Tecnologías Web para la presentación CSS

Aplicaciones Web / Sistemas Web

Juan Pavón Mestras
Dep. Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense Madrid

Material bajo licencia Creative Commons

CSS

- Hojas de estilo en cascada (*Cascading Style Sheets*)
  - Ficheros de texto con la extensión **.css**
  - Definen la apariencia de las páginas Web
  - Facilitan la gestión de sitios web grandes y sofisticados
    - Las hojas CSS se crean una vez y se pueden compartir entre varios desarrolladores WEB
      - Por ejemplo, estilo corporativo
    - Para realizar un cambio de estilo en todo el sitio solo hay que hacerlo en un lugar: la hoja CSS correspondiente

- Versiones
  - CSS nivel 1 (1996, revisado en 2008) – *http://www.w3.org/TR/CSS1*
  - CSS nivel 2 (1998), actual: 2.1 (2011) – *http://www.w3.org/TR/CSS21*
    - CSS 2.1 Soportado por todos los navegadores habituales
  - CSS nivel 3 (en desarrollo) – *http://www.w3.org/Style/CSS/current-work*
    - Varias características ya definidas (selectores, pseudoclases y muchas propiedades) las van soportando todos los navegadores habituales (MS Internet Explorer es el más retrasado, solo desde la versión 10.0)

CSS y XHTML

- Con HTML tanto el contenido como la presentación están mezclados

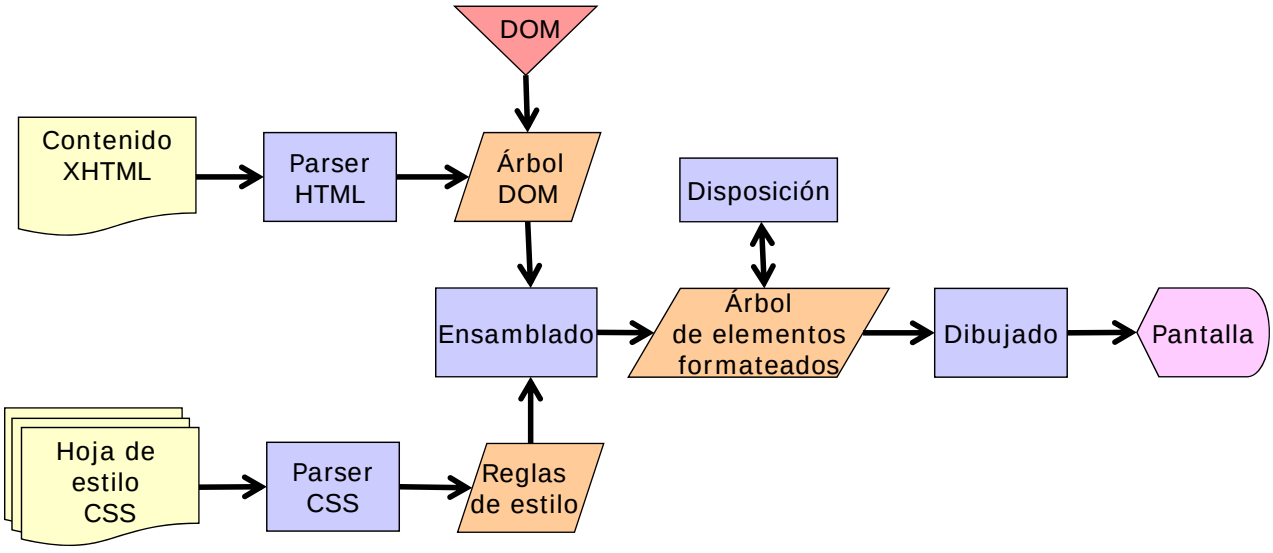
- Con XHTML + CSS
  - XHTML especifica contenidos y estructura

```
<body>
  <h1>Primera sección</h1>
  <p>Un párrafo de texto.</p>
</body>
```

  - CCS especifica presentación y formato

```
<style type="text/css">
  h1 { color: red;  font-family: Verdana; font-size: large;  }
  p  { color: gray; font-family: Verdana; font-size: medium; }
</style>
```

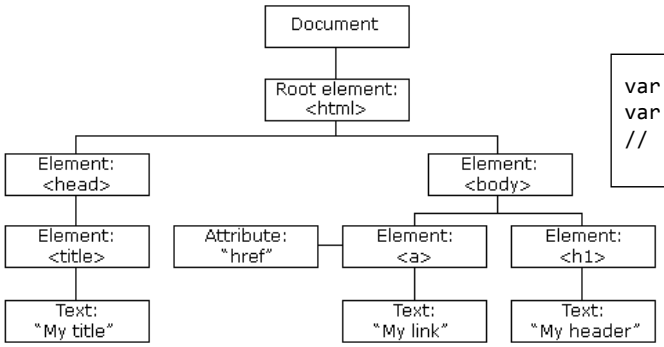

Proceso de XHTML y CSS en un navegador



Adaptado de:
Tali Garsiel. *How browsers work. Behind the scenes of modern web browsers*
http://taligarsiel.com/Projects/howbrowserswork1.htm

DOM

- Definido por W3C
- Modelo de Objetos del Documento (*Document Object Model*)
  - DOM define objetos y propiedades de los elementos HTML y XML, y los métodos para acceder a ellos
    - Representación de documentos HTML y XML
    - API para consultar y manipular los documentos (contenido, estructura, estilo)
- Los objetos de un documento se organizan en una jerarquía (árbol): jerarquía DOM




```
var elemento=document.getElementById("intro");
var cabecera=document.getElemnentByTag("title");
// ...
```


Integración de CSS en HTML/XHTML

- La definición de los estilos para un elemento se puede realizar:
  - Dentro del propio elemento (internas al elemento)
    - NO aconsejable

```
<p style="color: black; font-family: Verdana;">Bla bla bla</p>
```

  - Con una **hoja de estilo interna** (en el mismo documento)
    - Con la etiqueta <style> dentro de la sección <head>

```
<head>
...
<style type="text/css">
  p { color: black; font-family: Verdana; }
</style>
</head>
```

  - Con una **hoja de estilo externa**
    - La mejor opción (ver página siguiente)
  - Comportamiento por defecto del navegador
- Todas las definiciones se combinan en *cascada* para producir una única hoja de estilo
  - Las definiciones mas locales tienen prioridad
    - Navegador << Externo << Interno << Elemento

Integración de CSS en XHTML

- Ficheros CSS externos
  - El fichero externo es un fichero de texto donde se definen los estilos
    - Ejemplo: Dado el fichero `estilos.css`:

```
p { color: black; font-family: Verdana; }
```

  - Se puede incluir de dos maneras, dentro de la sección <head>:
    - Con la etiqueta <link>

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

    - Con la etiqueta <style> y una regla de tipo *@import*

```
<style type="text/css" media="screen">
  @import '/css/estilos.css';
</style>
```

    - Estas reglas deben ir antes de definir reglas con propiedades

- En general conviene que los estilos se definan en ficheros .css externos para separar mejor contenido de presentación
  - Se puede considerar hacerlo en el documento XHTML cuando es un proyecto simple o si son pequeñas modificaciones de uno más general

Integración de CSS en XHTML

- Inclusión de ficheros de estilo con la etiqueta <link>
  - Atributos:
    - **rel="stylesheet"**: relación entre los archivos CSS y el XHTML
    - **type="text/css"**: el tipo de recurso para CSS siempre es text/css
    - **href="url"**: URL del archivo CSS que contiene los estilos
      - La URL puede ser relativa o absoluta y apuntar a un recurso interno o externo al sitio web
    - **media**: indica el medio en el que se van a aplicar los estilos del archivo CSS
      - Esto permite definir distintos estilos según el dispositivo. Por ejemplo:
        - all — Todos los medios definidos
        - **screen** — Pantallas de ordenador
        - **print** — Impresoras y modo "Vista Previa para Imprimir"
        - **handheld** — Dispositivos de mano: móviles, PDA, etc.
        - tv — Televisores y dispositivos con resolución baja
      - Para personas con discapacidad:
        - braille — Dispositivos táctiles que emplean el sistema braille
        - embosed — Impresoras braille
        - speech — Sintetizadores para navegadores de voz

Reglas @media

- Tipo especial de reglas CSS para indicar el medio o medios en los que se aplicarán los estilos incluidos en la regla
- El medio en el que se aplican los estilos, se indica después de @media
  - Si los estilos se aplican a varios medios, se incluyen los nombres de todos los medios separados por comas

```
<style>
@media print {
  body { font-size: 10pt }
}
@media screen {
  body { font-size: 44px }
}
@media screen, print {
  body { line-height: 1.2 }
}
</style>
```

```
<link rel="stylesheet"
type="text/css"
href="basico.css"
media="screen, print" />

@import url("estilos_seccion.css")
screen;

@media print {
  /* Estilos para impresora */
}
```


Ejercicio

- Probar a definir dos estilos para *screen* y *print*, y observar si es distinto el tratamiento en el navegador
  - Definir distinto tipo de tamaño de letra (44px para screen y 10px para print) con las reglas que aparecen en el primer ejemplo de la página anterior
  - Abrir la página en el navegador y observar el tamaño del texto de la página
  - Utilizar la opción *Print preview* para observar cómo se vería si se fuera a imprimir la página

Sintaxis de las hojas de estilo

- Una hoja de estilo define un conjunto de reglas
- Cada regla de estilo consta de
  - **Selector**: elemento al que se aplica el estilo
    - Pueden ser varios, separados por **,**
  - **Lista de declaraciones**
    - Cada declaración es un par **propiedad:valor**
    - Separadas por **;**
- Ejemplos:

```
h1 { color: red;  font-family: Verdana; font-size: large;  }
p { color: blue; font-family: Verdana; text-align:center; }
```

selector propiedad valor

- Comentarios
  - Entre /* y */

```
/* Esto es un comentario muy original */
```


Selectores

- **Selector universal**
  - Todos los elementos de la página: ***

```
* { margin: 0; padding: 0; }
```

- **Etiqueta**
  - Indica el estilo aplicable a una etiqueta (se especifica sin < >)

```
p {text-align: center; color: red}
```

  - Se pueden agrupar varias etiquetas para aplicarles el mismo estilo
    - Separadas por coma **,**
    - Se pueden definir también otras reglas más particulares

```
h1, h2, h3, h4, h5, h6 {font-family: Verdana; }
h1 { font-size: 2em; }
h2 { font-size: 1.5em; }
```

Selectores

- **Clase**
  - A una etiqueta se le puede asociar una clase (o varias)

```
<h2 class="cabecera2">
<h2 class="clase1 clase2">
```

  - Esto permite aplicar estilos de la(s) clase(s) a la etiqueta
    - El nombre de la etiqueta se separa del nombre de la clase con **.** (*importante*: sin espacios entre los nombres y el punto)

```
h2.cabecera2 { text-align: center; }
```

  - Se puede definir una clase sin especificar tipo de etiqueta
    - Esto equivale a `*.clase`

```
.cabeceracentrada { text-align: center; }
<h1 class="cabeceracentrada">Título centrado</h2>
...
<h2 class="cabeceracentrada">Subtítulo centrado</h2>
```


Selectores

- **Selectores descendentes**
  - Separados por espacio: permite seleccionar los elementos especificados por un selector dentro del ámbito de otro
    - **`p .destacado { ... }`**

    → Se aplica a todos los elementos con atributo class="destacado" que estén dentro de un párrafo <p>

- ATENCIÓN: Los espacios y la puntuación son importantes:
  - **`p.destacado`** { ... }

  → Todos los párrafos <p> que estén declarados con atributo class="destacado"
  - **`p, .destacado`** { ... }

  → Todos los párrafos <p> y todos los elementos con atributo class="destacado"

Selectores

- Ejemplos de selectores descendentes:
  - **`div.principal span.especial { ... }`**

→ Todos los elementos span con atributo class="especial" que estén dentro de un elemento div con atributo class="principal"

  - **`p span em { ... }`**

→ Todos los elementos dentro de un dentro de un <p>

  - **`p * em { ... }`**

→ Todos los elementos dentro de algún elemento dentro de un <p>

    - Se aplica a
      `<p><span><a href="#">Enlace</a></span></p>`
    - No se aplica a
      `<p><a href="#">Enlace</a></p>`
  - **`p em { ... }`**
    - Se aplicaría a los dos casos anteriores

Selectores

- **ID**
  - Permite aplicar un estilo a un único elemento de una página
    - El elemento se identifica con un atributo id, que es único

```
<p id="destacado">Segundo párrafo</p>
```

    - El estilo para el ID se especifica precedido con el carácter **#**

```
#destacado { color: red; }
```

  - Como recomendación
    - Usar selector de clase, que se aplica a varios elementos, para dar una apariencia homogénea al documento
    - Usar selector de ID con mesura
- Rizando el rizo
  - **`p#destacado`** `{ ... }` → Se aplica a todos los párrafos <p> que tengan id="destacado"
    - Puede parecer absurdo porque solo habrá un elemento (párrafo u otro tipo) en la página con ese id, independientemente del tipo
    - Podría ser útil si se aplicara el estilo a varias páginas
  - **`p #destacado`** `{ ... }` → Se aplica a un elemento con id="destacado" que esté dentro de un párrafo <p>
  - **`p, #destacado`** `{ ... }` → Se aplica a todos los párrafos <p> y al elemento de cualquier tipo que tenga id="destacado"

Selectores

- **Pseudo clases**
  - Clases "virtuales" que no se insertan de manera explícita en las páginas (están predefinidas)
  - Se pueden definir sus propiedades como si fueran clases (utilizando **:** en vez de .)

```
a:link { color: blue }
```

- Pseudo clases de enlaces
  - **a:link** – Enlace no visitado o activado por el usuario
  - **a:hover** – Cuando se coloca el ratón sobre el enlace
  - **a:active** – Cuando se pincha sobre el enlace
  - **a:visited** – Enlace ya visitado por el usuario

```
a { background-color: white; color: blue;
  border: white 3px solid; outline: white 3px solid;
}
a:visited { background-color: yellow;  color: red; }
a:hover { background-color: black; color: white; }
```


Selectores

- Pseudo clases dinámicas
  - **:hover** – Cuando se coloca el ratón sobre el elemento
  - **:active** – Cuando se pincha sobre el elemento y se mantiene el botón del ratón pulsado
  - **:focus** – Cuando el elemento tiene foco
    - Pueden tener foco los elementos que reaccionan a entrada por teclado (los elementos de los formularios o los enlaces)
    - Ejemplo: al seleccionar un campo de un formulario, se resalta con un borde rojo:

```
input:focus { border: red 2px dotted; padding: 2px; }
```

- Otras pseudo clases
  - **:first-child** – Primer elemento de un tipo contenido dentro de otro

```
div p:first-child { color: red; }
```

  - **:lang()** – Aspecto de los elementos de un idioma determinado.

```
q:lang(es) {  quotes: "«" "»"; }
q:lang(en) {  quotes: '"' '"'; }
→ <p>El profesor dijo que <q lang="es">Muy bien</q>.</p>
→ <p>The teacher said <q lang="en">Very well</q>.</p>
```

Selectores

- **Pseudo elementos**
  - Similar a las pseudo clases

- Pseudo elementos típicos:
  - **:first-letter** – Primera letra en un elemento de bloque

```
h1:first-letter { font-size: 400%; }
```

  - **:first-line** – Primera línea en un elemento de bloque

```
p:first-line { text-transform: uppercase; }
```

  - **:before** y **:after** – Para añadir contenido antes o después de un elemento

```
p.cuidado:before { content: "Aviso: ";
  font-weight: bold;
  text-decoration: underline;
}
```

→ `<p class="cuidado">Esto es un mensaje de atención.</p>`

Aviso: Esto es un mensaje de atención.

Selectores

- **Selector de hijos**
  - Para seleccionar un elemento que es hijo directo de otro elemento
  - Se indica mediante el signo **>**

```
p > span { color: blue; }
```

    - Se aplica el selector → `<p><span>Texto1</span></p>`
    - No se aplica → `<p><a href="#"><span>Texto2</span></a></p>`
    - El selector descendente (`p span {…}`) se aplicaría en los dos casos

- **Selector adyacente**
  - Para seleccionar elementos que se encuentran seguidos
  - Se indica mediante el signo **+**

```
p + p { text-indent: 1.5em; }
```

    - Selecciona todos los párrafos de una página que estén precedidos por otro, esto es, todos menos el primero

Selectores

- **Selector de atributos**
  - Para seleccionar elementos en función de sus atributos
  - Cuatro tipos:
    - **[nombre_atributo]** – elementos que tienen definido el atributo llamado nombre_atributo, independientemente de su valor

      `[title] { color: red; }` → todos los elementos con el atributo *title*
    - **[nombre_atributo=valor]** – elementos que tienen un atributo llamado nombre_atributo con el valor especificado

      `[title="web"] { color: red; }` → todos los elementos con *title="web"*
    - **[nombre_atributo~=valor]** – elementos que tienen un atributo llamado nombre_atributo y al menos uno de los valores del atributo es el valor especificado

      `[title~="web"] { color: red; }` → todos los elementos que tengan *"web"* dentro del valor de *title*
    - **[nombre_atributo|=valor]** – elementos que tienen un atributo llamado nombre_atributo y cuyo valor es una serie de palabras separadas con guiones, pero que comienza con valor

      `*[lang=en] { ... }` → todos los elementos en inglés

`*[lang|="es"] { ... }` → todos los elementos en español: "es", "es-ES", "es-AR",...

Herencia

- Cuando se establece el valor de una propiedad CSS en un elemento, sus elementos descendientes heredan de forma automática el valor de esa propiedad
  - Por ejemplo, si se establece la regla

```
body { color: red; }
```

  - Todo lo que está dentro de *body* tendrá el color red mientras no se declare otra regla más específica, que será la que prevalezca

```
p { color: blue; }
```

    - Por ejemplo, para el código HTML

```
<h1>Sección 1</h1>
<p>Un texto con una parte<em>enfatizada</em></p>
```

    - "Sección 1" aparecerá en rojo, el párrafo en azul, y el texto "enfatizada" en azul a menos que haya alguna declaración de estilo específica para que aparezca en verde:

```
em { color: green; }
```

Cascada de estilos

- ¿Qué ocurre cuando hay dos declaraciones contradictorias?

```
p { color: red; }
p { color: blue; }
```

- En general, la regla básica es:
  - Cuanto más específico sea un selector, más importancia tiene su regla asociada
  - A igual *especificidad*, se considera la última regla indicada
    - De ficheros externos a estilos dentro del propio elemento
    - En una lista de reglas en el mismo ámbito, se elige la última

Cascada de estilos

- El proceso para resolver la regla que se aplica es el siguiente:
  1. Buscar todas las declaraciones aplicables a un elemento
     - Solo se tienen en cuenta las que correspondan al medio CSS seleccionado (screen, print, etc.)
  2. Si no existen reglas, se usa el valor heredado
     - Si no hubiera valor heredado, el navegador usa el valor por defecto
  3. Si existen reglas, se aplica la de mayor peso de acuerdo a los siguientes criterios:
     - Se asigna un peso según su origen
       - De mayor a menor: autor, usuario, navegador
     - Si un atributo tiene la palabra clave **!important** se le da mayor prioridad
       - `selector { atributo:valor ! important ; atributo: valor ; }`
     - Se da mayor prioridad a los selectores más específicos
       - etiqueta << clase << ID
     - Si después de aplicar las normas anteriores existen dos o más reglas con la misma prioridad, se aplica la que se indicó en último lugar

Ejercicios

- ¿A qué elementos se aplican las siguientes reglas?
  - `a:link img { border: solid blue }`
  - `a.external:visited { color: blue }`
  - `h2 + p:first-letter { font-size: 200%;}`
  - `p.inicial:first-letter { color: red }`
  - ```
    h1 { color: blue }
    em { color: red }
    ```
  - ```
    h1 { color: blue }
    h1 em { color: red }
    ```

- Realizar el ejercicio de
  http://librosweb.es/css/capitulo_15/ejercicio_2.html

Caracterización de los estilos

- Unidades de medida y colores

- Posicionamiento y visualización
  - Modelo de cajas
  - Modelo de posicionamiento
  - Propiedades de visualización

- Estilos de los elementos
  - Texto
  - Enlaces → con las pseudoclases a:link, a:hover, a:active, a:visited
  - Imágenes
  - Listas
  - Tablas
  - Formularios

Unidades de medida

- Para definir la altura, anchura, márgenes, tamaño de letra, ...
- Los valores se expresan como números enteros o con decimales
- Unidades absolutas
  - cm, centímetros
  - mm, milímetros
  - in, pulgadas (*inches*). 1in==2.54cm
  - **pt**, puntos. 1pt==0.35mm ← la más utilizada
  - pc, picas. 1pc==12 pt
- Unidades relativas
  - **em**, relativa respecto al tamaño de letra del elemento
    - Si el font-size es de 12pt, 1em==12pt
  - **ex**, relativa respecto a la altura de la letra **x** minúscula
  - **px**, (píxel) relativa respecto a la resolución de la pantalla del dispositivo (respecto al tamaño del canvas)
- Porcentajes
  - Valor numérico seguido del símbolo **%**

```
p { line-height: 150% } /* 150% del valor de 'font-size' */
```


Colores

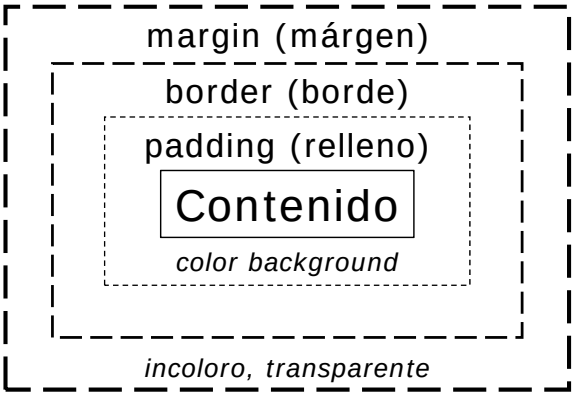
- Los colores se pueden especificar de varias maneras:
  - Palabras clave: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, orange, purple, red, silver, teal, white, yellow.
  - RGB hexadecimal: #ff0000 ← la manera más habitual
  - RGB decimal: rgb(255,0,0)
  - RGB porcentual: rgb(100%, 0%, 0%)

Imagen del estándar.
http://www.w3.org/TR/CSS21/syndata.html#value-def-color

El modelo de cajas

- Para cada etiqueta HTML se crea una caja rectangular que encierra los contenidos de ese elemento
- El tamaño de cada área (**margin, border, padding**) se define con propiedades relativas a las cuatro direcciones:
  - **top, bottom, left, right**

```
margin: 10px /* Propiedad resumida */
margin-top: 20px;
margin-right: 20px;
```



El modelo de cajas

- El tamaño del **contenido** se define con:
  - **width**
  - **height**
    - <medida>
    - <porcentaje>
    - auto: lo ajusta el navegador
    - inherit: valor del elemento padre
- Propiedades del **borde**
  - Estilo del borde: **border-style**
    - none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset | inherit
  - Anchura: **border-width**
  - Color: **border-color**
  - Lados individuales: **border-top-style, border-right-style, border-bottom-style, border-left-style**
  - Resumen: **border**
    - (<medida_borde> || <color_borde> || <estilo_borde>) | inherit

Ejercicio

- ¿Cuál será la anchura del siguiente bloque?

```
div {
  width: 400px;
  padding-left:  60px;
  padding-right: 60px;
  margin-left:   40px;
  margin-right:  40px;
  border: 10px solid black;
}
```


El modelo de cajas

- Fondo
  - Un color: **background-color**
    - <color> | transparent | inherit
  - Una imagen: **background-image**
    - <url> | none | inherit
    - La imagen se puede repetir, esto se controla con **background-repeat**
      - repeat | repeat-x | repeat-y | no-repeat | inherit
    - Qué ocurre con la imagen de fondo cuando se hace scrolling en la página: **background-attachment**
      - scroll | fixed | inherit
  - Todas las propiedades: **background**
    - (<background-color> || <background-image> || <background-repeat> || <background-attachment> || <background-position>) | inherit

El modelo de posicionamiento

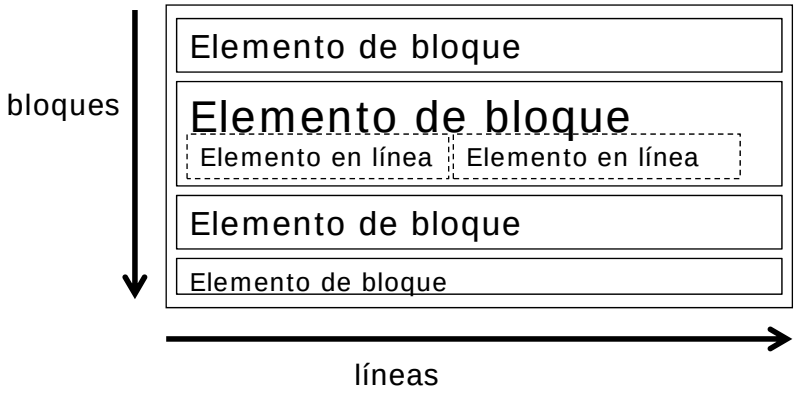
- Tipos de elementos
  - De bloque
    - Siempre empiezan en una nueva línea y ocupan todo el espacio disponible hasta el final de la línea
    - address, blockquote, center, dir, div, dl, fieldset, form, h1, h2, h3, h4, h5, h6, hr, menu, noframes, noscript, ol, p, pre, table, ul
  - En línea
    - Sólo ocupan el espacio necesario para mostrar sus contenidos
    - Se pueden insertar en elementos de bloque y de línea
    - a, abbr, acronym, b, basefont, bdo, big, br, cite, code, dfn, em, font, i, img, input, kbd, label, q, s, samp, select, small, span, strike, strong, sub, sup, textarea, tt, u, var
- Para cada elemento se genera una caja
  - La caja de un elemento contiene las cajas de los elementos que contiene
  - El elemento raíz del documento (<body>) genera una caja que sirve como bloque contenedor inicial para el resto de elementos

Tipos de posicionamiento

- Una caja se puede colocar según uno de los siguientes modos:
  - **position: static**
    - Normal o estático: es el modo de posicionamiento habitual
  - **position: relative**
    - Como el posicionamiento normal pero la caja se desplaza según los atributos left, right, top, bottom
  - **position: absolute**
    - La caja se coloca de manera absoluta con respecto a la primera caja contenedora no estática según los valores left, right, top, bottom
      - Si no estuviera contenido en ninguna caja no estática tomaría <body>
    - El atributo z-index permite controlar como se apilan las cajas
      - Las cajas que queden debajo se ocultarán
  - **position: fixed**
    - Tipo de posicionamiento absoluto, pero la caja mantiene su posición y no se mueve cuando se usan las barras de desplazamiento
  - **float**
    - Posicionamiento flotante: Desplaza las cajas todo lo posible hacia la izquierda o hacia la derecha de la línea en la que se encuentran.

Posicionamiento normal (estático)

- Las cajas se colocan seguidas verticalmente de arriba abajo, en el orden en el que aparecen los elementos correspondientes en el documento
  - Sólo se tiene en cuenta si el elemento es de bloque o en línea, sus propiedades width y height y su contenido



Posicionamiento relativo

- La caja se desplaza respecto de su posición normal
- El desplazamiento de la caja se controla con las propiedades top, right, bottom y left
  - Por defecto top, right, botton y left tienen el valor auto
  - El valor positivo de alguno de estos valores implica un desplazamiento
    - top hacia abajo desde el borde superior, bottom hacia arriba
    - left hacia la derecha desde el borde izquierdo, rigth hacia la izquierda
- Las cajas desplazadas de forma relativa no modifican su tamaño
  - Las propiedades left y right siempre cumplen que left = -right
    - Si left es auto, su valor será –right
    - Si left y right tienen valores no compatibles se toma el que determine la propiedad direction (ltr o rtl)
- El resto de las cajas no se ven afectadas

Figura de:
http://librosweb.es/css/capitulo_5/posicionamiento_relativo.html

Posicionamiento absoluto

- La posición de la caja se determina con las propiedades top, right, bottom y left respecto al primer elemento contenedor estático (si no hubiera, entonces se usa el <body>)
- El resto de las cajas sí pueden verse afectadas
  - Pueden ocupar la posición que ha dejado la anterior
  - Se pueden producir solapamientos

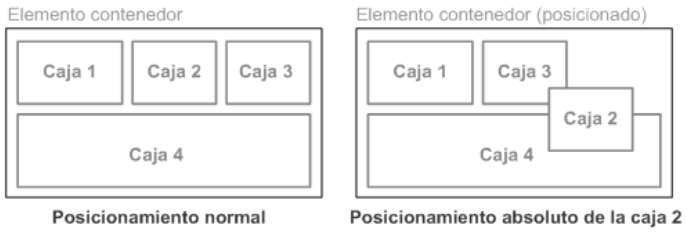


Figura de:
http://librosweb.es/css/capitulo_5/posicionamiento_absoluto.html

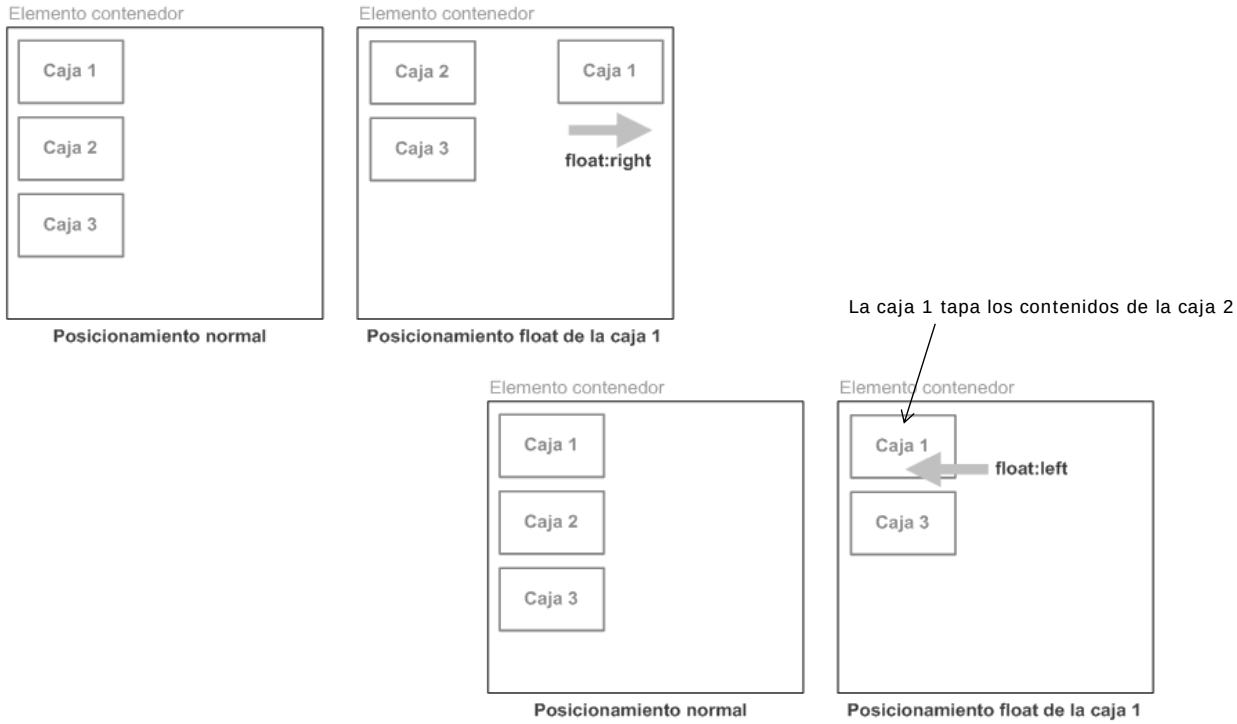
Posicionamiento fijo

- Igual que el anterior pero las cajas correspondientes no se mueven
- Puede servir por ejemplo en el medio print para imprimir páginas con encabezado y pie de página

Posicionamiento flotante

- Una caja flotante se declara con **float:right** o **float:left**
- Una caja flotante se desplaza horizontalmente
  - Hasta la zona más a la izquierda o más a la derecha posible
  - Si hay otras cajas flotantes, se van colocando seguidas
  - Si no existe sitio en la línea actual, la caja flotante baja a la línea inferior hasta que encuentra el sitio necesario para mostrarse lo más a la izquierda o lo más a la derecha posible en esa nueva línea
- La caja deja de pertenecer al flujo normal de la página, lo que significa que el resto de cajas ocupan el lugar que ha dejado
  - Para que un elemento no se vea afectado por lo que hagan las cajas flotantes a su alrededor puede definir la propiedad **clear** que fuerza a que el elemento se muestre debajo de cualquier caja flotante
  - Atributos de clear: none | left | right | both | inherit
    - Si **clear: left**, el elemento se desplaza de forma descendente hasta que pueda colocarse en una línea en la que no haya ninguna caja flotante en el lado izquierdo
    - Si **clear: both**, el elemento se colocará debajo de cualquier caja flotante

Posicionamiento flotante

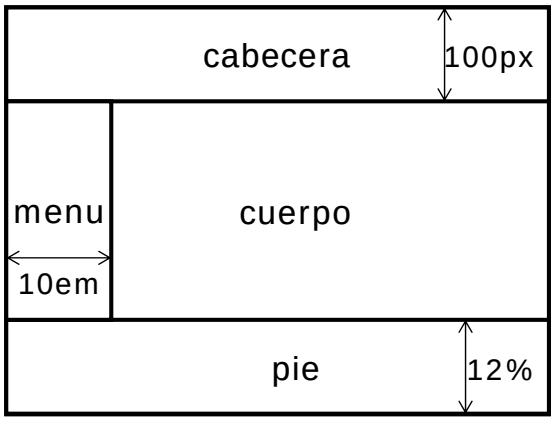


Figuras de:
http://librosweb.es/css/capitulo_5/posicionamiento_absoluto.html

Ejercicios de posicionamiento

- Crear los estilos para la siguiente configuración de página

```
<body>
<div id="cabecera">...</div>
<div id="menu">...</div>
<div id="cuerpo">...</div>
<div id="pie">...</div>
</body>
```



Propiedades de visualización

- **display**
  - Cambia el tratamiento de un elemento
  - Atributos
    - block: hace que se trate el elemento como un bloque
    - inline: hace que se trate el elemento como en línea
    - none: el elemento desaparece de la página
      - Y los demás elementos pueden ocupar su lugar
      - Se ignoran las propiedades float y position ya que la caja no se muestra

- **visibility**
  - Permite ocultar completamente un elemento que sigue ocupando su sitio
  - Atributos: visible | hidden | collapse | inherit
    - collapse es similar a hidden pero para filas y columnas de tablas

Propiedades de visualización

- **overflow**
  - Controla la forma en la que se visualizan los contenidos que sobresalen de sus elementos:
    - visible: el contenido no se corta y se muestra sobresaliendo la zona reservada para visualizar el elemento
      - Comportamiento por defecto
    - hidden: el contenido sobrante se oculta y sólo se visualiza la parte que cabe dentro de la zona reservada para el elemento
    - scroll: solamente se visualiza el contenido que cabe dentro de la zona reservada para el elemento, pero también se muestran barras de scroll que permiten visualizar el resto del contenido
    - auto: el comportamiento depende del navegador
      - Normalmente es el mismo que la propiedad scroll

- **z-index**
  - Indica las cajas que se muestran delante o detrás de otras cuando se producen solapamientos
    - Se especifica con un número: el mayor más arriba

Estructura de la página

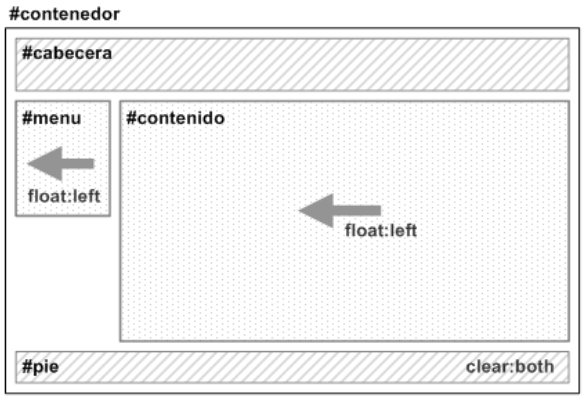
- Las etiquetas y en principio no tienen ningún significado para la presentación
  - Solo agrupan elementos
    - a nivel de bloque
    - a nivel de línea
- Combinados con selectores de clase permiten estructurar la aplicación de estilos en una página

- Para estructurar una página
  - Definir bloques con y su composición
    - A cada bloque asignarle una clase o un id
    - Cada bloque puede constar de otros bloques
    - A nivel de línea se pueden definir cajas con
  - Asociar propiedades de posicionamiento y visualización a cada una de las cajas definidas con clase o id

Estructura de la página

- Diseño a 2 columnas con cabecera y pie de página
  - De: *http://librosweb.es/css/capitulo_12/estructura_o_layout.html*
  - Uso de float y clear

```
#contenedor {
  width: 700px;
}
#cabecera {
}
#menu {
  float: left;
  width: 150px;
}
#contenido {
  float: left;
  width: 550px;
}
#pie {
  clear: both;
}
```

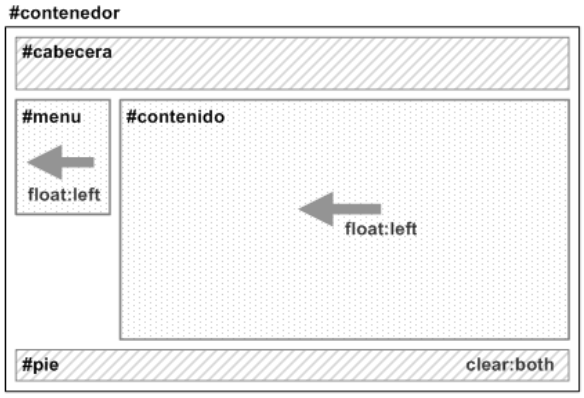



```
<body>
<div id="contenedor">
  <div id="cabecera">
  </div>

  <div id="menu">
  </div>

  <div
id="contenido">
  </div>

  <div id="pie">
  </div>
</div>
</body>
```


Estructura de la página

- Diseño a 2 columnas con cabecera y pie de página
  - Con anchura variable

```
#contenedor {
}
#cabecera {
}
#menu {
  float: left;
  width: 15%;
}
#contenido {
  float: left;
  width: 85%;
}
#pie {
  clear: both;
}
```




```
<body>
<div id="contenedor">
  <div id="cabecera">
  </div>

  <div id="menu">
  </div>

  <div
id="contenido">
  </div>

  <div id="pie">
  </div>
</div>
</body>
```

Trucos de posicionamiento

- Centrar una página horizontalmente con ancho determinado
  - Agrupar todos los contenidos de la página en un elemento
  - Asignarle a ese unos márgenes laterales automáticos con la propiedad margin de CSS

```
#contenedor {
  width: 400px;
  margin: 0 auto;
}
```

```
<body>
  <div id="contenedor">
    <h1>Sección 1</h1>
    <p> bla bla bla </p>
    ...
  </div>
</body>
```

Idea de *http:///librosweb.es/css/capitulo_12/centrar_una_pagina_horizontalmente.html*

Adaptación de la página al tamaño de pantalla

- Propiedades para limitar la anchura y altura mínima y máxima de cualquier elemento de la página
  - **min-width**: anchura mínima de un elemento
    - Por defecto: 0
  - **max-width**: anchura máxima de un elemento
    - Por defecto: none
  - **min-height**: altura mínima de un elemento
  - **max-height**: altura máxima de un elemento

    - Se pueden dar como un valor numérico o un porcentaje

```
#contenedor {
  min-width: 300px;
  max-width: 800px;
}
```

Adaptación de una página para impresión

- Para imprimir una página lo normal es eliminar menús y otros elementos, dejando el contenido básico con un formato más apropiado para la impresión
  - Recomendaciones adaptadas de:

http://librosweb.es/css/capitulo_13/version_para_imprimir.html

- Pasos para definir un formato de impresión:

1. Identificar el estilo de impresión (por ejemplo, en otro fichero .css):

```
<link rel="stylesheet" type="text/css"
          href="/css/imprimir.css"
 media="print" />
```

2. Ocultar los elementos que no se van a imprimir

```
#cabecera, #menu, #lateral, #comentarios {
  display: none !important;
}
```


Adaptación de una página para impresión

3. Cambiar la estructura de la página

- Si se han eliminado las columnas laterales, es conveniente reajustar la anchura y el posicionamiento de las zonas a imprimir

```
body, #contenido, #principal, #pie {
  float: none !important;
  width: auto !important;
  margin:  0 !important;
  padding: 0 !important;
}
```

4. Modificar colores, tipos de letra y otros elementos

- Por ejemplo, poner el texto en negro y poner un tipo de letra habitual para las impresoras

```
body {
color: #000; font: 100%/150% Georgia, "Times New Roman", Times, serif;
};
```

- Si se quiere que aparezcan impresas las URL de los enlaces, se puede hacer con la propiedad *content* y el *pseudo-elemento :after*

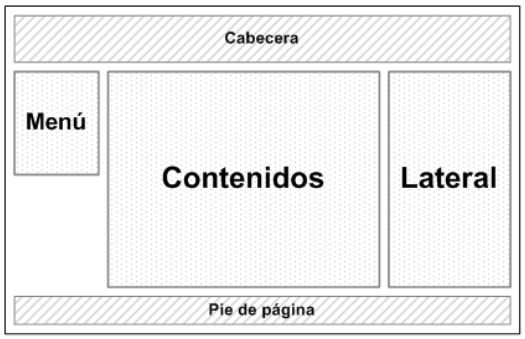
```
a:after {  content: " (" attr(href) ") "; }
```

Ejercicios de posicionamiento

- Realizar la página centrada de manera que ocupe el 80% respecto al ancho de la pantalla
- Centrar una página en horizontal y vertical
  - Solución: *http://librosweb.es/css/capitulo_12/centrar_una_pagina_verticalmente.html*
- Crea la barra siguiente al final de una página

<<Atrás Siguiente>>

- Diseñar una página con tres columnas:



Texto

- Propiedades tipográficas
  - **color**
    - <color> | inherit
  - **font-family**
    - ((<nombre_familia> | <familia_generica>) (,<nombre_familia> | <familia_generica>)*) | inherit
    - Familias genéricas: serif (*Times New Roman*), sans-serif (*Arial*), cursive (*Comic Sans*), fantasy (*Impact*) y monospace (*Courier New*)
  - **font-size**
    - <tamaño_absoluto> | <tamaño_relativo> | <medida> | <porcentaje> | inherit
    - **font-weight**
    - normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | inherit
  - **font-style**
    - normal | italic | oblique | inherit
  - **font-variant**
    - normal | small-caps | inherit
  - **font**
    - ((<font-style> || <font-variant> || <font-weight>)? <font-size> (/ <line-height>)? <font-family>) | caption | icon | menu | message-box | small-caption | status-bar | inherit

Texto

- Apariencia del texto
  - **text-align**
    - left | right | center | justify | inherit
  - **line-height**
    - normal | <numero> | <medida> | <porcentaje> | inherit
  - **text-decoration**
    - none | (underline || overline || line-through || blink) | inherit
  - **text-transform**
    - capitalize | uppercase | lowercase | none | inherit
  - **vertical-align**
    - baseline | sub | super | top | text-top | middle | bottom | text-bottom | <porcentaje> | <medida> | inherit
  - **text-indent**
    - <medida> | <porcentaje> | inherit
  - **letter-spacing** y **word-spacing**
    - normal | <medida> | inherit
  - **white-space** – tratamiento de los espacios en blanco
    - normal | pre | nowrap | pre-wrap | pre-line | inherit

Imágenes

- Propiedades:
  - Anchura y altura: width y height
  - Borde

```
img {
  width: 120px;
  height: 250px;
  border: none;
}
```

- En general no es habitual dar las mismas dimensiones a todas las imágenes y se suele definir su tamaño directamente en el código HTML

```
<img src="imagen.png" width="120" height="250" />
```

Listas

- Propiedades:
  - **list-style-type**
    - disc | circle | square | decimal | decimal-leading-zero | lower-roman | upper-roman | lower-greek | lower-latin | upper-latin | armenian | georgian | lower-alpha | upper-alpha | none | inherit
  - **list-style-position**
    - inside | outside | inherit
  - **list-style-image**
    - <url> | none | inherit
  - **list-style**
    - (<list-style-type> || <list-style-position> || <list-style-image>) | inherit

- Se pueden crear **menús** verticales, horizontales y tabulados a partir de listas
  - Ver pasos en http://librosweb.es/css/capitulo_9.html

Formularios

- Mejoras en los campos de texto
  - Adición de un relleno en los elementos <input> para que no aparezcan pegados

```
form.elegante input { padding: .2em; }
```

  - Etiquetas alineadas y formatedas
    - Ver http://librosweb.es/css/capitulo_11.html
  - Formularios con varias columnas
    - Ver http://librosweb.es/css/capitulo_11/estilos_avanzados.html

Ejercicio final

- Crea tu página personal separando claramente contenidos de presentación en varios ficheros .html y .css
  - Define claramente la estructura de los contenidos de la página
  - Crea una versión de estilos general, una particularizada para impresión, y otra para visualizar en dispositivos móviles
  - Estructura bien los ficheros .css

```
@import url("basico.css") screen;
@import url("seccion.css") screen;
@import url("impresora.css") print;
@import url("movil.css") handheld;
```

- Valida las hojas de estilo en *http://jigsaw.w3.org/css-validator*
- Consulta sitios web con buen diseño:
  http://librosweb.es/css/capitulo_14/sitios_web_de_inspiracion.html

Bibliografía

- http://librosweb.es/css
  - Muchos ejemplos y ejercicios
- http://www.w3schools.com/css
  - Permite probar ejercicios y modificarlos en el navegador directamente
- Estándar:
  - http://www.w3.org/TR/CSS1
  - http://www.w3.org/TR/CSS21