

Hojas de estilo CSS



CSS: Cascading Style Sheet

- CSS = **Hojas de Estilo en Cascada** (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento
- Se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación.
- Permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo.
- Cualquier cambio en el estilo para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento.

¿Por qué usar CSS?

- Los documentos HTML son más claros porque se enfocan en el contenido y no en la apariencia.
 - Permite controlar de manera centralizada el aspecto de un sitio, no importa lo grande que sea.
- Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario.
 - Permite definir diferentes estilos para diferentes medios o dispositivos (pantallas, impresoras, móviles, proyectores, etc.)
- Reduce tiempo en el diseño y la programación de páginas de Internet
- ¿Por qué cascada? porque los estilos “caen” y se aplican de arriba hacia abajo. Se puede hacer un “override” de un estilo, quedando aplicado siempre el más “cercano”

Usando una plantilla CSS (1)



Para cambiar el aspecto general de un sitio
basta con cambiar la plantilla CSS

Usando otra plantilla CSS (2)





UNIVERSIDAD NACIONAL DEL COMAHUE
Facultad de Informática



Martes, 22 de Septiembre de 2015 00:00

Nuevo Master en Software y Sistemas

Tenemos el agrado de informar que en Julio de 2015, el **Lic. Maximiliano Klemen**, docente y egresado de nuestra Facultad, aprobó su Tesis titulada "**Improved static analysis and verification of energy consumption and other resources via abstract interpretation**" obteniendo el título de **Master en Software y Sistemas**, en la Universidad Politécnica de Madrid.

Un orgullo para todos tus profesores y compañeros de la Fal. ¡**Felicitaciones Maxi!**

Martes, 22 de Septiembre de 2015 00:00

Mesas de Examen - Turno Extraordinario Septiembre 2015

Se encuentran disponibles las mesas y fechas de exámenes correspondientes al Turno Extraordinario de Septiembre para las asignaturas dictadas por Departamentos de otras unidades académicas (FaCE – FaEA) [[ver pdf](#)]

Martes, 22 de Septiembre de 2015 00:00

Pasantía Fal: Búsqueda de alumnos avanzados para ser Tutores de Pares

Llamado a inscripción de alumnos avanzados de las carreras de la Facultad para participar del Proyecto "**Consolidación de Tutorías de Pares**". Las actividades a realizar es acompañar a los alumnos del primer año en su inserción en el medio universitario.

INSTITUCIONAL

- Autoridades
- Staff administrativo
- Nuestra historia
- Acceso documentación CD
- Acceso a Despacho
- UNComa
- Nuestro logo

ACADÉMICA

- Ingresantes
- Alumnos
- Carreras
- Docentes

Para cambiar el aspecto general de un sitio
basta con cambiar la plantilla CSS

Usando otra plantilla CSS (3)



Para cambiar el aspecto general de un sitio
basta con cambiar la plantilla CSS

Cómo aplicar las reglas de estilo

- **Hoja Externa**

- La hoja de estilo se almacena en un archivo diferente al HTML con extensión CSS
- Mismo archivo de estilo sirve para muchos HTML
- Es la manera más potente, porque separa las reglas de formateo de la estructura básica de la página

- **Hoja Interna**

- Está incrustada dentro del documento HTML.
- Los estilos aplican solamente a ese HTML.

- **Estilo en Línea**

- Directamente dentro de una etiqueta HTML.
- Se recomienda utilizar solo de manera ocasional.

¿Cómo funciona?

- Sin CSS

```
<h1><font color="red" face="Arial" size="5">Titular de la página</font></h1>
```

```
<p><font color="grey" face="Verdana" size="2">Un párrafo de texto no muy largo.</font></p>
```

- Con CSS:

```
<style type="text/css">
```

```
h1 { color: red; font-family: Arial; font-size: large; }
```

```
p { color: grey; font-family: Verdana; font-size: medium; }
```

```
</style>
```

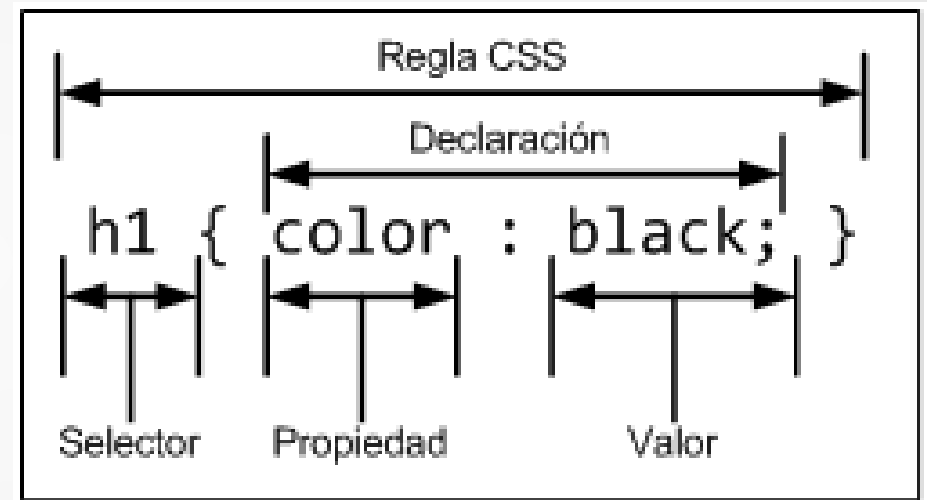
- El CSS aplica estilo a etiquetas <h1> y <p>
 - Igual para todos los <p> y <h1> que haya en la página
 - Todos mostrarán la misma fuente y color (o sea, estilo)

Cómo usar CSS en un HTML

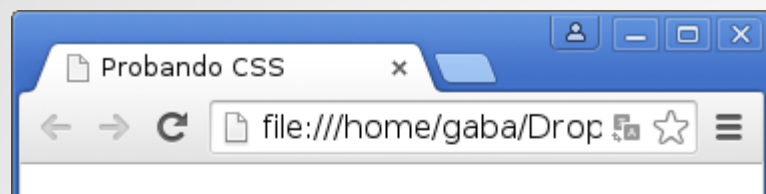
- En el mismo documento HTML
 - Dentro del header
 - `<style type="text/css"> ... </style>`
 - Dentro de un elemento HTML particular
 - `<p style="color: black; font-family: Verdana;">Un párrafo de texto.</p>`
- CSS en un archivo externo, invocándolo dentro del header del HTML
 - `<link rel="stylesheet" type="text/css" href="/css/estilos.css"/>`

Composición de una regla CSS

- **Regla:** cada uno de los estilos que componen una hoja de estilos CSS.
- Cada regla se compone de:
 - **Selector:** indica el elemento o elementos HTML a los que se aplica la regla CSS.
 - **Declaración:** estilos que se aplicarán (una o más propiedades CSS)
 - **Propiedad:** aspecto de una característica del elemento.
 - **Valor:** nuevo valor de la característica modificada en el elemento



Ejemplo para redefinir etiquetas h

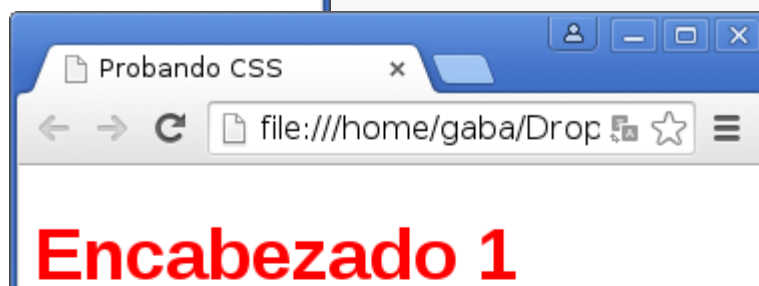


Encabezado 1

Encabezado 2

Encabezado 3

**sin
CSS**



Encabezado 1

Encabezado 2

Encabezado 3

**con
CSS**

Dentro de
<header> ...
</header>

```
<style type="text/css">
h1, h2, h3
{ font-weight: normal;
  font-family: Arial, Helvetica, sans-serif }
</style>
```

```
<style type="text/css">
h1 { color: #FF0000; font-size: 36px; }
h2 { color: #00FF00; font-size: 24px; }
h3 { color: #0000FF; font-size: 18px; }
</style>
```

Selectores Básicos de CSS

- **Selector universal:** representa a todos los elementos de una página. Ej: quitar márgenes a todos los elementos de la página
`* { margin: 0; padding: 0; }`
- **Selector de tipo o etiqueta:** todos los elementos de la página cuya etiqueta HTML coincide con el valor del selector. Ej: usar fuente Arial en todos los elementos párrafo <p>
`p { font-family:Arial; }`
- **Selector descendente:** Selecciona elementos que se encuentran dentro de otros elementos. Ejemplo: en rojo todos los elementos de la página que se encuentren dentro de un elemento <p>
`p span { color: red; }`

Selector de Clase

- Son los selectores más utilizados → Buena práctica
- Se define un estilo propio con un nombre empezado con “.”

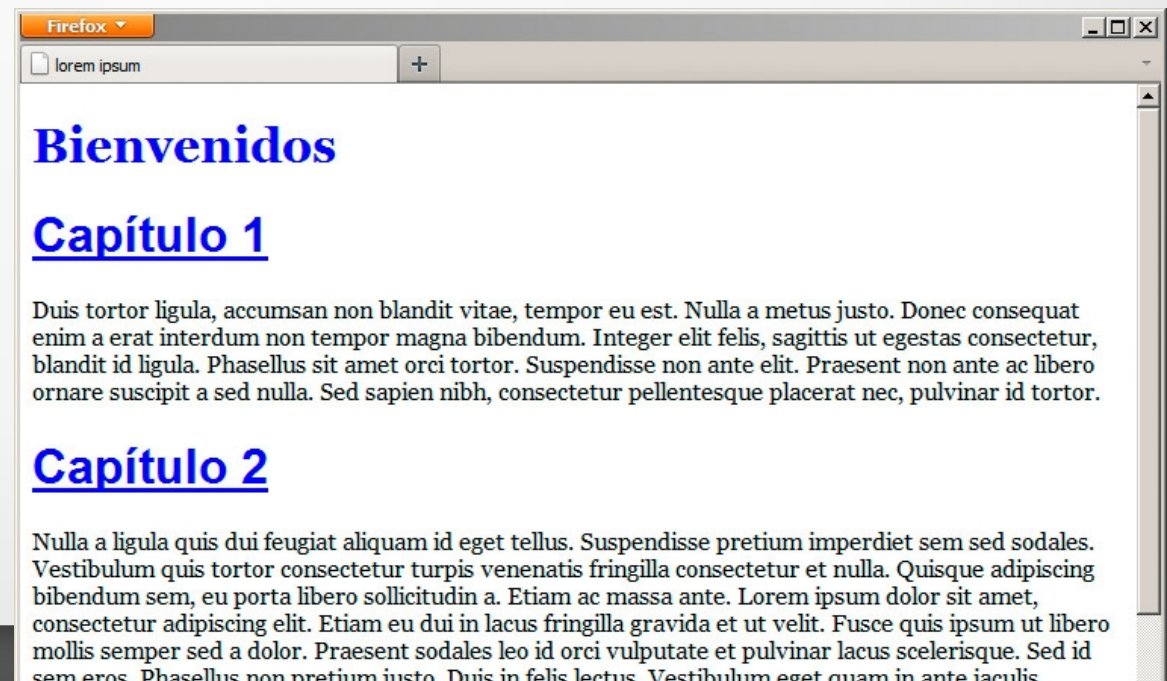
```
.titcol {  
    font-size : 10pt; font-family : verdana,arial,helvetica;  
    font-weight : bold; background-color : #F3F1D1;  
    background-image : url(images/fontit2.gif);  
    color : #333333; border-bottom: 1px solid #cccccc;  
}
```

- Para usarlo:
`<td class=titcol> Comentarios sin revisar</td>`
- En un archivo css puede haber tantas clases como hagan falta, centralizando la presentación de un sitio. El cambio en el css impactará instantaneamente en todos los markups de esa clase.

Ejemplo: Selector de clase

- Se desea cambiar la apariencia de algunos `<h1>`, pero no de todos: se define una clase **capítulo**
`<h1>Bienvenidos</h1>`
`<h1 class="capitulo">Capítulo 1</h1>`
`<h1 class="capitulo">Capítulo 2</h2>`
- La clase "capítulos" cambia el tipo de letra y el subrayado:
`.capitulos { font-family: sans-serif; text-decoration: underline;}`

Cómo se vería
en la página



Clases dependientes

- Además, las reglas de una clase se pueden aplicar exclusivamente a una etiqueta. Por ejemplo:
 - `h1.capitulos { font-family: sans-serif; text-decoration: underline; font-size:14px}`
 - `p.capitulos { font-family: sans-serif; text-decoration: underline;}`
- De este modo se puede reutilizar el nombre de la clase (capítulos) y definir distintos valores, dependiendo de si se aplica a una cabecera (h1) o a un párrafo (p), como en el ejemplo (la diferencia, en este caso, es el tamaño de la letra)

Selector ID: para elemento único

- Para aplicar estilos CSS a un único elemento de la página.
 - Permite seleccionar un elemento de la página por el valor de su atributo **id**
- Sintaxis muy parecida a selectores de clase, pero se utiliza el símbolo numeral (#) en lugar del punto (.)
- Ejemplo definición CSS
`#destacado { color: red; }`
- Ejemplo uso
`<p>Primer párrafo</p>`
`<p id="destacado">Segundo párrafo</p>`
`<p>Tercer párrafo</p>`

Sólo este párrafo va a estar en rojo. El resto va a tener el color heredado

Combinación de selectores básicos

- Los selectores se combinan para restringir el alcance de las reglas CSS.
- Ejemplo 1: seleccionar elementos con class="especial" que se encuentren dentro de cualquier elemento con un class="aviso"
`.aviso .especial { ... }`
- Ejemplo 2: seleccionar elementos de tipo con un atributo class="especial" que estén dentro de cualquier elemento de tipo <div> que tenga un atributo class="aviso"
`div.aviso span.especial { ... }`

Selector de hijos (>)

- Similar al selector descendente, pero diferente en su funcionamiento.
 - Se utiliza para seleccionar un elemento que es hijo directo de otro elemento. Se indica mediante el signo “mayor que” (>):
 - En CSS: `p > span { color: blue; }`
 - En HTML:
 - (1) `<p>Texto1</p>`
 - (2) `<p>Texto2</p>`
- En (1) `` cumple la condición del selector.
En (2) no se cumple porque `` es descendiente pero no es hijo directo de `<p>`

Selector adyacente (+)

- Se emplea para seleccionar elementos que se encuentran justo a continuación de otros elementos.
- Sintaxis : **elemento1 + elemento2 { ... }**
- Sea el siguiente HTML:
`<h1>Titulo1</h1>`
`<h2>Subtítulo</h2>`
`<h2>Otro subtítulo</h2>`
- Hay dos elementos `<h2>`, pero sólo uno de ellos se encuentra inmediatamente después del elemento `<h1>`. Si se quiere aplicar diferentes colores en función de esta circunstancia, el **selector adyacente** es el más adecuado:

`h2 { color: green; }`
`h1 + h2 { color: red }`
- Todos los `<h2>` de la página se verán verde, salvo aquellos `<h2>` que se encuentran inmediatamente después de cualquier elemento `<h1>` (Ej: Subtítulo), que se mostrarán de color rojo

Cuándo usar `` o `<div>`

- ¿Cuándo se usa **SPAN**?
 - Aplicar estilos a frases de un párrafo (no más de una línea)
 - Definir padding de un enlace, creando el efecto de botón
 - Agrupar varios elementos en una línea (imágenes, textos, enlaces)
- ¿Cuándo se usa **DIV**?
 - Maquetación web: crear la estructura de un sitio web
 - Crear contenido flotante: crear un menú, imagen u otro elemento flotante (usando la propiedad CSS float)
 - Contenedor de más contenedores: se puede poner un div dentro de otro div y así sucesivamente ... sin exagerar

Propiedades de tipografía

- **Font-family:** especifica el tipo de letra deseado
 - Nombres de familia: "times", "courier", "arial", ...
 - Familia genérica: "serif", "sans-serif", "cursive", "fantasy", "monospace"
 - Se empieza listando el que uno desea y luego algunos alternativos. El último debe ser un nombre genérico, que seguramente el browser va a encontrar alguno
- Ejemplo CSS
 - `.textoComun {font-family: Arial, Helvetica, sans-serif; }`
 - `h2 { font-family: "Courier New", Courier, monospace;}`

Propiedades de tipografía (2)

- **Font-size:** establece tamaño de forma relativa o absoluta
 - Tamaño absoluto:
 - Identificadores: `xx-small`, `x-small`, `small`, `medium`, `large`, `x-large`, `xx-large`
 - Puntos: `10pt` (1 punto equivale a 1 pulgada/72, es decir, 0.35 mm)
 - Tamaño relativo: Se toma como base el tamaño de letra del elemento contenedor
 - Identificadores: `larger` y `smaller`
 - Porcentaje: `50%` achica a la mitad, `150%` aumenta el 50% del actual
 - Fracción: `1.5em` (idem 150% anterior)
 - Pixel: `10px` son 10 pixeles. Depende de la resolución de la pantalla
- Ejemplos:
 - `body {font-size: 12px;}`
 - `h1 {font-size: larger;}`
 - `p {font-family: Arial, Helvetica, sans-serif; font-size: 1.5em;}`

Propiedades de tipografía (3)

- **Font-weight:** define grosor
 - Valores posibles: normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | inherit
 - Ejemplos:

```
body { font-weight: 500; }  
p {font-weight: bold;}
```
- **Font-style:** define estilo
 - Valores posibles: normal | italic | oblique | inherit
 - Ejemplos:

```
body {font-style: normal; }  
.textoNoticias {font-style: italic; }
```

Propiedades de tipografía (4)

- **font-variant** permite elegir una variante de la propiedad font-family
- Variantes disponibles en CSS 2.1: normal (**normal**) y versalita (**small-caps**).
 - La variante versalita (small-caps) muestra el texto con letras mayúsculas de un tamaño similar a las letras minúsculas.
- Ej:
 - Normal: Lorem ipsum Dolor sit Amet
 - Mayúsculas: LOREM IPSUM DOLOR SIT AMET
 - Versalitas: LOREM IPSUM DOLOR SIT AMET

Modelo de cajas

- "Box model": característica más importante del CSS
- Condiciona el diseño de todas las páginas web: Todos los elementos de la página se representan mediante cajas rectangulares
- Cada etiqueta HTML crea una nueva caja rectangular que encierra los contenidos de ese elemento

`<p>Párrafo de texto con algunas palabras resaltadas</p>`

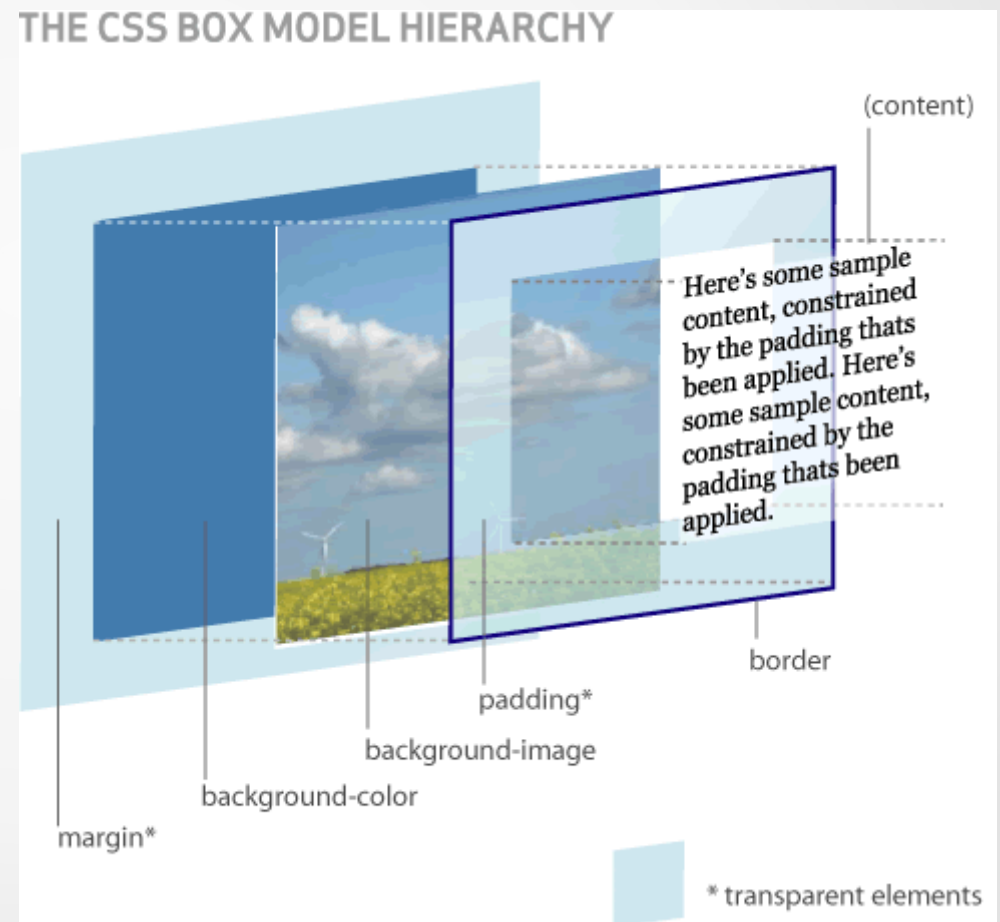
`<p>Otro párrafo</p>`

Párrafo de texto con **algunas palabras** resaltadas

Otro párrafo

Partes que componen una caja

- Contenido (**content**): palabras, imagen, lista de elementos, etc
- Relleno (**padding**): espacio libre opcional entre el contenido y el borde
- Borde (**border**): línea que encierra el contenido y su relleno
- Imagen de fondo (**background image**): imagen que se muestra detrás del contenido (tiene más prioridad que el color)
- Color de fondo (**background color**): color que se muestra detrás
- Margen (**margin**): separación opcional existente entre la caja y el resto de cajas adyacentes



Propiedades de una caja

- Ancho y altura: **width** y **height**

- Medida relativa (em, ex, px), Medida absoluta (in, mm, cm, pc, pt), Porcentaje (%)

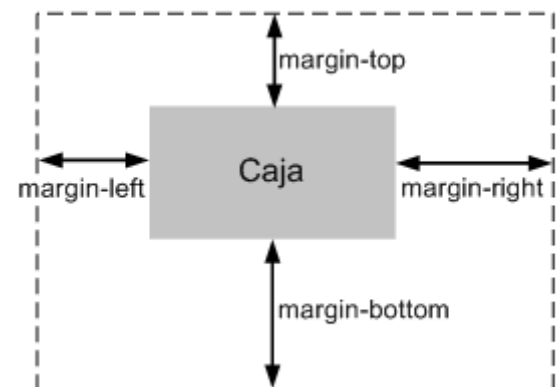
Ejemplo: valor de la altura del elemento <div> de cabecera

En CSS: `#cabecera { height: 60px; }`

En HTML: `<div id="cabecera"> ... </div>`

- Márgenes: controla espacios fuera de la caja

- **margin-top, margin-right, margin-bottom, margin-left**
- Unidades más usadas: píxeles (precisión total), em (para mantener proporciones) y porcentajes (para diseños fluidos)
- Ejemplo CSS: `.destacado { margin-left: 2em; }`

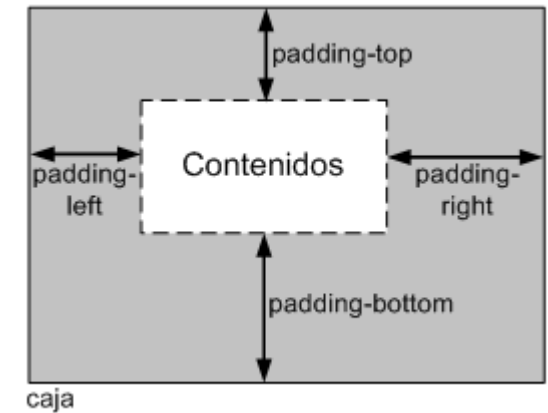


Propiedades de una caja

- Márgenes: otra manera
 - **margin**: permite definir más de un margen a la vez
 - 1 valor: todos los márgenes tienen ese valor.
 - 2 valores: 1º margen superior e inferior y 2º márgenes izquierdo y derecho.
 - 3 valores: 1º margen superior, 3º margen inferior y 2º márgenes izquierdo y derecho.
 - 4 valores: orden de asignación superior, derecho, inferior e izquierdo.
 - Ejemplo: `div img { margin-top: .5em; margin-bottom: .5em; margin-left: 1em; margin-right: .5em; }`
 - Equivale a: `div img { margin: .5em .5em .5m 1em; }`
Otra alternativa: `div img { margin: .5em; margin-left: 1em; }`

Propiedades de una caja

- Relleno: controla espacios de relleno horizontales y verticales dentro de la caja
 - **padding-top, padding-right, padding-bottom, padding-left**
 - **padding**: permite definir los 4 a la vez de forma equivalente a **margin**



Ejemplos:

`body {padding-top: 1em; padding-left: 2em} /* Sup = 1em, Izq = 2em */`

`body {padding: 2em} /* Todos los rellenos valen 2em */`

`body {padding: 1em 2em} /* Sup e inf = 1em, Izq y der = 2em */`

`body {padding: 1em 2em 3em} /* Sup = 1em, der = 2em, inf = 3em, izq = 2em */`

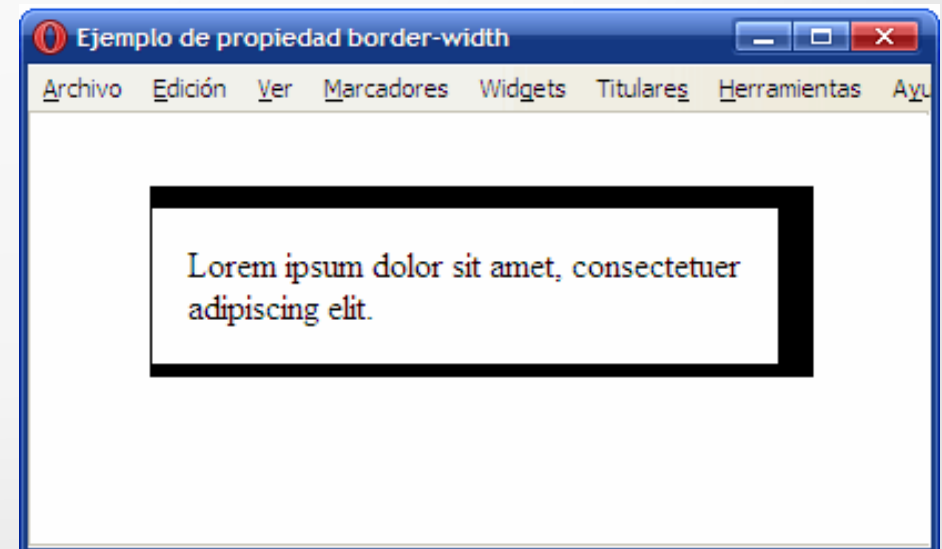
`body {padding: 1em 2em 3em 4em} /* Sup = 1em, der = 2em, inf = 3em, izq = 4em */`

Propiedades de los Bordes

- Anchura bordes

- Propiedades: **border-top-width**, **border-right-width**, **border-bottom-width**, **border-left-width**
- Valores: unidad de medida ó thin | medium | thick
- Ejemplo:

```
div { border-top-width: 10px;  
      border-right-width: 1em;  
      border-bottom-width: thick;  
      border-left-width: thin;}
```
- **border-width** permite definir todos los bordes a la vez,
Idem **margin** y **padding**

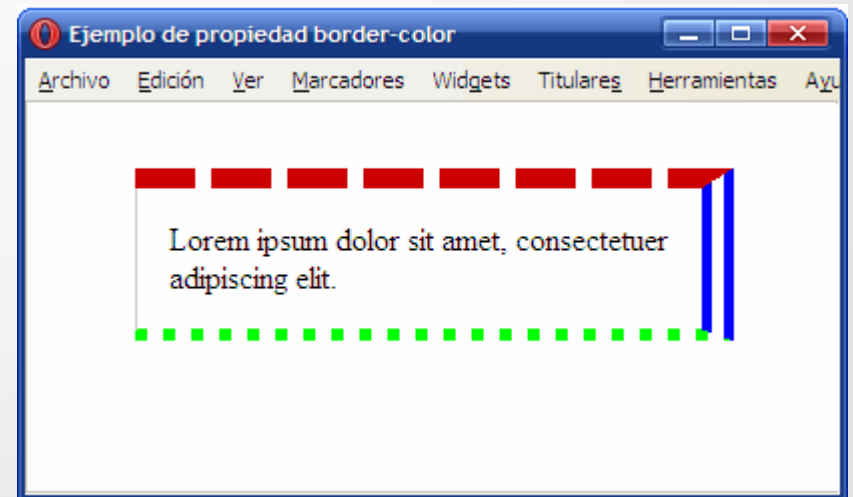


Otras propiedades de borde

- Color: color de la línea
 - border-color (idem margin)
 - border-top-color, border-right-color, border-bottom-color, border-left-color
- Estilo: Estilo de la línea (dashed, dotted, solid, double ...)
 - border-style (idem margin)
 - border-top-style, border-right-style, border-bottom-style, border-left-style

- Ejemplo

```
div { border-top-color: red;  
border-top-width: 3px; border-top-style: dashed;  
border-right-width: 3px; border-right-style: double;  
border-right-color: blue; border-bottom-width: 1px;  
border-bottom-style: dotted;  
border-bottom-color: green;}
```



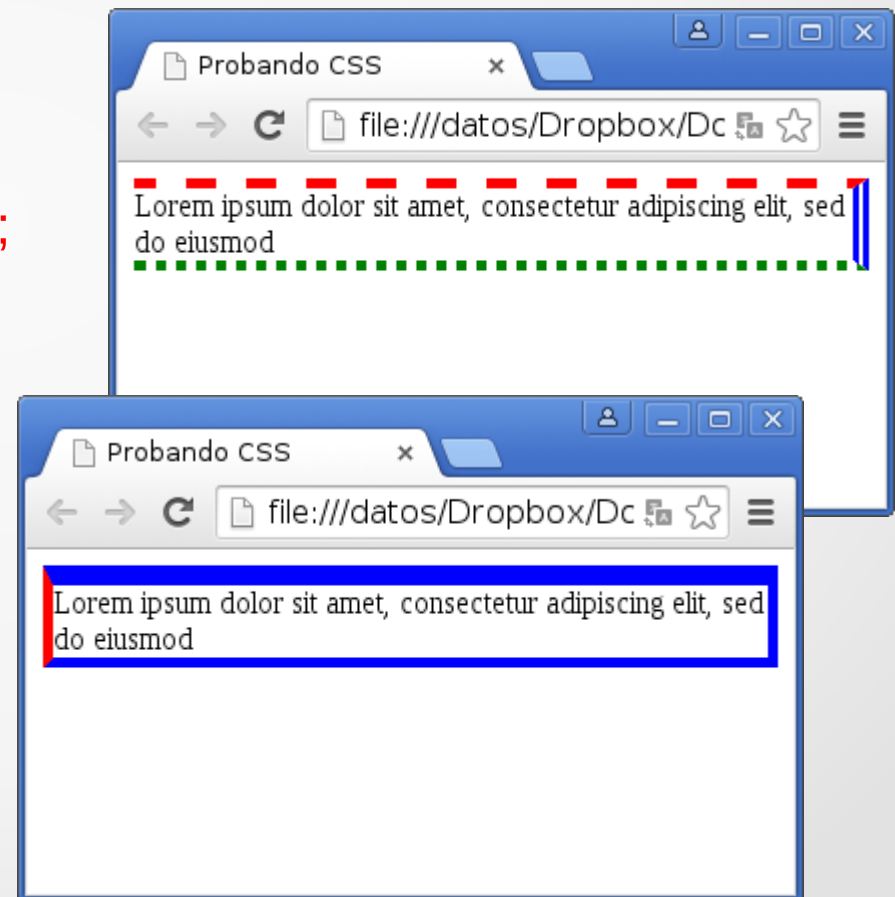
Otras propiedades de borde

- Más shorthands para borde!
 - **border-top, border-right, border-bottom, border-left**

```
div {  
  border-top: 5px dashed red;  
  border-right: 8px double blue;  
  border-bottom: 5px dotted green;  
}
```

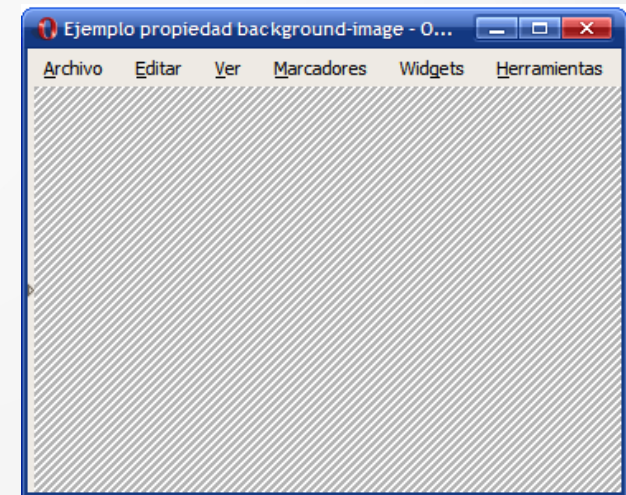
- **border** (define en general y luego se redefine con las propiedades más finas)

```
div { border: 5px solid blue;  
  border-top-width: 10px;  
  border-left-color: red;  
}
```



Relleno y fondo de cajas

- **background-color**: bloque de color
Ej: `body { background-color: #F5F5F5; }`
- **background-image**: coloca una imagen a partir de una url
 - Ej: `body { background-image: url("imagenes/fondo.png"); }`
 - Sea fondo.png una imagen pequeña, por defecto se repite tantas veces como haga falta para cubrir el espacio (repeat)
- También se puede usar **background-repeat** para repetir en un único sentido (repeat-x | repeat-y) o no repetirla (no-repeat)



Relleno y fondo de cajas (2)

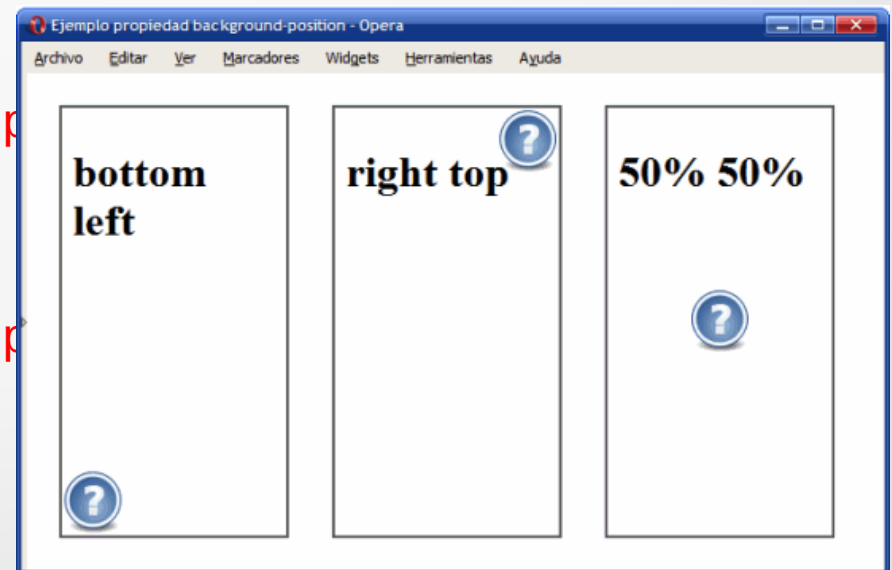
- Además de el tipo de repetición se puede controlar la posición de la imagen dentro del elemento
- **background-position**, recibe 2 valores que indican el desplazamiento dentro del elemento

- Ejemplos:

```
#caja1 { background-image: url("images/help.png");  
background-repeat: no-repeat;  
background-position: bottom left;}
```

```
#caja2 { background-image: url("images/help.p  
background-repeat: no-repeat;  
background-position: right top;}
```

```
#caja3 { background-image: url("images/help.p  
background-repeat: no-repeat;  
background-position: 50% 50%;}
```



Relleno y fondo de cajas (3)

- **background-attachment** (scroll | fixed)
 - Controla la forma en la que se visualiza la imagen de fondo: permanece fija cuando se hace scroll en la ventana del navegador o se desplaza junto con la ventana
 - Es un comportamiento que suele dar problema en algunos navegadores (no se recomienda)
- También hay un shorthand para **background**
 - Se puede definir color, imagen de fondo, y posición en un paso
`body { background: #222d2d url(./graphics/colorstrip.gif) repeat-x 0 0; }`
- Equivale a
 - `body { background-color: #222d2d;
background-image: url("./graphics/colorstrip.gif");
background-repeat: repeat-x; background-position: 0 0;}`

Dando estilo a los enlaces

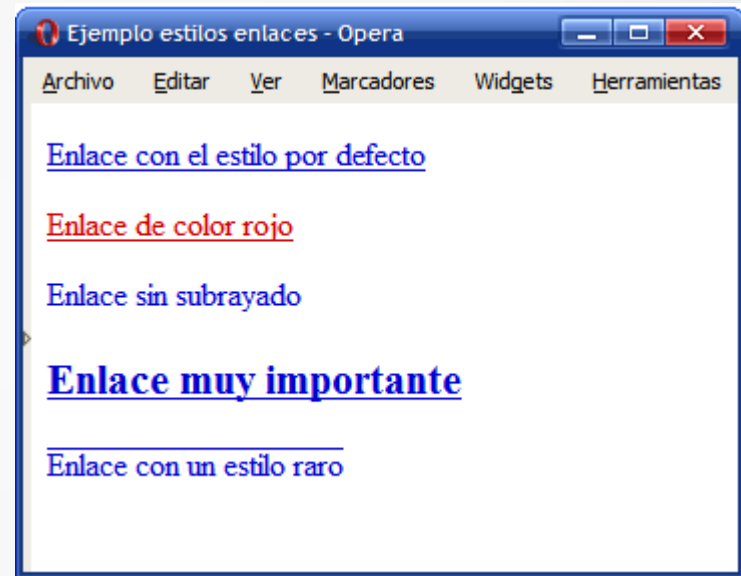
- Se personalizan definiendo la clase en base al color y estilo del texto

- Ejemplo en CSS

```
a { margin: 1em 0; display: block;}  
.alternativo {color: #CC0000;}  
.simple {text-decoration: none;}  
.importante {font-weight: bold;  
             font-size: 1.3em;}  
.raro {text-decoration: overline;}
```

- Como se usa en HTML

```
<a href="#">Enlace con el estilo por defecto</a>  
<a class="alternativo" href="#">Enlace de color rojo</a>  
<a class="simple" href="#">Enlace sin subrayado</a>  
<a class="importante" href="#">Enlace muy importante</a>  
<a class="raro" href="#">Enlace con un estilo raro</a>
```



Dando estilo a los enlaces (2)

- Personalizar aspecto de los enlaces en función de su tipo

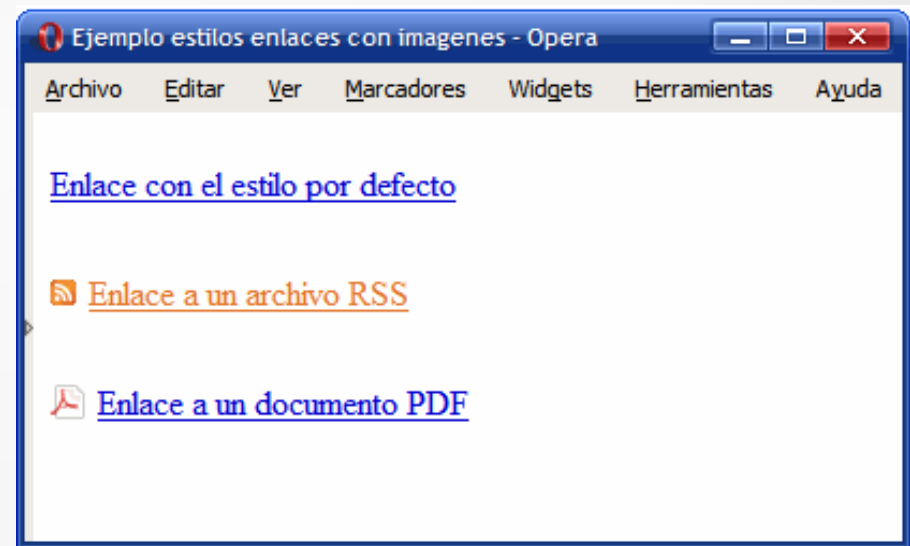
- En CSS

```
.rss { color: #E37529; padding: 0 0 0 18px;  
      background: #FFF  
      url(imagenes/rss.gif)  
      no-repeat left center;}
```

```
.pdf { padding: 0 0 0 22px;  
       background: #FFF  
       url(imagenes/pdf.png)  
       no-repeat left center;}
```

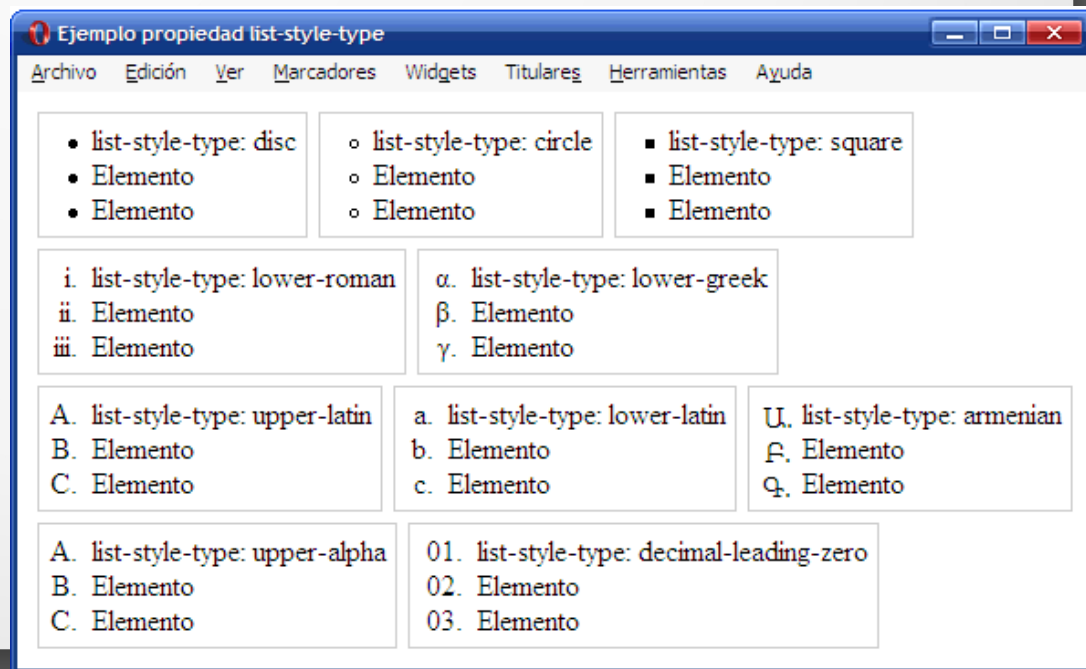
- En HTML

```
<a href="#">Enlace con el estilo por defecto</a>  
<a class="rss" href="#">Enlace a un archivo RSS</a>  
<a class="pdf" href="#">Enlace a un documento PDF</a>
```



Personalizando viñetas

- La propiedad `list-style-type` permite definir tres tipos de viñetas
 - Gráficos: `disc`, `circle` y `square`
 - Numéricos: `decimal`, `decimal-leading-zero`, `lower-roman`, `upper-roman`, `armenian` y `georgian`.
 - Alfanuméricos: `lower-latin`, `lower-alpha`, `upper-latin`, `upper-alpha` y `lower-greek`
- También se puede usar una imagen como viñeta seteando la propiedad **`list-style-image`**



Centrar página ancho fijo

- Con resoluciones superiores a 1024 x 768 píxel, las líneas de texto son demasiado largas para leerlas con comodidad.
- Además, los navegadores alinean las páginas a la izquierda de la ventana por defecto.
- Conviene crear páginas con anchura fija adecuada y centrar la página horizontalmente



Centrar página ancho fijo (2)

- Se usa la propiedad margin de CSS
 - Agrupar todos los contenidos de la página en un elemento <div> y asignarle a ese <div> márgenes laterales automáticos.
 - El <div> que encierra los contenidos se suele llamar contenedor (en inglés wrapper o container):

- En CSS

```
#contenedor { width: 300px; margin: 0 auto; } /* ancho fijo 300px */  
#contenedor { width: 70%; margin: 0 auto; } /* ancho variable 70% pantalla */
```

- En HTML

```
<body>  
  <div id="contenedor">  
    <h1>Lorem ipsum  
      dolor sit amet</h1>  
  
    ...  
  </div>  
</body>
```



Layout a 2 columnas (cab y footer)

- Se usa la propiedad **float** para el menú (left) y el contenido (left o right)
- Se usa **clear** en el pie de página (footer) para evitar solapamiento con los otros elementos posicionados con float

- En CSS

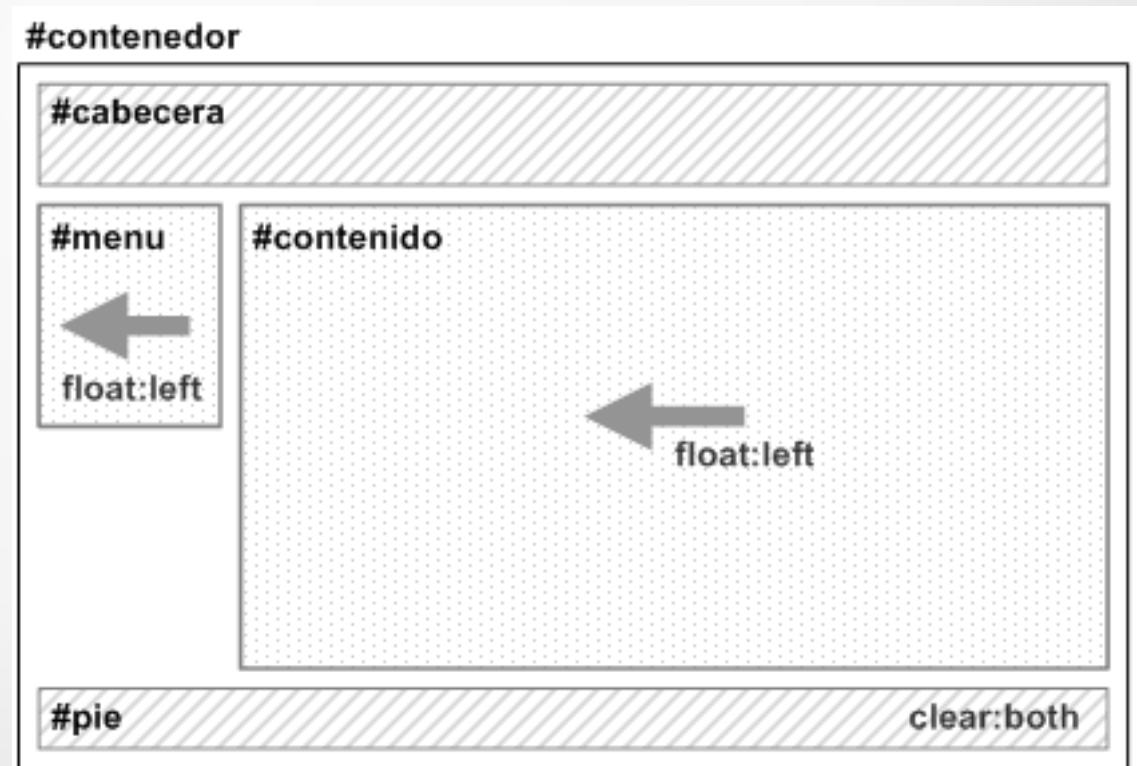
```
#contenedor {  
    width:.....; margin: auto; }
```

```
#cabecera { ...  
    definir alto,  
    decoración, etc ... }
```

```
#menu {  
    float: left;  
    width: 15%;}
```

```
#contenido {  
    float: left;  
    width: 85%; }
```

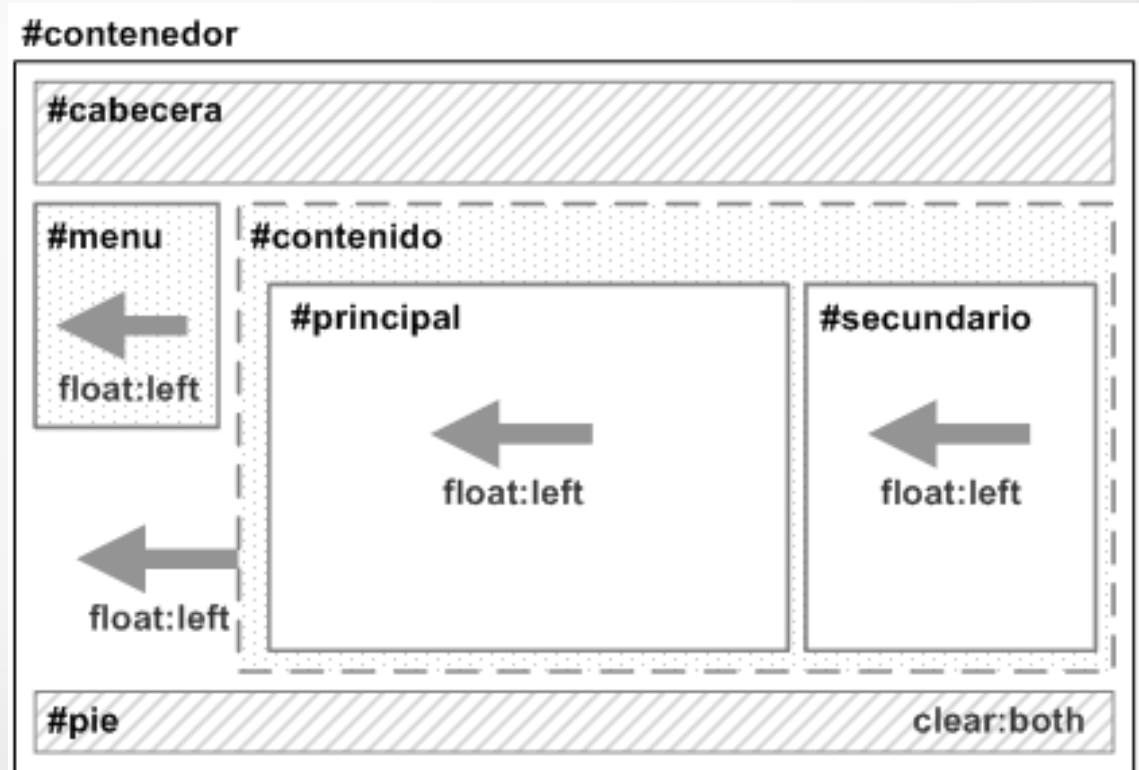
```
#pie {  
    clear: both; }
```



Layout a 3 columnas + cab y footer

- Además del diseño a 2 columnas, el más utilizado es el de 3 columnas con cabecera y pie de página.
- Los contenidos se dividen en 2 zonas: principal de contenidos y lateral de contenidos auxiliares o secundario. Todos se ubican a la derecha (float)
- En CSS

```
#contenedor { .... }  
#cabecera { ..... }  
#menu {  
    float: left; width: 15%;}  
#contenido {  
    float: left; width: 85%;}  
#contenido #principal {  
    float: left; width: 80%;}  
#contenido #secundario {  
    float: left; width: 20%;}  
#pie { clear: both;}
```



Altura/anchura máximas y mínimas

- Se suele optar por diseños de anchura fija o que se adapte a la anchura del navegador
- Generalmente conviene que la anchura de la página sea variable pero respetando ciertos límites:
 - Ni tan pequeña como para que no se vean los contenidos ni tan ancha para que las líneas de texto no puedan leerse cómodamente.
- 4 propiedades que permiten limitar anchura y altura mínima y máxima de cualquier elemento de la página:
 - **max-width, min-width, max-height y min-height**
- Ejemplo:

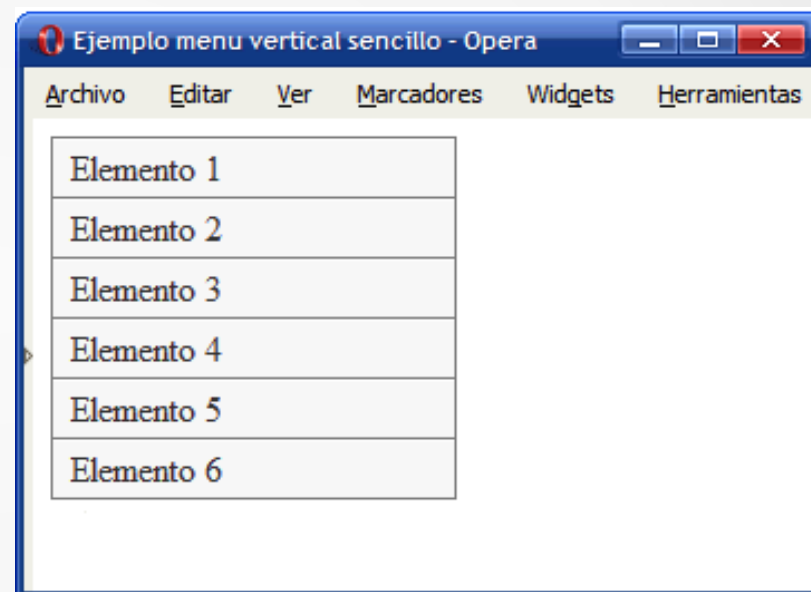
```
#contenedor {  
  min-width: 500px;  
  max-width: 900px; }
```


Cómo crear un menú vertical?

- Desafío: decorar la lista `` para que luzca como un menú vertical

- En HTML

```
<div id="menu">
  <ul>
    <li><a href="#">Elemento 1</a></li>
    <li><a href="#">Elemento 2</a></li>
    <li><a href="#">Elemento 3</a></li>
    <li><a href="#">Elemento 4</a></li>
    <li><a href="#">Elemento 5</a></li>
    <li><a href="#">Elemento 6</a></li>
  </ul>
</div>
```



- ¿En CSS qué hay que hacer?
Ayuda: para definir el estilo de la lista `` que está dentro de un `<div>` que tiene `id="menu"` se debe definir el estilo de **`div#menu ul {...}`**

Cómo crear un menú horizontal?

- Desafío: lograr que la lista `` luzca como un menú horizontal

- En HTML

Elemento 1	Elemento 2	Elemento 3	Elemento 4	Elemento 5
------------	------------	------------	------------	------------

```
<div id="menuHorizontal">
  <ul>
    <li><a href="#">Elemento 1</a></li>
    <li><a href="#">Elemento 2</a></li>
    <li><a href="#">Elemento 3</a></li>
    <li><a href="#">Elemento 4</a></li>
    <li><a href="#">Elemento 5</a></li>
  </ul>
</div>
```

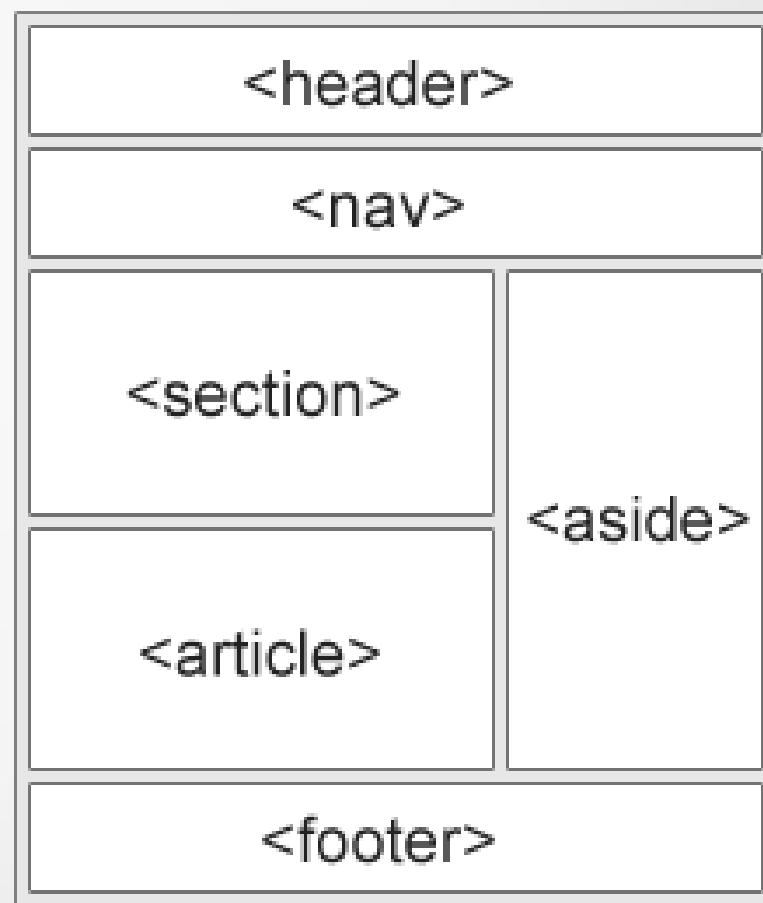
- ¿En CSS qué hay que hacer?

Ayuda: Una vez definido el **id #menuHorizontal** con **clear:both** y el ancho deseado **width**, para que cada elemento `` de la lista `` se pueda colocar en forma horizontal, se define:

```
div#menuHorizontal ul li{ float: left; width: ....%; }
```

Etiquetas semánticas de HTML5

- HTML5 tiene nuevos elementos semánticos para las distintas partes de una página
 - <article>
 - <aside>
 - <details>
 - <figcaption>
 - <figure>
 - <footer>
 - <header>
 - <main>
 - <mark>
 - <nav>
 - <section>
 - <summary>
 - <time>



Etiquetas semánticas de HTML5 (2)

- El comportamiento de estas etiquetas se redefine en el CSS como hicimos antes con el selector ID #menu
- Por qué HTML5 añade estas etiquetas?
 - Al mostrarse la página en dispositivos móviles, teléfonos, etc. el navegador puede tener la opción de esconder algún elemento (ej: el menú de navegación) y tener más espacio para otro contenido importante.
 - Esto es posible si sabe dónde empieza y termina el menú de navegación y eso se logra con la etiqueta nav, gracias a la semántica asociada
 - Idem con footer, header, etc

Ejemplo usando <nav>

- En HTML:

```
<nav>
  <ul>
    <li><a href=#>Elem 1</a></li>
    <li><a href=#>Elem 2</a></li>
    <li><a href=#>Elem 3</a></li>
    <li><a href=#>Elem 4</a></li>
  </ul>
</nav>
```

Elem 1
Elem 2
Elem 3
Elem 4

- En CSS:

```
nav { float: left; width: 15%; }
nav ul { border: 1px solid #7C7C7C; border-bottom: none; list-style: none;
  margin: 0; padding: 0; width: 100%;}
nav ul li { background: #F4F4F4; border-bottom: 1px solid #7C7C7C;
  border-top: 1px solid #FFF;}
nav ul li a { color: #333; display: block; padding: .2em 0 .2em .5em;
  text-decoration: none; }
```

Referencias

- **Introducción a CSS**
<http://www.librosweb.es/libro/css/>
- **El gran libro de HTML5, CSS3 y Javascript**
J. D. Gauchat, 2013, ISBN 9788426719959
- **HTML(5) Tutorial**
http://www.w3schools.com/html/html5_semantic_elements.asp