

Tema 2: Tecnologías para el desarrollo en el cliente

HTML y XHTML

CSS

Javascript

Sistemas de Información para Internet
3º del Grado de Ingeniería Informática (tres menciones)

J. Francisco Chicano

Departamento de Lenguajes y Ciencias de la Computación
Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga

Cascading Style Sheet (CSS)



CSS

Motivación

- CSS (Cascading Style Sheets) es un lenguaje que permite modificar la presentación del contenido de un documento HTML.
- En versiones anteriores de HTML, era posible cambiar el formato en el documento HTML (CSS no era estrictamente necesario).
- En la actual versión de HTML (5), no es posible establecer el formato en el documento HTML, siendo necesario utilizar CSS para esto.

CSS

¿Qué es?

- Las hojas de estilo nos permiten especificar cómo se va a presentar el contenido de las distintas etiquetas HTML
- De este modo, podemos cambiar la presentación del documento HTML sin necesidad de modificar el contenido del documento.
- Cuando se emplean de modo adecuado las hojas de estilo, habitualmente empleamos una única hoja de estilo para toda una web (varias páginas).
- Por ejemplo: con solo modificar una línea de dicha hoja de estilo podemos hacer que el color de la fuente de los párrafos de nuestro documento sea **azul**, o vuelva a ser **negra**, o emplee un tipo de fuente diferente.

CSS

¿Cómo se utiliza?

- Para utilizar los estilos definidos en un fichero CSS desde un fichero HTML debemos incluir una etiqueta dentro de la cabecera (<head>) del documento:

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

- El atributo **rel** indica qué tipo de recurso estamos referenciando
- **type** siempre toma el valor "text/css"
- El atributo **href** debe indicar la ruta de la hoja de estilo que queremos emplear en la página web.

CSS

Estructura

- Una hoja de estilo está formada por un conjunto de **reglas**
- Para definir una regla se usa la siguiente sintaxis:

selector { declaración }

- Por ejemplo:

`h1 { color: green }`

- Es posible agrupar varias reglas que tienen la misma declaración separando los selectores por comas

`h1 { color: green }`

`h2 { color: green }`

`h3 { color: green }`



`h1, h2, h3 { color: green }`

CSS

Declaración

- La declaración está formada por cero o más pares propiedad/valor
- Si hay más de una, los pares deben separarse por punto y coma ;
- La propiedad se separa del valor por dos puntos :
- Ejemplo:

```
p { color: blue; font-variant: small-caps }
```

- Detrás del último par puede haber punto y coma o no
- Los comentarios son como los de C : `/* comentario */`
- Las propiedades son utilizadas para determinar el formato de los elementos a los que afectan

CSS

Selectores

- Los selectores determinan a qué elementos del documento (X)HTML afecta la regla
- Para construir un selector se usan **selectores simples** unidos mediante **combinadores**
- Un selector simple es un **selector de tipo** o **universal** seguido de cero o más **selectores de atributo, de ID o de pseudo-clase**
- Un **combinador** es el **espacio en blanco**, el **+**, el **~** o el **>**
- Ejemplos:

```
* {color : green}    h1 > p {font-size:12pt}    p.rojo {color: red}
```


CSS

Selectores

- Selector de tipo es aquel que coincide con una etiqueta de (X)HTML
 - h1, h2, p, div, span, ...
- Selector universal es *
- Si está acompañado de un selector de atributo, ID o pseudo-clase se puede obviar
 - *.hola = .hola
 - *#tabla = #tabla
- Selectores de atributo: van entre corchetes: h1[color]
- Selectores de pseudo-clase: van separados por dos puntos: h1:first-child
- Selectores de ID: van separados por #: h1#datos

CSS

Resumen de selectores

Pseudo-clase

Atributos

Pseudo-classes

Pseudo-elementos

Patrón	Aplicado a:
<code>*</code>	Cualquier elemento
<code>e</code>	Elemento <code><e></code>
<code>e f</code>	Cualquier <code><f></code> que esté incluido en <code><e></code>
<code>e > f</code>	Cualquier <code><f></code> que es hijo de <code><e></code>
<code>e:first-child</code>	<code><e></code> cuando es el primer hijo de su elemento padre
<code>e + f</code>	<code><f></code> si está precedido inmediatamente por un <code><e></code>
<code>e[miAtr]</code>	<code><e></code> si posee el atributo <code>"miAtr"</code>
<code>e[miAtr="v"]</code>	<code><e></code> si posee el atributo <code>"miAtr"</code> y este tiene el valor <code>"v"</code>
<code>e[miAtr~="v"]</code>	<code><e></code> si su atributo <code>"miAtr"</code> es una lista de valores separados por espacios y uno es <code>"v"</code>
<code>e[lang "es"]</code>	<code><e></code> si su atributo <code>"lang"</code> es una lista de valores separados por guiones y comienza con <code>"es"</code>
<code>e.miClase</code>	<code><e></code> cuya clase es <code>"miClase"</code>
<code>e#miId</code>	<code><e></code> cuyo identificador es <code>miId</code>
<code>a:link</code>	Enlaces no visitados
<code>a:visited</code>	Enlaces visitados
<code>e:active</code>	<code><e></code> cuando es activado (tiempo entre que se pulsa un botón sobre él y se suelta)
<code>e:hover</code>	<code><e></code> cuando se posiciona el cursor sobre él pero no se activa
<code>e:focus</code>	<code><e></code> cuando tiene el foco posicionado en él
<code>e:lang(c)</code>	<code><e></code> si está marcado con el idioma <code>c</code>
<code>e:first-line</code>	Primera línea de <code><e></code>
<code>e:first-letter</code>	Primera letra de <code><e></code>
<code>e:before</code>	Aplica contenido antes de <code><e></code>
<code>e:after</code>	Aplica contenido después del elemento <code><e></code>
<code>e , f</code>	Aplica el mismo contenido a los elementos <code><e></code> y <code><f></code>

CSS

Ejemplos de selectores

- `div`
- `div p`
- `ul li`
- `p > em`
- `p + em`
- `p:first-child em`
- `.datos span[class]`
- `#tabla > tr[colspan] td:first-child + td`

CSS

Atributos id y class

- Hay dos atributos en (X)HTML de especial importancia: **id** y **class**
- El valor de **id** se usa en los selectores de ID para determinar el elemento al que afecta
- No debe aparecer más de un elemento con el mismo **id** en un documento
- El atributo **class** está relacionado con los selectores de clase
- Un elemento puede tener más de una clase y una clase puede estar asignada a más de un elemento

CSS

Elementos DIV y SPAN en (X)HTML

- Estos elementos no tienen presentación en el documento HTML por sí mismos, sino que suelen emplearse "de apoyo" para las hojas de estilo
- El elemento SPAN puede emplearse dentro de un texto para indicar que un fragmento de dicho texto va a representarse empleando un estilo diferente que el resto.
- Ejemplo:

```
<p> Mi madre tiene ojos <span class="azul"> azules </span>  
y pelo <span class="amarillo"> rubio </span>.</p>
```
- Esto nos va a permitir que las palabras "azules" y "rubio" tengan una presentación diferente que el resto del texto que aparece dentro del párrafo

CSS

Algunos ejemplos

```
#especial p{  
    font-size : 16pt;  
}
```

- Cambiará el tamaño de fuente del primer párrafo que se muestra a continuación, pero no del segundo:

```
<div id="especial" >  
    <p> Este es el primer párrafo </p>  
    </div>  
    <p>Y este es el segundo párrafo </p>
```

CSS

Algunos ejemplos

```
.especial{  
  font-size : 16pt;  
}
```

- Afectará al primer párrafo que se muestra a continuación, pero no al segundo:

```
<p class="especial" > Este es el primer párrafo </p>
```

```
<p>Y este es el segundo párrafo </p>
```

- La regla:

```
p.especial{  
  font-size : 16pt;  
}
```

- sólo afectará a elementos P, y no a cualquier otro, aunque ese otro también tenga asignada la clase "especial".

CSS

Valores

- Los valores que pueden tomar las propiedades en CSS se agrupan en distintos tipos (como en los lenguajes de programación)
- Cada propiedad admite **uno o varios** de estos tipos
- Los tipos más importantes son:
 - Enteros
 - Reales
 - Porcentajes
 - Longitudes
 - Cadenas de caracteres
 - URLs
 - Colores
- La lista exhaustiva se puede encontrar en
 - <http://www.w3.org/TR/CSS2/syndata.html>

CSS

Valores: números y porcentajes

- Los **enteros** se representan como es habitual en los lenguajes de programación:
 - 0
 - +3
 - -2
- Los **reales** se caracterizan por un punto decimal y al menos un dígito tras el mismo
 - .4
 - -.6
 - 4.7
- Los **porcentajes** son números seguidos del carácter '%'
 - 50%
 - 4.5%

CSS

Valores: longitudes

- Número seguido de una unidad de longitud. Ej.: 12pt
- Las unidades de longitud que se pueden usar son:

- cm: centímetro (10 mm)
 - mm: milímetro
 - in: pulgada (2.54 cm)
 - pt: punto (1/72 de pulgada)
 - pc: pica (12 puntos)
 - px: píxel (0.75 puntos)
 - em: valor de la propiedad font-size
 - ex: alto de la letra 'x'
-
- Unidades físicas
- Unidades absolutas
- Unidades relativas

- Dependiendo del dispositivo de salida se fija una unidad física (impresoras) o el píxel (pantallas)

CSS

Valores: cadenas y URLs

- Las cadenas pueden ir delimitadas por comillas dobles o simples:

```
"this is a 'string'"  
"this is a \"string\""  
'this is a "string" '  
'this is a \'string\''
```

- Se usa el mismo mecanismo de escape que es habitual en los lenguajes de programación tipo C
- Una URL es una “cadena” que va en el “argumento” del “operador url”

```
url("http://www.example.com/pinkish.png")
```

- La “cadena” no necesita llevar comillas

```
url(http://www.example.com/redball.png)
```

CSS

Valores: colores

- Para indicar un color se pueden usar etiquetas predefinidas o las componentes RGB del mismo

color: #ff0000;

color: #f00;

color: red;

Cada dígito se duplica

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		

- Los colores en CSS se pueden indicar en formato hexadecimal como #RRGGBB o #RGB.

CSS

Propiedades

- En CSS 2.1 se definen 115 propiedades
- La lista completa se encuentra en
 - <http://www.w3.org/TR/CSS2/propidx.html>
- Allí se indican los tipos y valores que admiten, los elementos a los que afectan, su valor inicial y si se hereda (entre otras cosas)
- Todas las propiedades admiten el valor `inherit`, que indica que el valor se hereda del elemento padre
- Es posible indicar que un valor es importante usando `!important` tras el valor

```
body {  
    color: black !important;  
    background: white !important;  
}
```

CSS

Propiedades

Nombre de la propiedad	Valores que puede tomar (lo que aparece entre ángulos son tipos o valores especiales)	Valor inicial	Elemento al que se aplica	¿A qué se refieren los porcentajes?	Se hereda?	Grupo
------------------------	---	---------------	---------------------------	-------------------------------------	------------	-------

Name	Values	Initial value	Applies to (Default: all)	Inherited?	Percentages (Default: N/A)	Media groups
'azimuth'	<angle> [[left-side far-left left center-left center center-right right far-right right-side] behind] leftwards rightwards inherit	center		yes		aural
'background-attachment'	scroll fixed inherit	scroll		no		visual
'background-color'	<color> transparent inherit	transparent		no		visual
'background-image'	<uri> none inherit	none		no		visual
'background-position'	[[<percentage> <length> left center right] [<percentage> <length> top center bottom]?] [[left center right] [top center bottom]] inherit	0% 0%		no	refer to the size of the box itself	visual
'background-repeat'	repeat repeat-x repeat-y no-repeat inherit	repeat		no		visual
'background'	['background-color' 'background-image' 'background-repeat' 'background-attachment' 'background-position'] inherit	see individual properties		no	allowed on 'background-position'	visual
'border-collapse'	collapse separate inherit	separate	'table' and 'inline-table' elements	yes		visual
'border-color'	[<color> transparent]{1,4} inherit	see individual properties		no		visual
'border-spacing'	<length> <length>? inherit	0	'table' and 'inline-table' elements	yes		visual

CSS

Propiedades: texto

- El tamaño de la fuente se indica mediante la propiedad `font-size`

`font-size: <tamaño>;`

donde el tamaño habitualmente se expresa en puntos:

`font-size: 16pt;`

- Existen valores especiales para esta propiedad

- Valores absolutos: `xx-small`, `x-small`, `small`, `medium`, `large`, `x-large`, `xx-large`
- Valores relativos: `smaller`, `larger`

- El color del texto se indica con la propiedad `color`

`color: red;`

- El color de fondo puede configurarse mediante la propiedad `background-color`

`background-color: blue;`

CSS

Propiedades: texto

- El tipo de fuente se declara empleando la propiedad `font-family`:

`font-family: <fuente>;`

- Siempre suele ser buena idea emplear una lista de varias fuentes, y no una única fuente, ya que corremos el riesgo de que el usuario no tenga instalada dicha fuente en su equipo. Por ejemplo:

`font-family: Times, Times New Roman, serif ;`

- Este estilo quiere decir que deberemos intentar usar la fuente Times; en caso de que esta fuente no esté disponible se intentará usar "Times New Roman". Si ésta tampoco estuviese disponible se emplearía una fuente predeterminada del grupo de Fuentes "serif".

CSS

Propiedades: texto

- Podemos indicar el estilo de la fuente empleando la propiedad `font-style`

`font-style: <estilo>;`

donde <estilo> puede tomar los siguientes valores:

- `italic`: el texto se mostrará en cursiva.
- `oblique`: el texto será oblicuo (muy parecido a la cursiva).
- `normal`: el texto se mostrará normal.



italic
normal

CSS

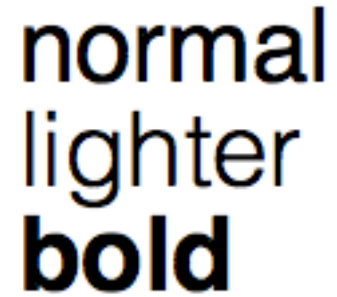
Propiedades: texto

- El grosor del texto se puede indicar con la propiedad `font-weight`:

- `font-weight: <grosor>;`

donde <grosor> puede tomar los siguientes valores:

- `normal`: mostrará el texto empleando un grosor normal
- `lighter`: mostrará el texto con un grosor menor del normal
- `bold`: mostrará el texto negrita.
- `bolder`: mostrará el texto en una negrita todavía más gruesa.
- Un número del 100 al 900 (en múltiplos de 100): donde el grosor mínimo será el número 100, y el máximo el número 900



normal
lighter
bold

CSS

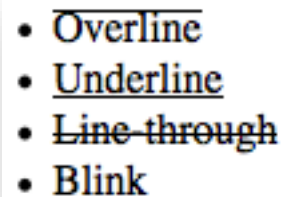
Propiedades: texto

- Se pueden indicar distintas decoraciones para el texto empleando la propiedad `text-decoration`:

- `text-decoration: <decoración>;`

donde <decoración> puede tomar los siguientes valores:

- `underline`: subraya el texto
- `overline`: dibuja una línea encima del texto
- `line-through`: tacha el texto
- `blink`: hace parpadear el texto (no en todos los navegadores)
- `none`: modo normal, sin emplear ningún tipo de decoración

- 
- Overline
 - Underline
 - ~~Line-through~~
 - Blink

CSS

Propiedades: texto

- Es habitual que queramos cambiar el aspecto de los enlaces visitados y no visitados
- Para esto, usamos las pseudo-clases `a:link` y `a:visited` junto con las propiedades `text-decoration` y `color`

```
a:link { text-decoration:none; color:#0000cc;}
```

```
a:visited {text-decoration:none; color:#ffcc33;}
```

- También es habitual querer cambiar el estilo de un elemento cuando el ratón pasa por encima:

```
a:hover { text-decoration:underline; color:#999999; font-weight:bold;}
```

CSS

Propiedades: texto

- Para configurar el alineado del texto podemos emplear la propiedad `text-align`:

- `text-align: <alineado>;`

donde <alineado> puede tomar los valores

- `left`: alineación izquierda
- `right`: alineación derecha
- `center`: centrado
- `justify`: justificación completa

Lorem ipsum ad his scripta blandit partiendo, eum fastidii
accumsan euripidis in, eum liber hendrerit an. Qui ut wisi vocibus
suscipiantur, quo dicit ridens inciderint id. Quo mundi lobortis
reformidans eu, legimus senserit definiebas an eos.

Lorem ipsum ad his scripta blandit partiendo, eum fastidii
accumsan euripidis in, eum liber hendrerit an. Qui ut wisi vocibus
suscipiantur, quo dicit ridens inciderint id. Quo mundi lobortis
reformidans eu, legimus senserit definiebas an eos.

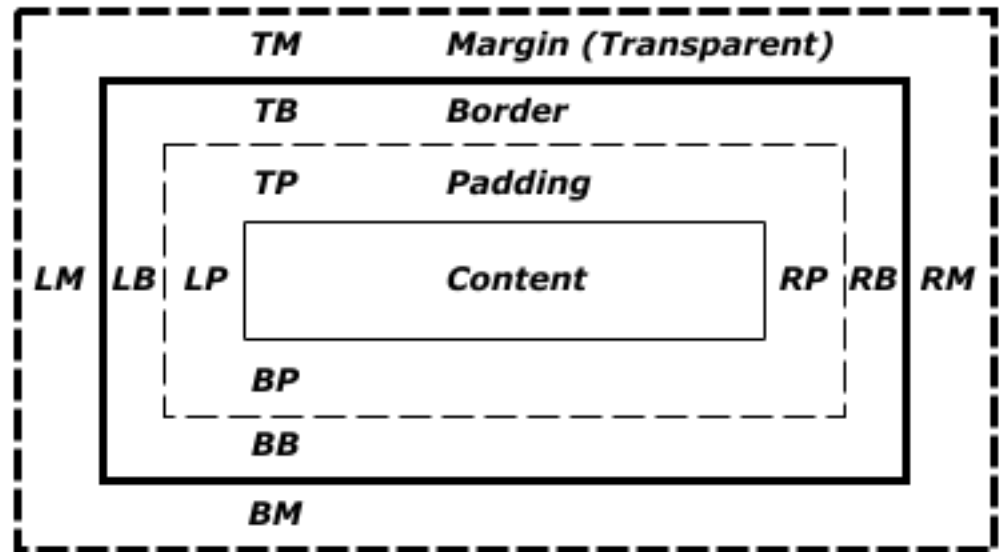
Lorem ipsum ad his scripta blandit partiendo, eum fastidii
accumsan euripidis in, eum liber hendrerit an. Qui ut wisi vocibus
suscipiantur, quo dicit ridens inciderint id. Quo mundi lobortis
reformidans eu, legimus senserit definiebas an eos.

Lorem ipsum ad his scripta blandit partiendo, eum fastidii
accumsan euripidis in, eum liber hendrerit an. Qui ut wisi vocibus
suscipiantur, quo dicit ridens inciderint id. Quo mundi lobortis
reformidans eu, legimus senserit definiebas an eos.

CSS

Propiedades: modelo de caja

- CSS usa el siguiente modelo para la visualización de los elementos de una página. Define:
 - Margen (margin)
 - Borde (border)
 - Relleno (padding)
- Cada uno tiene 4 lados
 - top
 - right
 - bottom
 - left



- Margin edge
- Border edge
- Padding edge
- Content edge

CSS

Propiedades: márgenes

- Es posible configurar los márgenes que hay alrededor de un elemento empleando las propiedades:
 - `margin-top: <cantidad>;`
 - `margin-bottom: <cantidad>;`
 - `margin-left: <cantidad>;`
 - `margin-right: <cantidad>;`
- Donde <cantidad> es una longitud o un porcentaje
- Es posible indicar los cuatro márgenes con una sola propiedad:

```
body { margin: 2em }           /* todos los márgenes a 2em */
body { margin: 1em 2em }       /* arriba y abajo = 1em, derecha e izquierda = 2em */
body { margin: 1em 2em 3em }   /* arriba=1em, derecha=2em, abajo=3em, izquierda=2em */
body { margin: 1em 2em 3em 2em } /* arriba=1em, derecha=2em, abajo=3em, izquierda=2em */
```


CSS

Propiedades: relleno

- Es posible configurar el relleno que hay alrededor de un elemento empleando las propiedades:
 - `padding-top: <cantidad>;`
 - `padding-bottom: <cantidad>;`
 - `padding-left: <cantidad>;`
 - `padding-right: <cantidad>;`
- Donde <cantidad> es una longitud o un porcentaje
- Es posible indicar los cuatro rellenos con una sola propiedad:

```
body { padding: 2em }           /* todos los rellenos a 2em */
body { padding: 1em 2em }       /* arriba y abajo = 1em, derecha e izquierda = 2em */
body { padding: 1em 2em 3em }   /* arriba=1em, derecha=2em, abajo=3em, izquierda=2em */
body { padding: 1em 2em 3em 2em } /* arriba=1em, derecha=2em, abajo=3em, izquierda=2em */
```

CSS

Propiedades: borde

- Cada borde tiene tres propiedades asociadas: `width`, `color` y `style`
- Cada una de estas propiedades se puede definir con una sola propiedad
 - `border-width`
 - `border-color`
 - `border-style`
- Siguen la misma regla que el margen y el relleno cuando se especifica más de un valor
- También se pueden especificar de forma separada: `border-top-width`, `border-right-color`, ...

CSS

Propiedades: borde

- El **ancho** de un borde debe ser una **longitud** o uno de los siguientes valores:
 - thin
 - medium
 - thick
- El **color** es un valor de tipo **color** o el valor **transparent**
- El **estilo** puede ser: **none**, **hidden**, **dotted**, **dashed**, **solid**, **double**, **groove**, **ridge**, **inset**, **outset**

none

hidden

dotted

dashed

solid

double

groove

ridge

inset

outset

CSS

Propiedades: borde

- Es posible indicar el ancho, estilo y color de un borde con una única propiedad:
 - border-top
 - border-right
 - border-bottom
 - border-left
 - border
- En tal caso se especifican los valores separados por espacios

```
h1 { border-bottom: thick solid red }
```
- En total hay 20 propiedades relacionadas con el borde

CSS

Propiedades: posicionamiento

- Con CSS podemos mover los elementos de sitio usando las propiedades `position`, `left`, `top`, `right` y `bottom`
- Los valores de `position` son:
 - `static` (valor por defecto): el elemento se muestra como indique el flujo normal
 - `relative`: se ubica según el flujo normal y se desplaza
 - `absolute`: se saca del flujo normal y se ubica en una posición absoluta con respecto al elemento padre
 - `fixed`: igual que `absolute` con la particularidad de que si se hace `scroll` el elemento sigue en el mismo sitio
- La propiedad `z-index` permite establecer la “profundidad” del elemento
 - Valores mayores quedan encima de los inferiores

CSS

Propiedades: posicionamiento

- Otras propiedades útiles cuando se posicionan elementos:
 - `width`: ancho del elemento
 - `height`: alto del elemento
 - `visibility`: indica si se ve el elemento
 - `clip`: permite especificar el rectángulo visible del elemento
 - `overflow`: determina qué ocurre si el contenido sobrepasa el tamaño del elemento
 - `display`: cambia el modelo de objetos (bloque o elemento en línea)

CSS

Asignación de valores

- Dada una propiedad para un elemento, es posible que haya varias reglas que casen con el elemento y tengan un valor diferente para la propiedad
- A esto debemos sumar que pueden existir **varios ficheros CSS**
- También es posible especificar estilos usando el atributo **style** de los elementos HTML
- Además, el navegador posee una hoja de estilos para el **usuario** y **otra propia**
- ¿Cómo se determina el valor de una propiedad para un elemento?

CSS

Asignación de valores

- Hay dos conceptos clave en el mecanismo para asignar valores:
 - Cascada
 - Herencia
- Dada una propiedad de un elemento, el mecanismo de cascada determina su valor de la siguiente forma:
 - Se determinan todas las declaraciones que afectan a la propiedad
 - Se ordenan estas reglas en las siguientes 5 categorías:
 - Declaraciones del navegador
 - Declaraciones normales del usuario
 - Declaraciones normales del autor
 - Declaraciones importantes del autor
 - Declaraciones importantes del usuario

Las fuentes están dadas en
orden ascendente de preferencia

CSS

Asignación de valores

- Dada una propiedad de un elemento, el mecanismo de cascada determina su valor de la siguiente forma (cont):
 - Se ordenan las declaraciones por especificidad dentro de cada categoría
 - a= 1 si la declaración se encuentra en un atributo style; 0 en otro caso
 - b= número de atributos ID en el selector
 - c= número de otros atributos y pseudo-clases en el selector
 - d= número de elementos y pseudo-elementos en el selector
 - La especificidad de la declaración se define como el número formado concatenando a, b, c y d
 - El orden usado para determinar la mayor especificidad es el orden lexicográfico

```
*           {} /* a=0 b=0 c=0 d=0 -> specificity = 0,0,0,0 */
li          {} /* a=0 b=0 c=0 d=1 -> specificity = 0,0,0,1 */
li:first-line {} /* a=0 b=0 c=0 d=2 -> specificity = 0,0,0,2 */
ul li       {} /* a=0 b=0 c=0 d=2 -> specificity = 0,0,0,2 */
ul ol+li    {} /* a=0 b=0 c=0 d=3 -> specificity = 0,0,0,3 */
h1 + *[rel=up] {} /* a=0 b=0 c=1 d=1 -> specificity = 0,0,1,1 */
ul ol li.red {} /* a=0 b=0 c=1 d=3 -> specificity = 0,0,1,3 */
li.red.level {} /* a=0 b=0 c=2 d=1 -> specificity = 0,0,2,1 */
#x34y       {} /* a=0 b=1 c=0 d=0 -> specificity = 0,1,0,0 */
style=""     {} /* a=1 b=0 c=0 d=0 -> specificity = 1,0,0,0 */
```

CSS

Asignación de valores

- Dada una propiedad de un elemento, el mecanismo de cascada determina su valor de la siguiente forma (cont):
 - En caso de empate la que esté definida más tarde en la hoja de estilos tiene preferencia
- Si como resultado de esta cascada obtenemos un valor, se debe usar el mismo
- Si no es así debemos recurrir a la **herencia**
- Si la propiedad se hereda y el elemento no es la raíz, se toma el valor de la propiedad del elemento padre
- En caso contrario, se usa el **valor inicial** de la propiedad

CSS

Ejemplo

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" >
    <Title>Ejemplo de estilos</Title>
    <link rel="stylesheet" title="Nombre de la hoja CSS" type="text/css" href="transpas.css">
  </head>
  <body>
    <h1>Este será el título de la página </h1>
    <h2>Párrafos</h2>
    <div class="especial" >
      <p>
        Este es el primer párrafo. Aquí hay una ruptura de línea
        <br>
        de texto
      </p>

    </div>
    <p class="borde">
      Este texto tiene borde
    </p>
    <p class="especial">
      Este párrafo tiene letra más grande. Y esto será un enlace: <a href=" http://javaHispano.org"> javaHispano.org</a>
    </p>
    <p id="enorme">
      Esto se verá enorme.
    </p>
```

CSS

Ejemplo

Una lista; cómo es desordenada tendrá color de fondo:

```
<ul>
  <li>
    Primer elemento
  </li>
  <li>
    Segundo elemento
  </li>
  <li>
    Tercer elemento
  </li>
</ul>
```

Una lista ordenada:

```
<ol>
  <li>
    Primer elemento
  </li>
  <li>
    Segundo elemento
  </li>
  <li>
    Tercer elemento
  </li>
</ol>
```

```
<h1>Encabezado 1</h1>
<h2>Encabezado 2</h2>
<h3>Encabezado 3</h3>
<h4>Encabezado 4</h4>
<h5>Encabezado 5</h5>
<h6>Encabezado 6</h6>
```

```
</body>
</html>
```

CSS

Ejemplo

```
p /*cambiamos apariencia de los párrafos*/
  font-size: 12pt;
  font-family: arial, helvetica;
  font-weight: normal;
}
h1 /*cambiamos la apariencia de los encabezados de primer nivel*/
  font-size: 36pt;
  font-family: verdana, arial;
  text-decoration: underline;
  text-align: center;
  background-color: Teal;
}

ul > li /*cambiamos la apariencia de los elementos de las listas desordenadas*/
  font-size: 10pt;
  font-family: verdana, arial;
  background-color: #666666;
}

body/*cambiamos la apariencia de todo el cuerpo*/ {
  background-color: #001600;
  font-family: arial;
  color: White;
}
p.especial /*clase que se aplica sólo a los párrafos*/
  font-size: 16pt;
}
#enorme /* mediante identificador*/
  font-size: 28pt;
}
. . . . .
```

CSS

Ejemplo

```
.borde {/*clase para bordes*/
    border-top: solid 2px #A00000;
    border-bottom: outset 3px #00FF00;
    border-left: dotted 2px #A00000;
    border-right: dashed 2px #0000FF;
}
A:link {/*pseudoclase para los enlaces*/
    text-decoration: none;
    color: #0000cc;
}
A:visited {
    text-decoration: none;
    color: #ffcc33;
}
A:hover {
    text-decoration: underline;
    color: #999999;
    font-weight: bold;
}
A:active {
    text-decoration: none;
    color: #ff0000;
}
```

CSS

Ejemplo

Este será el título de la página

Párrafos

Este es el primer párrafo. Aquí hay una ruptura de línea de texto

Este texto tiene borde

Este párrafo tiene letra más grande. Y esto será un enlace: javaHispano.org

Esto se verá enorme.

Una lista; cómo es desordenada tendrá color de fondo:

- Primer elemento
- Segundo elemento
- Tercer elemento

Una lista ordenada:

1. Primer elemento
2. Segundo elemento
3. Tercer elemento

Encabezado 1

Encabezado 2

Encabezado 3

Encabezado 4

Encabezado 5

Encabezado 6

CSS

Preprocesadores

- Algunos inconvenientes de CSS
 - no permite usar variables, cambiar un color supone cambiar todas las apariciones
 - No está estructurado en árbol, dificulta su lectura
 - Navegadores antiguos hace necesario incluir etiquetas específicas (-moz, -webkit, -ms, -o)
- Los preprocesadores utilizan una estructura similar a las hojas de estilo pero la extienden para añadir nuevas funcionalidades.
- Mediante una compilación genera la hoja de estilo final a incluir en la página web.

CSS

Preprocesadores

- Permite usar variables

- Permite crear *mixings*

```
@azul: #0023BB;

.documento {
    color: @azul;
}

.borde-redondo(@radio) {
    -webkit-border-radius: @radio;
    -moz-border-radius: @radio;
    border-radius: @radio;
}

.caja-redonda {
    .borde-redondo(4px);
}
```

- Estructuración

```
p {
    font-size: 12px;
    color: #333;
    .azul {
        color: @azul;
        font-weight: bold;
    }
}

p {
    font-size: 12px;
    color: #333;
}

p.azul {
    font-size: 12px;
    color: #0023BB;
    font-weight: bold;
}
```

- Preprocesadores: LESS, SASS, Stylus

CSS

Bootstrap

- **Bootstrap** es un *framework* para desarrollar Web adaptables a distintos dispositivos (*responsive*)
- Se basa en la definición de una serie de **clases** para aplicar a los elementos
- Esto permite crear botones, etiquetas, menús, etc. Fácilmente y con posibilidad de adaptarse
- La adaptabilidad se basa en una matriz con 12 columnas
- Incluye fuentes con multitud de iconos (*glyphicon*)

Para ampliar conocimientos

- Especificación CSS 2.1:
<http://www.w3.org/TR/CSS2>
- Bootstrap
<http://getbootstrap.com>