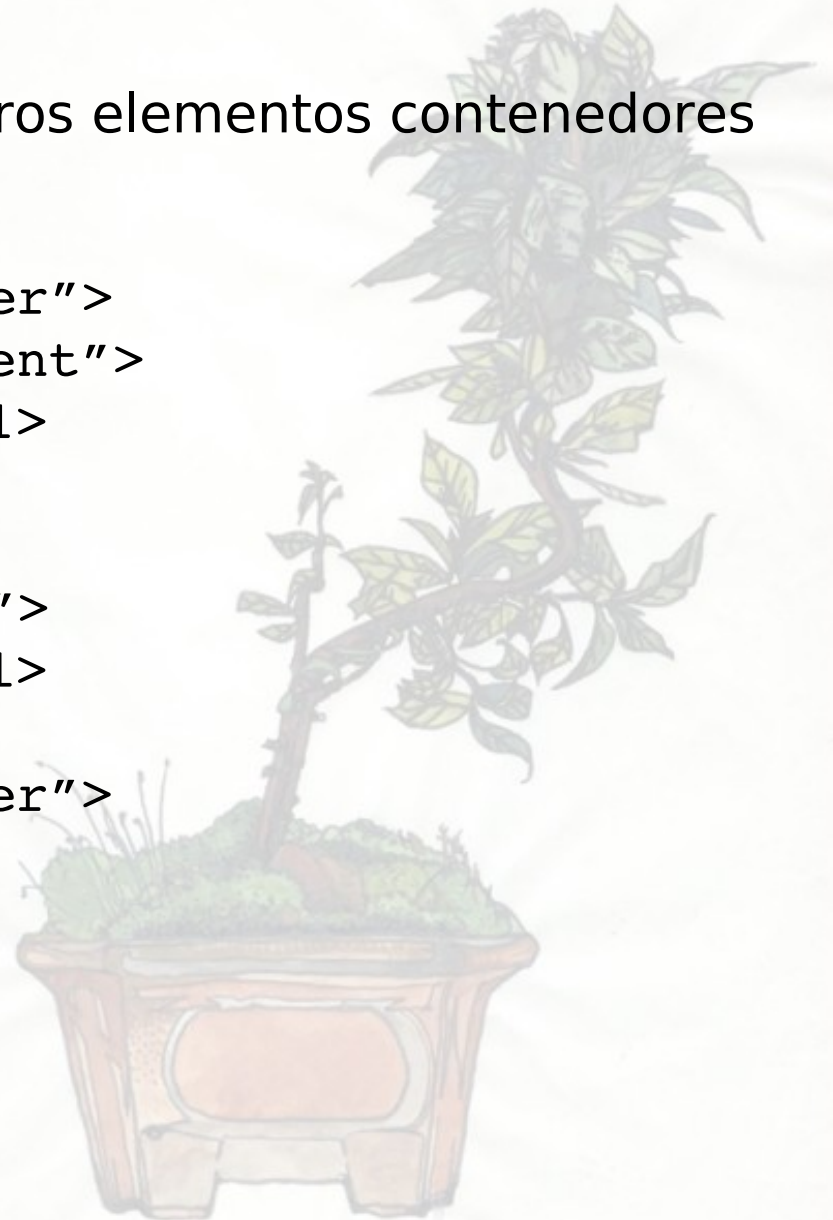
```
ul {list-style-type: circle;}
ul ul {list-style-image: url(imagenes/bolo.gif);}
li {list-style-position: inside;}
ul li {list-style: url(bullet.gif) square outside;}
```



# Recomendaciones

- Hacer un documento **HTML** correcto con una buena estructura facilita darle estilo
  - Englobar los elementos dentro de otros elementos contenedores

```
<body>
  <div id="container">
    <div id="content">
      <h1>...</h1>
      <p>...</p>
    </div>
    <div id="menu">
      <ul>...</ul>
    </div>
    <div id="footer">
      <p>...</p>
    </div>
  </div>
</body>
```



# Recomendaciones (II)

- Utilizar cada elemento en función de su estructura, no para hacer efectos visuales: H1, H2, H3 como encabezados, tablas para contenidos que admiten tablas (ej, catálogos...)
- Para dar estilo a todo un bloque del documento, utilizar `<div>`
- Para dar estilo a toda una línea, utilizar `<span>`
- Para resaltar un trozo de texto, olvidemos `<b></b>` e `<i></i>` y usemos `<strong></strong>` y `<em></em>`, que definiremos después como prefiramos, por ejemplo:

```
strong { font-weight: bold; }  
em { font-style: italic; }
```

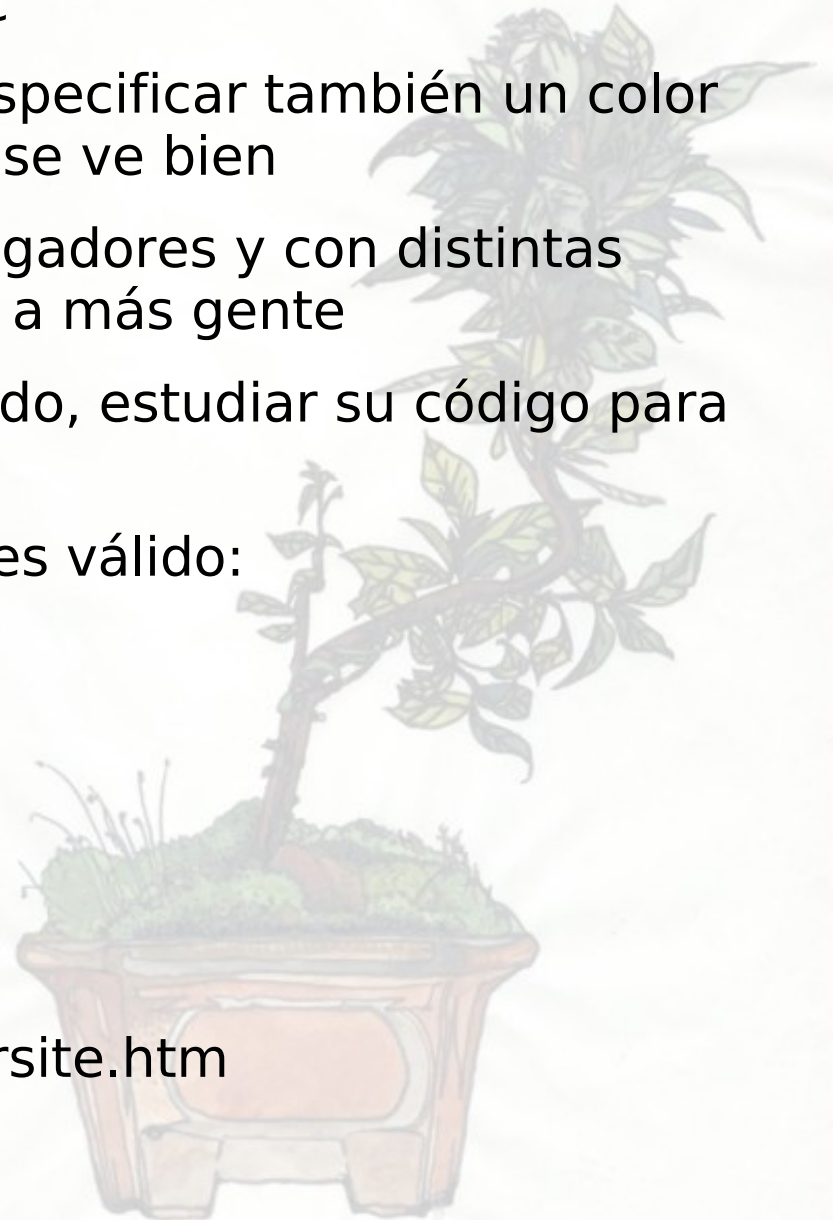
- comprobar que nuestro código es válido: <http://validator.w3.org/>
- En cuanto al **CSS**:
  - Unidades: evitar las unidades absolutas, como `px`, ya que algunos navegadores no permiten redimensionarlas. Usar `em` en su lugar.

# Recomendaciones (y III)

- Fuentes: especificar más de una familia -> si no está instalada la primera, se mostrará la segunda, etc
- Si se especifica un color de frente, especificar también un color de fondo sobre el cual sepamos que se ve bien
- Probar el resultado en distintos navegadores y con distintas resoluciones si queremos que llegue a más gente
- Si encontramos algo bonito navegando, estudiar su código para ver cómo se ha hecho
- Comprobar que nuestro código CSS es válido:  
<http://jigsaw.w3.org/css-validator/>

- **Utilidades:**

- Extensión Web Developer de Firefox
- Web Checker (Craig Cecil)  
<http://www.craigcecil.com/checkyoursite.htm>



# Ejemplos y Referencias

- **Ejemplos**

- CSS Zen Garden: <http://www.csszengarden.com/>
- CSS Mania: <http://cssmania.com/>
- CSS Play de Stu Nicholls: <http://www.cssplay.co.uk/index.html>

- **Referencias**

- W3C: especificaciones y otras referencias  
<http://www.w3.org/Style/CSS/>
- Tutorial CSS echoecho:  
<http://www.echoecho.com/cssintroduction.htm>
- Cascading Style Sheets, 2nd Ed. Eric Meyer. O'Reilly
- Comparativa sintaxis HTML-CSS:  
<http://www.webguild.org/tech/tips/css2html.php>
- Esta charla estará en <http://ana.saizgarcia.net>



**¡Gracias por  
vuestra atención!**

