



FENW	miw.etsisi.upm.es • 2
<h2>Índice</h2>	
<ul style="list-style-type: none"> • CSS2: Introducción <ul style="list-style-type: none"> ◦ Atributos HTML para CSS ◦ Reglas ◦ Selectores ◦ Clases e identificadores ◦ Incorporación de CSS • Propiedades <ul style="list-style-type: none"> ◦ Fuentes, texto ◦ Fondo, colores • Modelo de caja <ul style="list-style-type: none"> ◦ Propiedades de margen, relleno, bordes y contorno ◦ Propiedades de dimensiones, listas y tablas • Herencia y Cascada 	<ul style="list-style-type: none"> • CSS3: <ul style="list-style-type: none"> ◦ Introducción ◦ Nuevos selectores ◦ Colores y gradientes ◦ Bordes, contornos y fondo ◦ Sombras y columnas ◦ Transformaciones ◦ Transiciones ◦ Animaciones

Introducción : Hojas de Estilo en Cascada CSS2.1

- Las hojas de estilo en cascada (*Cascading Style Sheets, CSS*) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML
- Permiten **separar** la estructura de un documento de su presentación, es decir, el diseño del contenido
- Permiten generar “imagen de marca”.
- Existen pequeñas diferencias entre los distintos navegadores.

Introducción : Hojas de Estilo en Cascada CSS2.1

- Permite aplicar al documento formato de modo mucho más exacto:
 - Podemos definir la distancia entre líneas del documento, párrafo, etc.
 - Se puede aplicar indentado, color de fondo, imagen de fondo, etc., a las primeras líneas del párrafo
 - Podemos colocar elementos en la página con mayor precisión
 - Definir la visibilidad de los elementos, márgenes, subrayados, tachados...
 - Medidas en: píxels (px), porcentaje (%), pulgadas (in), puntos (pt), centímetros (cm)

Introducción : Hojas de Estilo en Cascada CSS2.1

- Documentación manejada:

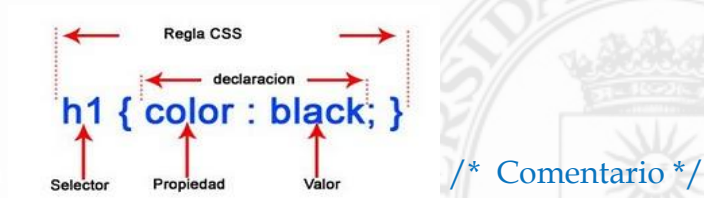
- Estándar CSS2.1: <http://www.w3.org/TR/2004/CR-CSS21-20040225/>
- Traducción del estándar CSS2.1: http://www.w3.org/Style/css21-updates/css2.1_spa.pdf
- Tutorial en inglés con ejemplos <http://www.w3schools.com/>
- Tutorial en castellano <http://www.programacion.com/html/tutorial/csspractico/>
- Tutorial en castellano <http://www.tejedoresdelweb.com/w/CSS>
- Dynamic HTML: the definitive reference *Danny Goodman* O'Really 2006
- Vadidador CSS2.1: <http://jigsaw.w3.org/css-validator/>

Elementos y Atributos HTML para CSS 2.1

- Elementos HTML:
 - '**style**': que sirve para definir una hoja de estilo dentro de un documento HTML
 - '**div**' y '**span**': no imponen una semántica de presentación; se usan para agrupar y estructurar un documento:
 - '**div**' es un elemento de nivel de **bloque** (ej, '**p**', '**h1**', '**ol**')
 - '**span**' es un elemento de nivel de **línea** (ej, '**a**', '**img**', '**b**')
 - '**link**': permite importar hojas de estilo desde un fichero
- Atributos HTML:
 - "**style**": permite aplicar estilos directamente a un elemento
 - "**class**": permite definir una hoja de estilo que podrá ser aplicada a distintos elementos
 - "**id**": sirve para identificar un elemento y aplicar un estilo definido expresamente para ese elemento

Reglas de estilo CSS 2.1

- Una regla de estilo se compone de:
 - Un selector que indica que parte del documento va a aplicar la regla. Los más sencillos son etiquetas HTML.
 - Un conjunto de declaraciones separadas por el carácter “;”
 - Cada declaración es un par (separado por el carácter “:”) formado por el nombre de la propiedad y el valor o valores de la misma.



CSS 2.1: Selector de etiquetas HTML

- Mediante los selectores asociamos partes de un documento HTML con los estilos que les queremos aplicar
- Un selector es el nombre de un elemento HTML


```
hr { width: 200px; }
p { font-family: arial narrow; }
```
- Se pueden agrupar como una lista separada por comas:


```
h1, h2 { font-family: arial narrow }
```
- El selector Universal (“*”) representa a cualquier elemento


```
* { font-family: sans-serif }
```

CSS 2.1: Selector de clase e identificador

- Las **clases** permiten definir un estilo y aplicarlo a cualquier elemento. También se pueden definir **clases particulares** aplicables sólo al **elemento** sobre el que se define

```
.enrojo {color: red; }
p.centrado {text-align: center;}
<b class="enrojo">texto en rojo</b>
<p class="centrado">párrafo centrado</p>
```

- Mediante las clases **alternativas** se pueden aplicar más de una clase a un elemento. Los conflictos se resuelven con la cascada: se aplica la última regla declarada ('escrita')

```
<span class="clase1 clase2">hola</span>
```

- Se pueden definir estilos para un elemento específico por medio de un identificador

```
#bordes {border: 2px solid black;}
p#resaltado { font-weight: bold; }

<p id="resaltado">párrafo<p>
```

Incorporación de hojas de estilo CSS 2.1

- Principalmente, existen 3 maneras para incorporar estilos a las etiquetas HTML:

- Directamente en la etiqueta (estilo en línea)

```
<p style="color:red">
```

- Definiendo una hoja de estilo en la cabecera del propio documento(interna)

```
<style> p{color:red} </style>
```

- Incorporando una hoja de estilo descrita en otro documento(externa)

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

FENW
miw.etsisi.upm.es • 11

CSS 2.1: Propiedades relacionadas con fuentes

font-size:	<p>Tamaño de la letra</p> <pre><porcentaje> <medida> xx-small x-small small medium large x-large xx-large smaller larger</pre>
font-weight:	<p>Peso de la letra</p> <p>En entero (Normal=400, Bold=700)</p> <pre>normal bold bolder lighter 100 200 300 400 500 600 700 800 900</pre>
font-style:	<p>Estilo del tipo de letra.</p> <pre>normal italic oblique</pre>

FENW
miw.etsisi.upm.es • 12

CSS 2.1: Propiedades relacionadas con fuentes

font-family:	<p>Familia de tipos de letra normal o genérica</p> <pre>[[<nombre-de-familia> <familia-genérica>],]* [<nombre-de-familia> <familia-genérica>]</pre>
font-variant:	<p>Con el valor small-caps las mayúsculas ocupan lo mismo que las minúsculas</p> <pre>normal small-caps</pre>
font:	<p>Genérico para definir los anteriores en una sola propiedad</p> <pre>[font-style font-variant font-weight]? font-size[/line-height]? font-family]</pre>

CSS 2.1: Unidades de medida

▪ Absolutas

pt	Puntos
pc	Picas
mm	Milímetros
cm	Centímetros
in	Pulgadas

▪ Relativas

px	Pixels
ex	Altura de la letra x
em	Altura de la fuente del elemento

CSS 2.1: Propiedades relacionadas con texto

text-decoration:	Estilo especial para el texto none [underline overline line-through blink] ('blink' no funciona en IE)
text-transform:	Transformación de texto (conversión a mayúsculas, etc.) capitalize uppercase lowercase none
text-align:	Alineación horizontal de elementos de texto left right center justify
text-indent:	Sangrado de la primera línea de texto <medida> <porcentaje>
direction:	Dirección de escritura del texto ltr rtl

CSS 2.1: Propiedades relacionadas con texto

letter-spacing:	Permite definir el espaciado entre las letras normal <medida>
word-spacing:	Separación entre palabras normal <medida>
white-space:	Tratamiento de los espacios en blanco normal pre nowrap
line-height:	Distancia entre líneas de un párrafo normal <medida> <porcentaje>
vertical-align:	Alineación vertical de los elementos con respecto al elemento padre de los mismos baseline sub super top text-top middle bottom text-bottom <porcentaje> <medida>

CSS 2.1: Propiedades de color y fondo

color:	Color de primer plano del elemento html expresados en rgb: #RRGGBB, yellow,..., rgb(0-255, 0-255, 0-255), rgb(0-100%, 0-100%, 0-100%) <color>
background-color:	Color de fondo de un elemento html. <color> transparent
background-image:	Imagen de fondo de un elemento html none url("imagen.jpg")
background-repeat:	Forma en que se repite una imagen en el fondo repeat repeat-x repeat-y no-repeat

CSS 2.1: Propiedades de color y fondo

background-position: Indica la posición dentro del contenedor donde se sitúa la imagen de background

```
[[<porcentaje> | <medida> ]{1,2} |
[[left | center | right] ||
[top | center | bottom]] ]
```

background-attachment: Indica si la imagen se mantiene fija cuando se realice un scroll (valor **fixed**) o no (valor **scroll**)

```
scroll | fixed
```

background:

Es una propiedad resumen de las anteriores

```
[background-color || background-image ||
background-repeat || background-attachment ||
background-position]
```

CSS 2.1: Modelo de formato basado en cajas

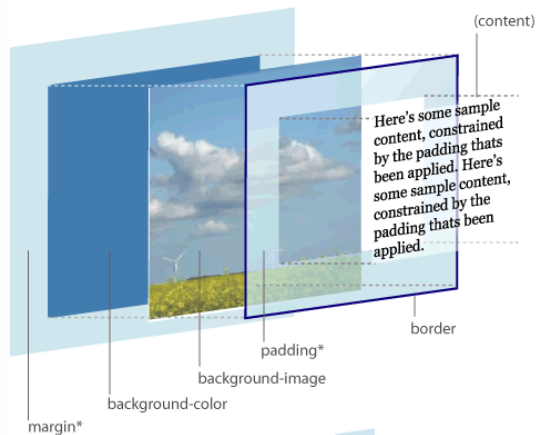


relleno

margen

borde

THE CSS BOX MODEL HIERARCHY



<http://www.hicksdesign.co.uk/boxmodel/>

* transparent elements

FENW

miw.etsisi.upm.es • 19

CSS 2.1: Propiedades de margen y relleno

margin:

4 valores que indican el tamaño del margen por arriba, a la derecha, abajo y a la izquierda. Pueden fijarse por separado a través de las propiedades *margin-top*, *margin-right*, *margin-bottom* y *margin-left*.
[<medida>| <porcentaje>] {1,4}

padding:

4 valores que indican el tamaño del relleno por arriba, a la derecha, abajo y a la izquierda. Pueden fijarse por separado a través de las propiedades *padding-left*, *padding-right*, *padding-top* y *padding-bottom*.
[<medida>| <porcentaje>] {1,4}

FENW

miw.etsisi.upm.es • 20

CSS 2.1: Propiedades de bordes

border-style:

Estilo del borde del elemento
[hidden | solid | dotted | dashed | double | groove | ridge | inset |outset]{1,4}

border-width:

Anchura del borde. También con *border-top-width*, *border-left-width*, *border-rigth-width*, *border-bottom-width*
[<medida> | <porcentaje> | thin | medium | thick]{1,4}

border-color:

Color del borde
<color>{1,4} | transparent

border:

Propiedad resumen de los anteriores valores de borde
border-width border-style border-color

CSS 2.1: Propiedades de contorno

outline-style:	Estilo del contorno del elemento [hidden solid dotted dashed double groove ridge inset outset]{1,4}
outline-width:	Anchura del contorno [<medida> <porcentaje> thin medium thick]{1,4}
outline-color:	Color del contorno <color>{1,4} transparent
outline:	Propiedad resumen de los anteriores valores de contorno outline-width outline-style outline-color

CSS 2.1: Propiedades de dimensiones

width:	Anchura de la zona del elemento <medida> <porcentaje> auto
min-width:	Anchura mínima de la zona del elemento <medida> <porcentaje> none
max-width:	Anchura máxima de la zona del elemento <medida> <porcentaje> none
height:	Altura de la zona del elemento <medida> <porcentaje> auto
min-height:	Altura mínima de la zona del elemento <medida> <porcentaje> none
max-height:	Altura máxima de la zona del elemento <medida> <porcentaje> none
overflow:	Indica qué ocurre con el contenido de un elemento si es mayor que el área de propio elemento visible hidden auto scroll

CSS 2.1: Propiedades de listas

list-style-type:	Viñeta de acompañamiento del elemento de la lista disc circle square armenian cjk-ideographic georgian hebrew decimal decimal-leading-zero hiragana hiragana-iroha katakana katakana-iroha lower-alpha lower-greek lower-latin lower-roman none upper-alpha upper- latin upper-roman
list-style-position:	El valor <i>inside</i> hace que la sangría sea francesa. inside outside
list-style-image:	Permite definir una imagen como viñeta. url("imagen.jpg")
list-style:	Propiedad resumen de listas list-style-type list-style-position list-style-image

CSS 2.1: Herencia

- Se puede considerar una relación padre-hijo para las etiquetas que están contenidas dentro de otras:

```
<div>
  <h3>Bondades de CSS?</h3>
  <p>CSS proporciona un <i>control preciso</i> del
    aspecto de las páginas web</p>
</div>
```

- Relaciones familiares:

- h3 y p son **hijos** de div
- h3, p e i son **descendientes** de div
- El **padre** de h3 y p es div
- La etiqueta i es **hija** de p, o sea, su **padre** es p
- La etiqueta i NO es una hija de div, es nieta o **descendiente**
- div es un **ancestro** de H3, p e i
- h3 y p son **adyacentes**

CSS 2.1: Herencia

- Es la transmisión de características de un elemento a sus descendientes.
 - Sí se heredan: fuentes, alineaciones, colores de fondo y de primer plano...
 - NO se heredan: dimensiones (anchos y altos), márgenes, rellenos (padding), bordes...
- Las propiedades son transmitidas a todos los descendientes directos e indirectos
- La herencia se anula definiendo la propiedad que no se quiere heredar
- El valor **inherit** indica que la propiedad es heredada del antecesor más cercano

CSS 2.1: Selectores relacionales

- Selector **descendente** (**A B**): afecta al elemento **B** que está contenido dentro de otro elemento.


```
p em {text-decoration: underline}
```

`<p>hola</div>` ☺

`<p><a >...hola...</p>` ☺
- Selector **hijo** (**A > B**): afecta únicamente a los **B** que son hijos directos de **A**

```
p > em {text-decoration: underline}
```

`<p>hola</p>` ☺

`<p><a>hola</p>` ☹
- Selector **hermano adyacente** (**A + B**): afecta a todos los elementos que **B** comparten padre con **A** y que aparecen justo después de **A**

```
h1 + h2 {margin-top: 1em; }
```

CSS 2.1: Selectores de atributos

- CSS2.1 permite definir reglas que se aplicarán a aquellos elementos que tengan declarado un determinado atributo e, incluso, con un determinado valor
 - **[atrib]**: cualquier elemento con el atributo 'atrib' declarado


```
img[alt] {border: 1px solid red; }
```
 - **[atrib="valor"]**: elementos con el atributo 'atrib' declarado y cuyo valor es 'valor'


```
td[colspan="5"] {background-color: blue;}
```
 - **[atrib~="palabra"]**: elementos con el atributo 'atrib' y cuyo valor es una lista de palabras, separadas por blancos, que contiene la palabra 'palabra'


```
a[title~="mensaje"] {text-decoration: none; }
<a href='...' title='envíame un mensaje'>.....
```
 - **[atrib|="palabra"]**: elementos en el que 'atrib' es una lista de palabras, separadas por guiones, que comienza con la palabra 'palabra'


```
p[title|"parrafo"] { color: red; font-style: italic; }
<p title="parrafo-tipo-1">Párrafo 1</p> ☺
<p title="parrafo 2">Párrafo 2</p> ☹
```

pseudo-clases y pseudo-elementos

- Las pseudo-clases son aquellas partes de un documento que no existen como elementos html porque son elementos en un estado particular (:active, :link, :visited, :hover, :focus) o son parte de otros (:first-child)
- Pseudo-clases **dinámicas**:
 - **:link** estado de enlace sin visitar
 - **:visited** estado de enlace visitado
 - **:active** estado cuando se pulsa sin soltar en el enlace
 - **:hover** estado cuando el ratón se encuentra encima del elemento
 - **:focus** estado cuando un elemento de formulario tiene el foco
- Pseudo-clases **estructurales**: el selector **:first-child** hace referencia al primer descendiente inmediato, al primer hijo, de un elemento


```
p:first-child {font-size: 110%;}
```
- Los pseudo-elementos sirven para aplicar reglas a aquellas partes del documento que no existen en el código html, por ej, la primera línea de un párrafo
 - El pseudo-elemento **:first-line** identifica a la primera línea de texto visual


```
p:first-line {text-transform: uppercase; }
```
 - El pseudo-elemento **:first-letter** identifica a la primera letra de cualquier elemento


```
p:first-letter {font-size: 200%; color: red;}
```

CSS 2.1: Cascada

- Conjunto de criterios para decidir cómo se aplican reglas CSS a los elementos en caso de **conflictos** entre ellas; establece prioridades en la aplicación de las reglas.
- Las hojas de estilo pueden ser definidas por: el *autor*, el *usuario (lector)*, el *Agente de Usuario (navegador)*.
- Para resolver los conflictos se aplica 1) la procedencia, 2) la especificidad y, por último, si es necesario, 3) el orden
- **Procedencia**: las reglas de estilos se aplican según el orden de prioridad:
 - a) Primero las que el *usuario* haya definido como importantes (**!important**). Ej: `p{color:red ; border:1px solid red !important}`
 - b) Las que el *autor* haya definido como importantes (**!important**)
 - c) Los estilos del *autor*
 - d) Los estilos de *usuario*
 - e) Los estilos del *agente de usuario* (navegador)

CSS 2.1: Cascada

- **Especificidad**: se mide por medio de cuatro valores separados por comas (**/a, b, c, d/**), el número a la izquierda pesa más, por lo tanto, dota a la regla de mayor prioridad en la aplicación del estilo. Estos toman los valores:
 - a: número de estilos en línea aplicados al elemento atributo **style**
 - b: número de selectores de la regla de estilo que son un **id**.
 - c: número de selectores **class**, un **selector de atributo** o una **pseudoclase**
 - d: número de selectores que son un **elemento** o un **pseudoelemento**

```

p{ ... } /* 0,0,0,1 */
body p{ ... } /* 0,0,0,2 */
.intro{ ... } /* 0,0,1,0 */
p.intro{ ... } /* 0,0,1,1 */
#contenido{ ... } /* 0,1,0,0 */
#contenido p{ ... } /* 0,1,0,1 */
div#contenido p{ ... } /* 0,1,0,2 */
div#contenido p.intro{ ... } /* 0,1,1,2 */
body div#contenido p.intro{ ... } /* 0,1,1,3 */
body div#contenido p.intro:hover{ ... } /* 0,1,2,3 */
body div#contenido p.intro:first-line{ ... } /* 0,1,1,4 */

```
- **Orden**: Si siguen existiendo conflictos, se aplicará el orden de definición: se utilizará aquella regla que se halla **definido** en **último lugar**

CSS 2.1: Propiedades de posicionamiento flotante

- Un elemento con la propiedad `float` no sigue el flujo de disposición del texto en el navegador: el texto que lo rodea flotará a su alrededor (al lado 'contrario' del valor)
 - valores: `left`, `right`, `none`
 - Cualquier elemento (de bloque o no) puede flotar en la página
 - Al asignarle `float` lo estamos convirtiendo en una caja...
 - ...que debe tener definida su anchura (expresa o implícitamente), si no, será tratado como no flotante
- `clear` Indicar cuándo un elemento permite ser invadido por un elemento flotante anterior a él
 - `none`: el elemento permite otros flotantes a ambos lados
 - `left`: el elemento no permite flotantes a su izquierda
 - `right`: el elemento no permite flotantes a su derecha
 - `both`: el elemento no permite flotantes en ningún lado

CSS 2.1: Propiedades de posicionamiento absoluto

- Con la propiedad `position` podemos sacar al elemento fuera del flujo habitual de disposición (*layout*). Definiendo un marco de referencia u origen de coordenadas sobre el que se define un desplazamiento (`top`, `left`, `right` y `bottom`). Valores:
 - `static`: flujo normal (el inicial por defecto)
 - `absolute`: El elemento toma como referencia el elemento antecesor más cercano que se encuentre posicionado. Si no existe ningún elemento anterior posicionado se toma como referencia a `body`
 - `relative`: El origen de coordenadas del elemento es la posición del elemento en el flujo normal
 - `fixed`: el posicionamiento será fijo, con el marco de referencia es la ventana de visualización.
- La posición de los elementos absolutos se fija determinando su coordenadas, dadas por los atributos `top`, `left`, `right` y `bottom`. Valores:
 - <porcentaje> | <medida> | auto

CSS 2.1: visibilidad, recorte y profundidad

- La propiedad `visibility` determina si un elemento es visible o no; aunque esté oculto, ocupa el mismo espacio.
visible | hidden | collapse
- Con `clip` decidimos qué parte/área de un elemento será visible. No afecta al espacio ocupado por el elemento en el flujo HTML. Sintaxis: `rect(top right bottom left)`
 - ej: `clip:rect(20px 30px 40px 10px)`; recortará al elemento creando un cuadro visible cuya esquina superior izquierda está a 10 pixels por la izquierda y 20 por arriba de la de la capa y cuyo tamaño sería de 30-10 de ancho y 40-20 de alto
- **Profundidad:** cuando dos elementos se **superponen**, se mostrará aquel que tenga el valor más alto en su propiedad `z-index`. Por defecto, los elementos se apilan en el orden en que aparecen en el código html: el último más arriba

CSS 2.1: Propiedades display y cursor

- `display` determina cómo se muestra un elemento
inline | block | list-item | run-in | inline-block | table | inline-table | table-row-group | table-header-group | table-footer-group | table-row | table-column-group | table-column | table-cell | table-caption | none
 - **block:** el elemento se mostrará como uno de nivel de bloque con sendos cambios de línea antes y después del mismo
 - **inline:** el elemento se muestra como uno de nivel de línea
 - **none:** no muestra el contenido, **ni ocupa espacio**
 - **list-item:** los elementos se muestran como items de una lista
 - **inline-block:** se sitúa como uno en línea (en la misma línea y a continuación) pero se comporta como un bloque
- `cursor` determina el aspecto del cursor
[[url(curosor.cur)],]* [auto | crosshair | default | pointer | move | e-resize | ne-resize | nw-resize | n-resize | se-resize | sw-resize | s-resize | w-resize | text | wait | help | progress]]

CSS 2.1: hojas de estilo específicas

- Podemos diseñar e indicar estilos que sólo van a ser aplicados en determinados medios de 'representación'. Valores: screen | print | aural | handheld | tv | tty | braille | embossed | projected
- Formas de indicarlos:
 - Utilizamos el atributo '**media**' en la etiquetas **link** y/o **style**
 - Declarándolos bajo la etiqueta **style: @media [valor] {...}**
- Ejemplos:


```
<link rel='stylesheet' media='print'... />
<style type='text/css' media='tv'>..</style>
<style type='text/css'>
  @media print { ... }
  @media screen { ... }
</style>
```

CSS 2.1: Características de pantalla

- Sintaxis completa:


```
<link rel="stylesheet" media="mediatype and|not|only
(característica)" href="url_fichero.css">
ó bien:
@media [not|only] mediatype and (característica) {
  ...
}
```
- ```
<style>
div{width:100px;height:100px}
#div1{position:absolute;top:100px;left:10px;
background-color:yellow}
@media (max-width: 400px) {
 #div1 { background-color: blue }
}
@media (min-width: 400px) and (max-width: 600px) {
 #div1 { background-color: red }
}
@media (min-width: 600px){
 #div1 { background-color: green }
}
</style>
```

## Uso de fuentes externas (I)

- La regla `@font-face` permite la utilización, a través de su descarga, de fuentes que el usuario puede no tener instaladas.
- Es antigua pero no tenía soporte real hasta hace poco
- Sintaxis:
 

```
@font-face{
 font-family: identificador_a_usar;
 src: url('url_de_la_fuente');
 [font-weight: peso;]
 [font-style: estilo;]
}
```
- El `identificador_a_usar` es la referencia para usar posteriormente en los elementos a los que se les asigna
- Dependiendo de los navegadores, se soportan unos formatos u otros, pero en general, ".ttf" (true type) y ".otf" (open type PS) son los más soportados

## Uso de fuentes externas (II)

- Ejemplo

```
<style>
 @font-face{
 font-family: olympic;
 src: url('Olympic Branding.ttf')
 }
 h1 {
 font-family: "olympic", "Times", serif;
 font-size: 6.8em;
 }
</style>
```

- El identificador "olympic" es un identificador elegido por el diseñador del documento CSS
- La URL puede ser local o externa
- El navegador será capaz de representarla sin necesidad de que el usuario se la instale en su equipo

## Uso de fuentes externas (III)

- Google pone a disposición de los creadores de documentos CSS un amplio conjunto de fuentes que se pueden utilizar accediendo a ellos a través de la api proporcionada.
- El conjunto de fuentes puede ser consultada/seleccionada desde:
  - <https://www.google.com/fonts/>
- Ejemplo:
 

```
<link rel="stylesheet" type="text/css" href="http://fonts.googleapis.com/css?family=Lobster">
h1 {
 font-family: "Lobster", "Times", serif;
 font-size: 1.8em;
}
```
- Se pueden incorporar también con
 

```
@import url(http://http://fonts.googleapis.com/css?family=Lobster")
```
- Una vez importadas se pueden utilizar con el identificador asociado a cada una de ellas

## Introducción : Hojas de Estilo en Cascada CSS3

- Las hojas de estilo en cascada (*Cascading Style Sheets*, CSS) versión 3 **ampliarán** las posibilidades de la versión 2.1
- Se está trabajando en un estándar que se ha dividido en partes independientes. Sólo está aprobada la parte dedicada a la definición de colores.
- Algunas propiedades solo funcionan, de momento, con un prefijo de vendedor:
  - **-moz-** para Firefox
  - **-o-** para Opera
  - **-webkit-** para Safari y Chrome
  - **-ms-** para IE

## Introducción : Hojas de Estilo en Cascada CSS3

- Se proponen nuevas posibilidades en todos los ámbitos:
  - Nuevos valores para las propiedades que permiten manejar colores, gradientes, grados, segundos, milisegundos...
  - Nuevos selectores para identificar la parte del contenido al que se le aplican las reglas: relacionales, de atributos, pseudo-elementos, pseudo-clases...
  - Nuevas propiedades para definir transiciones, bordes y contornos, fondos, transformaciones, trabajo con columnas,...

## Introducción : Hojas de Estilo en Cascada CSS3

- Documentación manejada:
  - **Estándar de partes de CSS 3.0** [http://www.w3.org/TR/#tr\\_CSS](http://www.w3.org/TR/#tr_CSS)
  - Tutorial en inglés con ejemplos <http://www.w3schools.com/css/default.asp/>
  - Validador CSS3(activando opciones): <http://validator.w3.org/>
  - Generador de CSS3 <http://css3generator.com/>
  - Manual de CSS3 <http://www.desarrolloweb.com/css/>
  - Tutorial, diseños CSS3, herramientas... <http://www.cssportal.com/>
  - Diseños con CSS <http://www.csszengarden.com>
  - Soporte de CSS3 por los Navegadores [http://en.wikipedia.org/wiki/Comparison\\_of\\_layout\\_engines\\_%28CSS%29](http://en.wikipedia.org/wiki/Comparison_of_layout_engines_%28CSS%29)

## CSS3: Selectores relacionales y de atributos

- Nuevo selector **relacional**:
  - Selector **hermano general**(A ~ B): afecta a todos los elementos que B comparten padre con A
 

```
h1 ~ h2 {margin-top: 1em;}
```
- Nuevos selectores de **atributos**
  - **[atrib^=valor]**: Un elemento con el atributo 'atrib' declarado y cuyo valor comienza por 'valor'
 

```
img[alt^=imagen] {border: 1px solid red; }
```
  - **[atrib\$=valor]**: Un elemento con el atributo 'atrib' declarado y cuyo valor acaba por 'valor'
 

```
img[alt$=fin] {margin-top: 1em; }
```
  - **[atrib\*=valor]**: Un elemento con el atributo 'atrib' declarado y cuyo valor contiene la cadena 'valor'
 

```
img[alt*=cadena] {border: 2px solid blue; }
```

## CSS3: Selectores pseudoclasas nuevas

- **:enabled** representa a un elemento de la interfaz que está habilitado. Por ejemplo un elemento **input**.
- **:disabled** representa a un elemento de la interfaz que está deshabilitado.
- **:target** representa el destino de un enlace interno.
- **:checked** representa a radiobutton o checkbox chequeados (sólo en Opera).
- **:valid** representa a un elemento de formulario cuyo contenido es válido.
- **:invalid** representa a un elemento de formulario cuyo contenido no es válido.
- **:required** representa a un elemento de formulario con este atributo.
- **:optional** representa a un elemento de formulario sin el atributo **required**

## CSS3: Colores

- CSS3 ofrece nuevos formatos para los colores
  - Se introduce el formato `hsl(hue, saturation, lightness)` siendo:
    - El primer valor indica en grados (0-360deg) el color dentro del círculo de color (0=red, 60=amarillo, 120=verde...).
    - El segundo valor indica la saturación del color (100% saturación al máximo, 0% sin saturación, es decir, en la escala de grises)
    - El tercer valor representa en porcentaje la luminosidad (100% White, 50% el color actual y 0% negro)`hsl(30, 50%, 70%)`
  - Es posible definir el nivel de opacidad (valor decimal típicamente entre 0.0 y 1.0) de un color añadiendo un nuevo parámetro al formato `rgb()` ó `hsl`, que ahora son `rgba()` y `hsla()`
`rgba(120, 70, 30, 0.7)`      `hsla(30, 50%, 70%, 0.1)`

`opacity` define el nivel de opacidad (0.0 y 1.0), pero en este caso se aplica al elemento y a todos sus hijos

## CSS3: Gradientes lineales

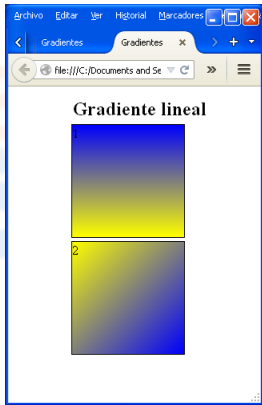
- Es un valor para la propiedad `background`
- CSS3 permite definir un gradiente lineal que es la transición se realiza a lo largo de un vector de dos ó mas colores. Se indica el color de comienzo, uno o más colores de finalización a lo largo del vector y el navegador dibuja líneas perpendiculares al vector que realizan una transición suave entre los colores. En IE se debe hacer con un filtro. En el resto de navegadores es necesario anteponer el **prefijo** del navegador (`-moz-`, `-o-`, `-webkit-`). Los parámetros son los siguientes:
 

```
linear-gradient([[top | bottom] || [left | right]] |
<angle>[,]? <color-stop>[, <color-stop>]+)
```

  - Primero, se indica el vector por medio de la combinación de los valores (`[[top | bottom] || [left | right]]`) ó indicando un ángulo por medio de un valor entre 0 y 360 acompañado de la unidad `deg`. Si no se indica este parámetro el gradiente es vertical de arriba abajo.
  - Segundo, se indica una lista separada por comas de al menos dos parejas `<color-stop>`. Este parámetro puede estar formado por un color ó por un color y una posición de parada expresada en %.
- Otra forma de incluir gradientes es por medio de la definición de un fichero SVG (Scalable Vector Graphics) que funciona en todos los navegadores.
- Se pueden repetir por medio de `repeating-linear-gradient`

FENW
miw.etsisi.upm.es • 47

## Gradientes: Ejemplo



```

<style>
 h2{text-align:center;margin:6px}
 p{width:140px; height:140px;margin:4px;border:1px solid black}
 .div1{position:absolute;left:75px;}
 #lineargradient1{
 background: linear-gradient(90deg, yellow , blue);
 background: -o-linear-gradient(90deg, yellow, blue);
 background: -moz-linear-gradient(90deg, yellow, blue);
 background: -webkit-linear-gradient(90deg, yellow, blue);
 }
 #lineargradient2{
 background: linear-gradient(top left, yellow, blue);
 background: -o-linear-gradient(top left, yellow, blue);
 background: -moz-linear-gradient(top left, yellow, blue);
 background: -webkit-linear-gradient(top left, yellow, blue);
 }
</style>
</head>
<body>
 <div class="div1">
 <h2>Gradiente lineal</h2>
 <p id="lineargradient1">1</p>
 <p id="lineargradient2">2</p>
 </div>
</body>

```

FENW
miw.etsisi.upm.es • 48

## CSS3: Propiedades de bordes y contornos

**border-radius:** permite definir bordes redondeados. Es la propiedad resumen de **border-top-left-radius**, **border-top-right-radius**, **border-bottom-right-radius** y **border-bottom-left-radius**.

<medida> | <porcentaje>

**outline:** permite definir el aspecto del contorno del elemento que es lo que queda fuera del borde. Tiene el mismo formato que **border**. Se puede definir individualmente por medio de: **outline-width**, **outline-style** y **outline-color**.

<width> <style> <color>

**border-image:** Permite definir una imagen como borde. Los valores indican: la url de la imagen, el desplazamiento interno del borde, el tamaño del borde, lo que sobresale la imagen del borde y como se repite la imagen. En IE no funciona y en el resto es necesario utilizar el prefijo de vendedor.

<source> <slice> <width> <outset> <repeat>



## CSS3: Propiedades de background

### background-clip:

permite definir hasta qué caja abarca el *background*.  
padding-box | border-box | content-box

### background-image:

permite definir varias imágenes de fondo superpuestas, siendo la primera que se defina la que se encuentra en primer plano. Si se acompaña de varios valores de [background-position](#), entonces se produce una correspondencia entre imagen y posición de la misma.  
`url("imagen.gif") [, url("imagen.png")] +`

### background-size:

permite definir el tamaño de la imagen. El valor [contain](#) escala la imagen preservando el aspecto de la misma. El valor [cover](#) escala la imagen hasta que cubre todo el elemento por lo que no tienes por qué respetar el aspecto de la imagen.  
`[ <medida> | <porcentaje> | auto ]{1,2} | cover | contain`

## CSS3: Propiedades de sombras

### box-shadow:

permite definir una o varias sombra de la caja de un elemento. El valor [inset](#) indica que la sombra aparece dentro de la caja.

`inset? <medida>{2,4} <color>?  
[, inset? <medida>{2,4} <color>?]*`

Los valores de medida son :

- Es el desplazamiento horizontal de la sombra. *Positivo = Derecha, Negativo = Izquierda*
- Es el desplazamiento vertical de la sombra. *Positivo = Abajo, Negativo = Arriba*
- Define la cantidad de la sombra que se muestra difuminada
- Define la cantidad que se incrementa la sombra

### text-shadow:

permite definir una o varias sombras de un texto. El valor [inset](#) indica que la sombra aparece dentro de la caja.

`<medida>{2,3} <color>?  
[, <medida>{2,3} <color>?]*`

Los valores de medida son :

- Es el desplazamiento horizontal de la sombra. *Positivo = Derecha, Negativo = Izquierda*
- Es el desplazamiento vertical de la sombra. *Positivo = Abajo, Negativo = Arriba*
- Define la cantidad de difuminado con la que se muestra la sombra

## CSS3: Propiedades de columnas

- CSS3 contempla la definición de contenido en varias columnas.
- Las propiedades más importantes son:
  - `column-count` que determina el número de columnas
  - `column-gap` que determina el espacio de separación entre dos columnas expresado como una medida
  - `column-width` que determina el espacio de cada columna. Si entra en conflicto con `column-count` porque en el contenedor no caben el número de columnas especificadas con el tamaño indicado, gana `column-width`.
  - `column-rule` establece una línea de separación entre las columnas. Su formato es similar a `border`. Se puede expresar individualmente como `column-rule-width`, `column-rule-style` y `column-rule-color`
  - `column-break` solo funciona en `-webkit-` y permite determinar cuando ocurre una ruptura de columna (similar a `page-break`)
  - `column-span` solo funciona en `-webkit-` y permite determinar cuando una columna se expande.

## CSS3: Propiedades para transformaciones

- Permiten trasladar, rotar, escalar y “desalinear” un elemento. La propiedad se denomina `transform`, pero de momento es necesario utilizar los prefijos de cada fabricante para que funcione `o-transform`, `-ms-transform`, `-ms-transform` y `-webkit-transform`. Los valores que se puede tomar son:
  - `translate(despx, despy)` es igual que posicionamiento relativo con valores de `left` y `top`. Desplaza el elemento según los valores `despx` (horizontal) y `despy` (vertical) que pueden ser un porcentaje o una medida.
  - `translatex(despx)` para desplazamiento horizontal
  - `translatey(despy)` para desplazamiento vertical
  - `scale(valorx, valory)` escala el elemento según los valores `valorx` (horizontal) y `valory` (vertical) que son valores con punto decimal.
  - `scale(valorx)` para escalado horizontal
  - `scale(valory)` para escalado vertical

## CSS3: Propiedades para transformaciones

- La propiedad `o-transform`, `-moz-transform`, `-ms-transform` o `-webkit-transform` también puede tomar los valores:
  - `rotate(grados)` Rota el elemento según los grados definidos. Los grados se expresan acompañados de la unidad `deg`. Ej: `10deg`
  - `skew(gradosx, gradosy)` permite desalinear un elemento a lo largo de los ejes horizontal (`gradosx`) y vertical (`gradosy`) expresados en grados.
  - `skew(gradosx)` Desalinamiento horizontal.
  - `skew(gradosy)` Desalinamiento vertical.
  - `matrix(a,b,c,d,e,f)` matriz de transformación ver la definición en <http://www.w3.org/TR/SVG/coords.html#TransformMatrixDefinition>
- La propiedad `transform-origin` (con su **prefijo** correspondiente) permite cambiar el origen de la transformación que por defecto está en el centro del elemento. Se puede expresar por medio de un par de medidas, porcentaje o utilizando las palabras reservadas `left`, `center`, `right` y `top`, `center` y `bottom`.

## CSS3: Propiedades para transiciones

- Una **transición** hace que el cambio de una propiedad CSS se realice gradualmente (Ej: el paso de un color a otro o una transformación). Precisa de dos estados: inicial y final. En el estado inicial se define la propiedad que va a transitar, la duración, el retardo antes de comenzar y la función de tiempo de la transición. El estado final se alcanza por un cambio de la propiedad (Javascript o estado CSS). No funcionan en IE y en el resto es necesario utilizar el **prefijo** del fabricante. La propiedad resumen es `transition`:
  - `transition-property` recibe una lista separada por comas de propiedades CSS que van a realizar la transición. En general, pueden transitar las relacionadas con el modelo de caja, colores y background.
  - `transition-duration` establece en segundos(1.3s) o milisegundos(1300ms) la duración de la transición.
  - `transition-timing-function` establece la función de tiempo de aplicación de la transición. Los valores que puede tomar son: `linear`, `ease`, `ease-in`, `ease-out`, `ease-in-out`, `cubic-bezier(n,n,n,n)`.
  - `transition-delay` establece en segundos(1.3s) o milisegundos(1300ms) el retardo antes de comenzar la transición.
- Se pueden realizar múltiples transiciones definiendo una lista de valores para cada una de las propiedades anteriores ó en la propiedad resumen `transition`.

## CSS3: Propiedades para animaciones

- Una **animación** permite establecer puntos intermedios para las transiciones. Consta de las propiedades y de los diferentes estados por los que puede pasar la animación:
  - **animation-name**: nombre de la animación que describe los estados.
  - **animation-duration**: establece en segundos(1.3s) o milisegundos(1300ms) la duración de la animación.
  - **animation-iteration-count**: número de veces que se repite la animación (**infinite** la repite indefinidamente)
  - **animation-direction**: si se establece **alternate** indica que al acabar una animación debe realizarla en el otro sentido
  - **animation-timing-function**: establece la función de tiempo de aplicación de la animación.
  - **transition-delay**: establece el retardo antes de comenzar la transición.

## CSS3: Propiedades para animaciones

- La propiedad **@keyframes** permite definir los estados por los que pasa la animación y definir el nombre que se asigna en la propiedad animation-name
- Hay que utilizar el prefijo del fabricante:
  - **@-moz-keyframes**
  - **@-webkit-keyframes**
- Se define, por lo menos el estado inicial y el final con las palabras "**from**" y "**to**" (ó **0%** y **100%**)
- Se pueden definir tantos estados intermedios como se desee mediante el % de transición

FENW
miw.etsisi.upm.es • 57

## CSS3: Propiedades para animaciones

```

<!doctype html>
<html>
<head> <title>Animaciones CSS3</title>
<style type="text/css">
 h1 {
 position: absolute;
 left:90%;
 -webkit-animation-name: entrada;
 -webkit-animation-duration: 5s;
 }

 @-webkit-keyframes entrada {
 from {
 left:90%;
 }
 to {
 left:0%;
 }
 }
</style>
</head>
<body>
<h1>Me nuevo</h1>
</body>
</html>

```

```

<!doctype html>
<html><head> <title>Animaciones CSS3</title>
<style type="text/css">
 h1 {
 position: absolute;
 left:90%;
 -webkit-animation-name: entrada;
 -webkit-animation-duration: 5s;
 -webkit-animation-iteration-count: infinite;
 -webkit-animation-direction: alternate;
 }

 @-webkit-keyframes entrada {
 from {
 left:90%; }
 50% {
 color:blue;
 }
 to {
 left:0%; }
 }
</style>
</head>
<body>
<h1>Me nuevo</h1>
</body>
</html>

```

FENW
miw.etsisi.upm.es • 58

## Introducción : Bootstrap

Hay que incorporar: CSS y js de bootstrap y jQuery:

```

<html>
 <head>
 <meta charset="utf-8">
 <meta content="viewport" content="width=device-width, initial-scale=1.0">
 <!-- CSS de Bootstrap -->
 <link href="css/bootstrap.min.css" rel="stylesheet" >
 </head>
 <body>
 <p>Aquí iría la información a publicar</p>
 <!-- Se debe incorporar jQuery para los plugins de javascript -->
 <script src="http://code.jquery.com/jquery.js"></script>
 <!-- Y por último, los plugins JavaScript de Bootstrap -->
 <script src="js/bootstrap.min.js"></script>
 </body>
</html>

```

FENW

miw.etsisi.upm.es • 59

## Bootstrap: Rejilla (grid)

- El diseño adaptativo en bootstrap se basa en la utilización de la división del ancho de la pantalla en 12 columnas que pueden ser ocupadas por los elementos del documento.
- El diseño se realiza indicando, para cada elemento, cuántas de esas columnas ocupa mediante la clase correspondiente.
- Se pueden indicar diferentes clases dependiendo de la resolución del dispositivo con el que se esté visualizando la página.
- Resoluciones disponibles:

Resolución	Dispositivo	Clase CSS
< 768px	móviles	.col-xs-
>= 768px	tablets	.col-sm-
>= 992px	Pantallas medianas	.col-md-
>= 1200px	Pantallas grandes	.col-lg-

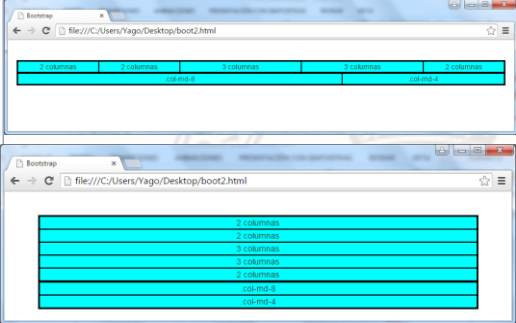
FENW

miw.etsisi.upm.es • 60

## Bootstrap: Rejilla (ii)

- Las rejillas deben estar siempre en un elemento contenedor tipo *container* o *container-fluid*
- La pantalla se organiza por filas, de tal manera que las columnas crean celdas dentro de cada fila. Una fila se crea con un contenedor de clase *.row*
- El máximo de celdas (columnas) en una fila son 12
- Se pueden definir el número de celdas (columnas) que ocupa cada elemento añadiéndole el número al final del nombre de la clase CSS.
- Si se cambia el tamaño de la ventana del navegador, las celdas se reordenan en más filas para mantener una visualización correcta.

```
<div class='container'>
<div class="row">
 <div class="col-md-2">2 columnas</div>
 <div class="col-md-2">2 columnas</div>
 <div class="col-md-3">3 columnas</div>
 <div class="col-md-3">3 columnas</div>
 <div class="col-md-2">2 columnas</div>
</div>
<div class="row">
 <div class="col-md-8">.col-md-8</div>
 <div class="col-md-4">.col-md-4</div>
</div>
</div>
```



## Bootstrap: Rejilla (iii)

- Se pueden definir diferentes clases para cada tipo de dispositivo => Si se visualiza en alguno de ellos se adaptará al tamaño

```
<div class='container'>
 <div class="row">
 <div class="col-xs-12 col-sm-6 col-md-4">Texto publicado</div>
 <div class="col-xs-12 col-sm-6 col-md-4">Texto publicado</div>
 <div class="col-xs-6 col-sm-4 col-md-2">Texto</div>
 <div class="col-xs-6 col-sm-4 col-md-2">Texto</div>
 </div> </div>
```

