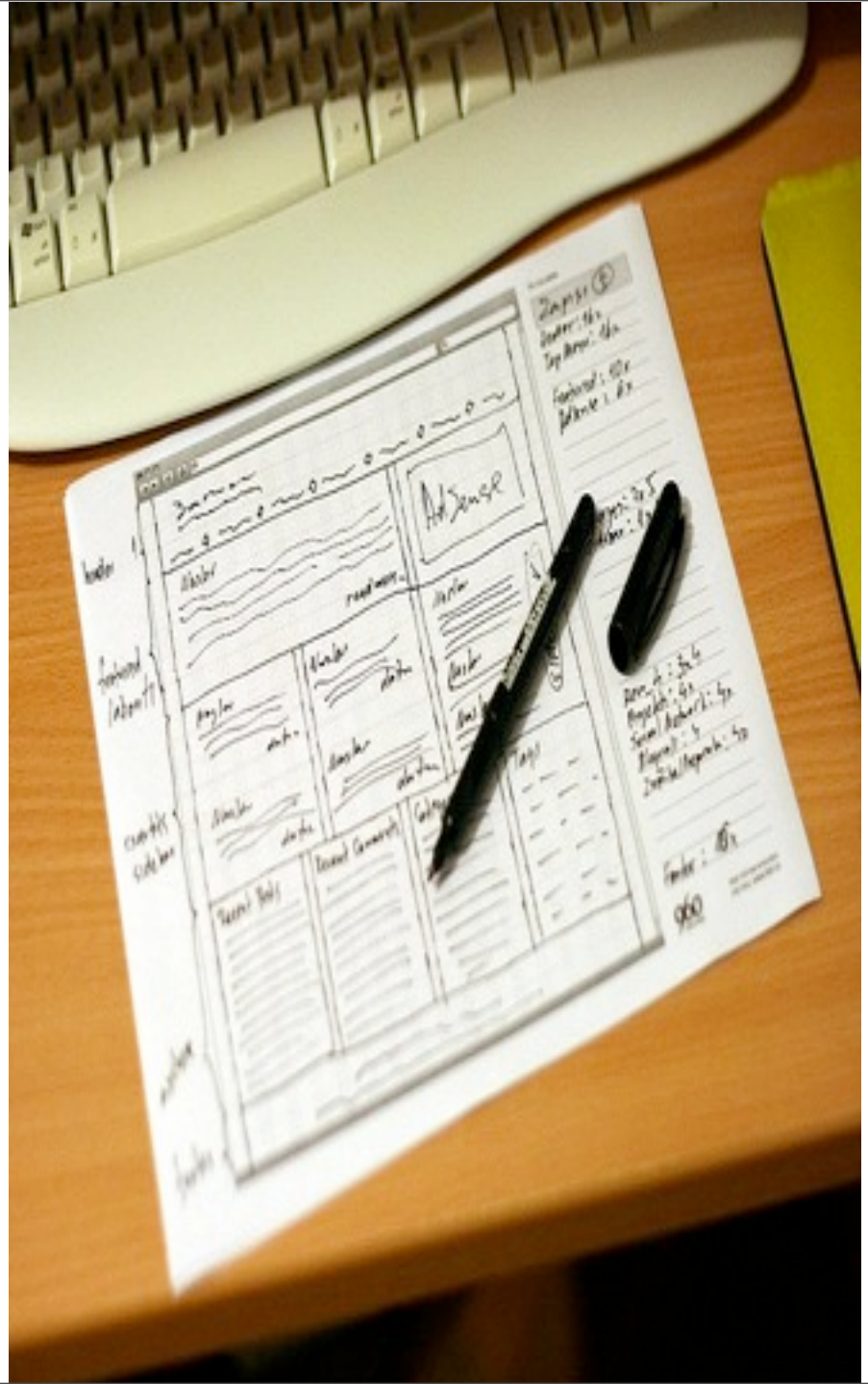


http://www.flickr.com/photos/urke/3223200963

Tema 2b

Hojas de estilo (CSS)

Dando estilo visual a las páginas HTML

Hojas de estilo

- Separar la **estructura** del documento (etiquetas HTML) del estilo de **presentación** (hoja de estilo)
  - Son un asunto más de diseñadores que de desarrolladores, pero hay que conocerlo, ya que de él dependen efectos Javascript como animaciones, etc
- Estándar del W3C: **CSS** (Cascade Style Sheets)
- Versión actual ("en producción"): **CSS2**. La versión 3 está en desarrollo desde hace años
  - Las versiones antiguas de Explorer tenían problemas serios con CSS2 (hasta la versión 6 inclusive)
  - El soporte para CSS3 es todavía parcial en los navegadores. El que va más por detrás es Explorer

Regla de estilo

- Asocia una etiqueta HTML con su aspecto

```
/* Selector  { Propiedades }*/
h1           {color:blue; font-size:25px;}
```

  - Para cada etiqueta HTML el navegador determina el aspecto a partir de la/s regla/s aplicable/s. Si no hay ninguna, se usa un estilo "por defecto"

- El aspecto se "hereda" (podemos ver el HTML como un "árbol genealógico" en el que si una etiqueta está dentro de otra es "hija" de esta) . La herencia no se aplica a todas las propiedades, aunque sí a la mayoría.

```
body  {color:blue; font-size:25px;}
p     {font-size:15px;}
```

¿Dónde colocar las reglas de estilo?

- a) En un fichero aparte (por convenio extensión .css)
  - Hay que vincular en el HTML el CSS asociado

```
<head>
  <link href="estilo.css" rel="stylesheet" media="screen">
</head>
```

- b) Incluir el código CSS en la cabecera

```
<head>
    <style media="screen">
       h1 {color:red}
    </style>
</head>
```

- c) En el atributo style de una etiqueta
  - Recomendado únicamente si el estilo se usa una sola vez
  - Solo se especifican las propiedades

```
<h1 style="color:blue; font-size:25px"> Hola, amigos <h1>
```

La cascada CSS

- Los efectos CSS se aplican "en cascada", ya que podemos tener varias hojas de estilo distintas en la misma página
  - El color puede tomarse de un estilo vinculado en un .css mientras que el tamaño puede venir de un estilo en la cabecera

- Si los estilos son contradictorios, se emplean varios criterios
  - Quién ha definido el estilo: el usuario puede definir sus propios estilos y también el navegador tiene estilos por defecto.
  - **Especificidad**: ganan los selectores más específicos
  - **Orden de definición**: gana la última regla

Selectores: tipos (I)

- Combinaciones de etiquetas
  - Poner varias etiquetas en la misma regla, para ahorrar repeticiones. Se ponen separadas por comas

```
h1,  h2     {background:blue}
```

  - Que afecte a una etiqueta solo si está dentro de otra. Afecta a la de la derecha
    - A la que está inmediatamente dentro (separar por un espacio)
    - A cualquier "descendiente" (símbolo >)

```
p em    {color:red}    /* em inmediatamente dentro de p */
p > em {color:blue}    /* em dentro de p, inmediatamente o no*/
```

```
<p> <em> Yo soy rojo </em> </p> <em> pero yo no </em>
<p> <code> <em>Yo soy azul </em> </code></p>
```

Selectores: tipos (II)

- ¿Cómo cambiar el estilo solo de cierto/s <p> (o la etiqueta que sea) y no de todos los que aparecen en la página?
- **Clase** (para utilizar varias veces)
  - Se definen con un nombre precedido de un punto, para que el navegador sepa que no es una nueva etiqueta, sino una clase

```
.hortera      {color:red; background:yellow;}
```

  - En el HTML se usa el atributo *class* en la/s etiqueta/s que queramos

```
<p class="hortera">rojo sobre amarillo</p>
<h1 class="hortera">Y yo</h1> <p>pero yo no</p>
```

- **Identificador** (solo una vez)

```
/* margin:auto centra el elemento HTML */
#pie      {margin:auto; font-size:8px; color:gray;}
```

```
<div id="pie">Copyright DCCIA 2011</p>
<p id="pie">En teoría cada id solo se debe usar una sola vez, por lo que la regla CSS no debería aplicarse aquí</p>
```

Selectores: tipos (III)

- **Pseudoclases**: estados de etiquetas
  - a:visited – enlace ya visitado
  - :hover - elemento por debajo del ratón

```
a:hover      {background:yellow}
```

- **Pseudoelementos**: partes de elementos
  - :first-letter
  - :first-line

```
p:first-letter   {font-size:3em}
```

- **CSS3** introduce muchos más selectores

```
input[type="text"]  /* comprueba el valor de un atributo */
p:first-child       /* el primer párrafo dentro de otro elemento */
tr:nth-child(2n)    /* Las filas 2, 4, 6, ... */
. . .
```

CSS para etiquetas HTML5

- En principio no debería haber diferencia. Aunque el navegador no "entienda" HTML5, se limitará a darle a cada etiqueta el estilo asociado...

```
aside   { border: 1px solid, background: yellow}
```

- ... excepto Explorer 8, que no permite dar estilo a etiquetas desconocidas
  - Hay un "pequeño" truco consistente en crear dinámicamente una instancia de cada etiqueta con Javascript, meterlas en el documento y luego borrarlas (!) para que IE a partir de ahí permita dar estilo a dichas etiquetas.
  - *Script* que hace esto: http://remysharp.com/2009/01/07/html5-enabling-script

Tipos de propiedades

- **Fuentes**: tipos letra, tamaño, negrita, cursiva,...
- **Color**: de texto, de fondo, imagen de fondo,...
- **Texto**: justificación, espacio entre letras y/o palabras
- **Caja**: bordes a los 4 lados
- **Formato visual**: posición en la página, mostrar o no...
- **Tablas**: similares a las tablas HTML pero no "ensucian" la página
- **Listas, numeración automática**: permiten controlar el aspecto de las listas con viñetas y numeradas, y también numerar automáticamente párrafos o cualquier otra etiqueta

Unidades de medida

- **Palabras clave**: small, bolder
- **Unidades absolutas**: px (pixeles), cm, mm, in (pulgadas), pt (=1/72 in), pc (=12 pt)
- **Unidades relativas**: em (≈tamaño por defecto. Ej: 2em=2 veces el tamaño por defecto)
- **URLs**: url(http://www.dccia.ua.es/logo.gif) (también vale con comillas dobles o simples)
- **Colores**: yellow, #ff0000, rgb(255,0,0), rgb(100%,0%,0%)

Propiedades de fuentes


```
.p          {
              font-size:25px;                          /* tamaño */
              font-family:"Arial narrow", sans-serif;  /* fuente */
              font-weight: 400                         /* negrita */
            }
```

Color


```
P       {
          color:rgb(0,0,255);                    /* color texto */
          background-image: url("mosca.gif");    /* imagen fondo */
          background-repeat: repeat              /* fondo en mosaico */
        }
```

El "Modelo de caja"

- Cada elemento HTML se supone "encerrado" en una caja rectangular
  - Normalmente las cajas ocupan todo el ancho de la ventana y van una debajo (o justo al lado) de la otra, salvo que especifiquemos posiciones
- Hay etiquetas que fuerzan retorno de carro antes y después (elementos "**de bloque**"). Otras no (elementos "en línea" o ***inline***). Se puede cambiar el comportamiento por defecto con la propiedad `display` (valores posibles: block, inline, hidden)
  - "hidden" indica que el elemento no se mostrará. Solo tiene sentido si se va a cambiar dinámicamente con Javascript

```
/* los enlaces dentro de la sección con id="navegacion"
aparecerán uno debajo del otro */
#navegacion > a { display:block;}
```

Partes del “Modelo de Caja”

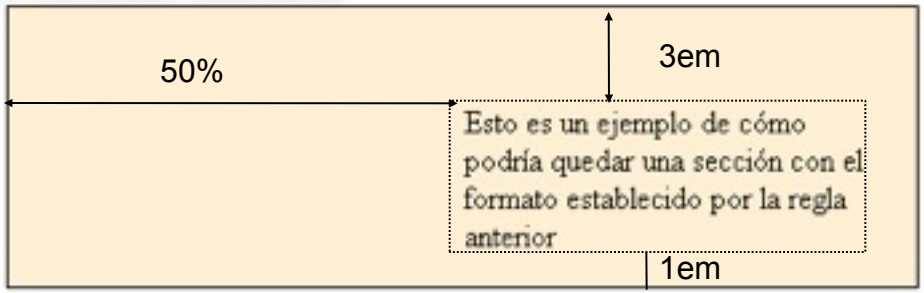
CSS

```
#aviso  {
        padding-left: 50%;
        padding-right: 20px;
        padding-top:3em;
        padding-bottom: 1em;
        background-color: #FFEFD5;
        border: 1px solid;
}
```

HTML

```
<div id=”aviso”> Esto es un ejemplo de cómo podría quedar una sección con el formato establecido por la regla anterior
</div>
```

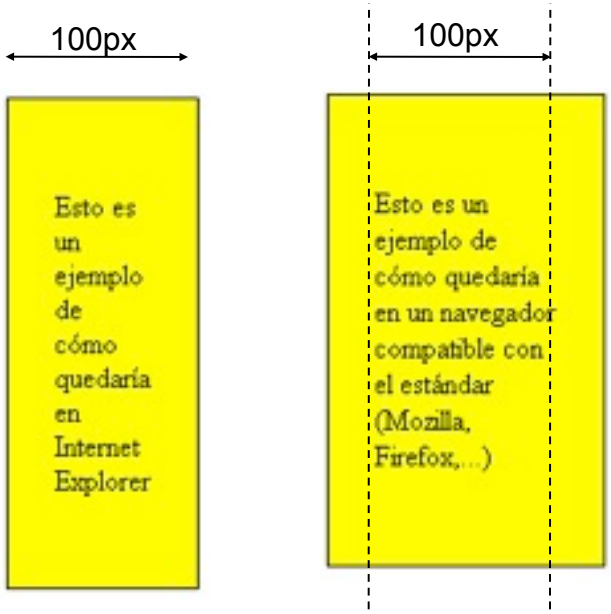
Resultado



Dimensiones de caja

- Especificar ancho y alto: propiedades **width** y **height**
- El IE (hasta el 7) incumple el estándar si no se pone el DOCTYPE "strict" (recordad el "DOCTYPE switching")
  - Explorer: width = contenido + padding + border
  - Estándar: width = solo contenido

```
#aviso2 {
      width:100px;
      background-color : yellow;
      padding: 50px 25px 50px 25px;
      border: 1px solid;
    margin: auto; /* para que quede centrado */
}
```



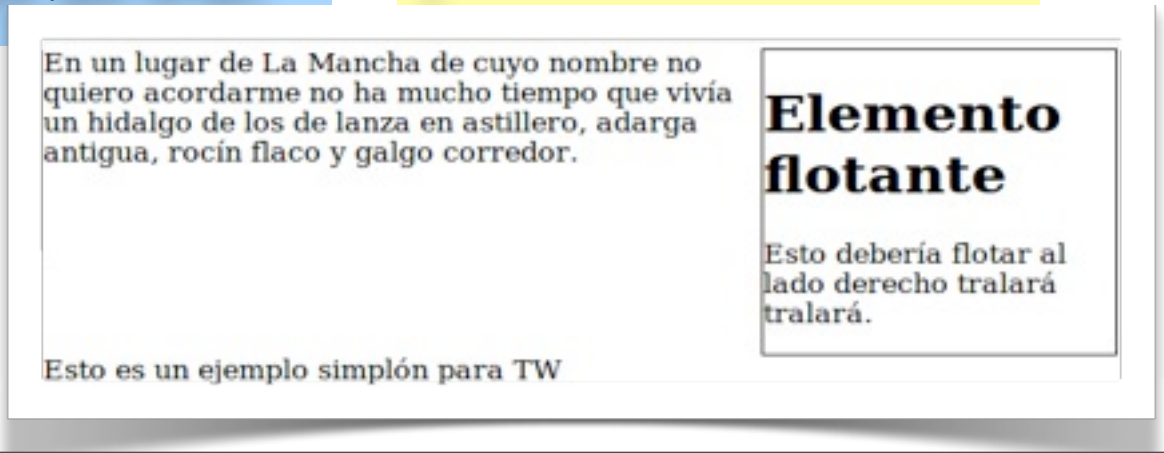
Elementos flotantes

- Propiedad **float** (valores: left, right, none-por defecto)
  - El elemento se alinea al lado especificado y los siguientes elementos lo "rodean" dejando espacio para él
  - Con la propiedad `clear` podemos hacer que un elemento no "rodee" a otro flotante por la izq, der o ambos lados

```
<div id="flotante">
   <h1>Elemento flotante</h1>
   <p>Esto debería flotar al lado derecho tralará tralará.</p>
</div>
<p>En un lugar de La Mancha de cuyo  nombre no quiero acordarme...
…
</p>
<div id="pie">Esto es un ejemplo simplón  para TW</div>
```

```
#flotante {
        float:right;
        width: 200px;
        border: 1px solid;
}

#pie {
        clear: both;
}
```



Formato visual. Posicionamiento

- Especificar tipo de posicionamiento: atributo **`position`**
  - **absolute**: referente al elemento contenedor del posicionado. Se "saca" del flujo normal de la página
  - **fixed**: referente a la ventana (no va en Explorer 6). Cuando se hace *scroll*, se queda "quieto"
  - **relative**: referente a la posición que ocuparía normalmente. Deja un hueco en dicha posición.
- Especificar posición
  - **Esquina superior izquierda (origen de coords): left**: x, **top**: y
  - **Esquina inferior derecha: right**: x, **bottom**: y (normalmente se especifica solo una de las dos esquinas, la otra vendría implícita por width y height del elemento)
  - **z-index**: z (cuanto mayor, más cerca). Si dos elementos se solapan se ve por encima el más cercano

CSS

```
...
#abs {
        position: absolute;
        left: 100px;
        top: 100px;
        z-index: 1;
}

#rel {
        position: relative;
        color: red;
        left: 30px;
        top: 30px;
        z-index: 2;
}
...
```

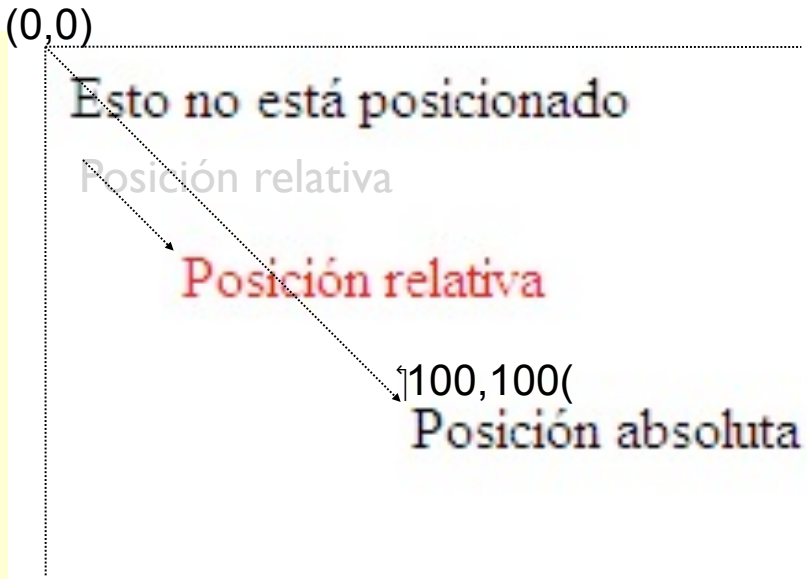
HTML

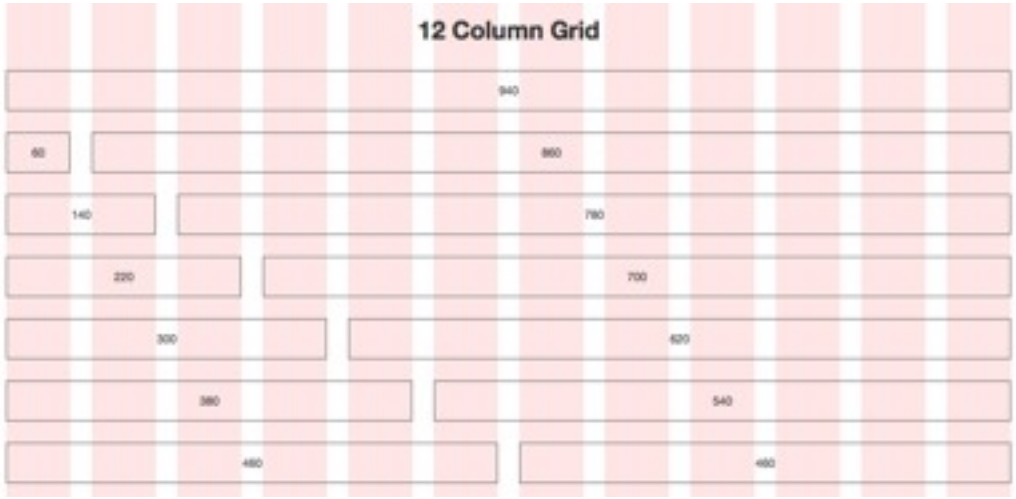
```
<body>

  <div id="abs"> Posición
     absoluta</div>

  <div>Esto no está
        posicionado</div>

  <div id="rel">Posición
        relativa </div>
  ...
```



Frameworks CSS

- "Plantillas" configurables que solucionan algunos problemas comunes
  - Proporcionan una "rejilla" a partir de la que construir la estructura de la página
  - Incompatibilidades entre navegadores
- Ejemplos:
  - 960.gs
  - blueprint CSS
  - YUI grids