

Indice:

1 Conectar un fichero HTML a un fichero CSS. Uso de Link.

2 Reglas Básicas

3. Las clases CSS

4. Uso de margin, padding border.

5. Uso de CSS con enlaces

6. Uso de CSS con listas Ol, Ul.

7. so de CSS con imagenes.

8. Uso de CSS con tablas.

9. Uso de CSS con Formularios.

1. Conectar un fichero HTML a un fichero CSS. Uso de Link.

En este ejemplo podemos observar la forma de conectar un fichero HTML con un fichero CSS haciendo uso de la etiqueta <link...>.

Podemos observar también como cambia la sintaxis de los dos lenguajes.

Fichero HTML	Fichero estilos2.css
<pre><html> <head> <link rel="Stylesheet" Type="text/css" href="estilos2.css"> <title>Página que lee estilos</title> </head> <body> <h1>Página que lee estilos</h1> Página que enlaza con una hoja de estilos

 <table width="300" cellspacing="2" border="0"> <tr> <td>Estilo propio dentro de un td</td> </tr> <tr> <td>segunda fila del td</td> </tr> </table> </body> </html></pre>	<pre>P { font-size:12pt; font-family:arial,Helvetica; font-weight:normal; } H1 { font-size:36pt; font-family:verdana,arial; text-decoration:underline; text-align:center; background-color:teal; } TD { font-size:10pt; font-family:verdana,arial; text-align:center; background-color:#666666; } BODY { font-family:arial; background-color#:006666; color:pink; }</pre>

2. En CSS son REGLAS BÁSICAS:

Para definir una propiedad. Se usa **atributo:valor;** **Ej:** font-size:10pt:

Para definir el estilo de una etiqueta. Se usa **Etiqueta{atributo:valor;...}**
Ejemplo: H1 { text-align:center; color:black;}

Mismo estilo a un conjunto de etiquetas. Puedo separarlas por comas. **Ej:** **H1,p**{color:red;}

Unidades de Medidas: relativo, absoluto, porcentaje.

- **Relativas:** se ajustan al medio donde se muestran (portatil. Móvil).

Fuente actual: em.

Ej: Si la fuente por defecto es 12pt 2em será 24 pts.

Altura de la letra: 1ex es la altura de la fuente.

Pixeles: px puntos por pantalla que depende de la resolución.

- **Absolutas** (fijas): pt , cm centímetro, mm milímetro, px pixel.

- **Porcentaje:** %

Si la fuente tiene 12pt y definimos la unidad 150% nos referimos a 18 puntos.

Colores. Basados en el sistemas RGB.

Hexadecimal.- #ffffff

Hexadecimal abreviada.- #fff = #ffffff

Nombre de color.-

Color en %: rgb(33%,0%,0%)

Color en decimales: rgb(200,255,0)

Color transparente:

ejemplo: background:#ff8800 ;

ejemplo: background:#f80:

ejemplo: color:red:

Ejemplo: color:rgb(33%,0%,0%);

Ejemplo: color:rgb(200,255,0);

background-color:transparent;

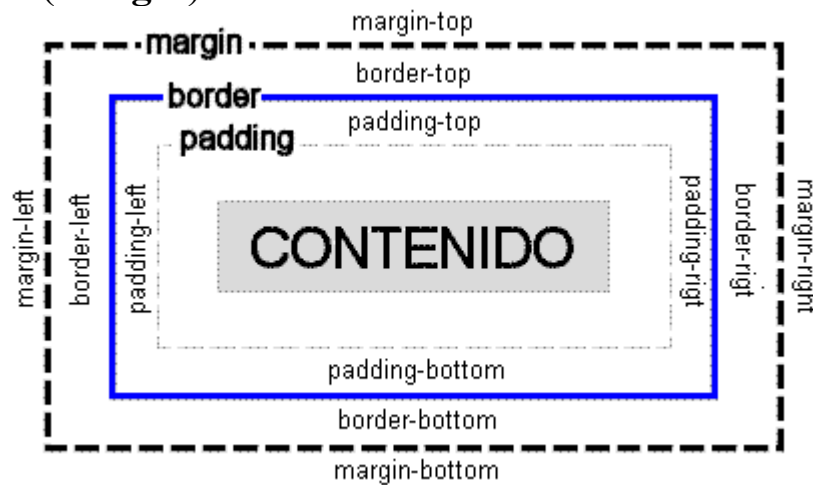
Valor de las URL:

Absoluta.- URL("http://www.iescamilojosecela.com/images/miimagen.gif")

Relativa.- URL("../images/otramagen.jpg")

3. Bordos, márgenes y rellenos

Podemos ver que con estas propiedades configuramos el borde en relación a un texto (**border**), distancia del borde con el texto (**padding**) y distancia de los márgenes al borde (**margin**).



Las propiedades de los bordes son:

border-width: thin (fino), medium (mediano), thick (grande), o 4px (cuatro lados);

border-color: blue (4 bordes azules) o blue red #ffcc00 black (cada borde de un color);

border-style: tipos de bordes; dotted, dashed, solid, double, ridge inset, outset.

Aclarar que podemos indicar características concretas para el borde superior, derecho, inferior e izquierdo con instrucciones como:

Border-width-top:4px; Border-width-right:3px; Border-width-botton:2px; Border-width-top:1px;

Existen también formas abreviadas a la hora de asignar valores:

Border-width:6px; significa que todos los bordes tienen el mismo grosor.

Border-width: 5px,8px; significa que el superior mide 5px, el derecho 8px, el inferior 5px y el izquierdo 8px. (Siguiendo el orden de las agujas del reloj).

Estas características son idénticas para el margin y el padding.

Tipos de bordes:

Dotted

dashed

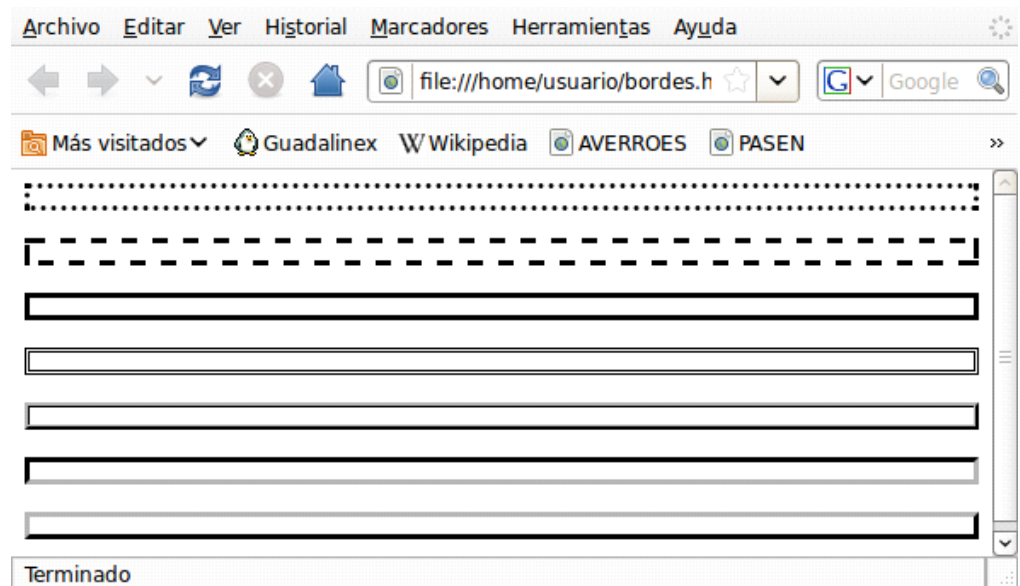
solid

double

ridge

inset

outset



Margin Los mágenes: distancia entre los bordes y el relleno.

Ej: `table{margin: 20px 20px 20px 20px}` o `table{margin: 20px}`

Podemos referirnos a cada margen:

`margin-top` (superior), `margin-right` (derecho), `margin-bottom` (inferior), `margin-left` (Izqdo).

Padding: distancia entre los bordes y los elementos internos.

Ej: `padding{20px 40px 20px 40px}`

También podemos especificar: `padding-top`, `padding-right`, `padding-bottom`, `padding-left`.

4. Definir estilos utilizando clases.

Definiciones de estilo que se pueden utilizar repetidas veces. Se pueden declarar de dos formas:

- Etiqueta <style>
- Fichero externo.

Declaración de sintaxis: `.nombredelaclase{atributo:valor;atributo2:valor2;...}`

Una vez declarada la podemos usar con cualquier etiqueta html.

Uso sintaxis: <Etiquetas class="nombredelaclase">

Ejemplos:

Definición de clase	Aplicación
<code>.miclase{color:blue;}</code>	<code><p class="miclase">Parrafo con clase</p></code>
Definición de clase	Aplicación
<code>body, td,p{ background-color:#000000;color:#ffffff;} .inverso { background-color:#ffffff; color:#000000; }</code>	<code><body> <p>Parrafo normal</p> <p class="inverso">Parrafo invertido</p> <table> <tr> <td class="inverso">inverso</td> <td>normal</td> </tr> </table> </body></code>

Uso de id

El atributo "id" es similar a [class](#). El atributo ID sirve para **identificar un elemento**, y **solo debe usarse para idetificar un único elemento**, si queremos que pertenezcan a varios elementos entonces debemos usar las [clases](#). Se podría decir que las clases son para grupos. Con el atributo ID también se puede dirigir al visitante a un punto de la página (`texto`).

Un uso incorrecto de las clases sería definir las clases para un único elemento, porque se puede utilizar #ID para este propósito. Por ejemplo, cuando debemos identificar una única cabecera (`class="header"`).

Un uso incorrecto de los #ID's es definir los estilos de cada elemento de forma individual en lugar de agrupar todos los estilos que sean comunes y aplicarlos por [clases](#).

Ejemplo:

Aplicación	Definición
<code><div id="elegante"></code>	<code>#elegante{</code>

Soy un simple texto en un div. <code></div></code>	<code>width: 90%;</code> <code>padding: 10px;</code> <code>border: 2px dashed #CCC;</code> <code>background: white;</code> <code>}</code>
---	---

¡ATENCIÓN! Si te fijas, hemos puesto una **almohadilla** (#elegante) **en vez de un punto** (.elegante) como hacíamos con el atributo class, esta es otra de las diferencias respecto al otro atributo.

5. Estilos en los enlaces:

Hacer enlaces quitándoles el subrayado o varios enlaces de diferente color en una misma página.

Se pueden declarar de dos formas:

- Etiqueta <style>
- Fichero externo

Enlaces normales	A:Link {atributos}
Enlaces visitados	A:visited{atributos}
Enlaces Activos	A:active {atributos}
Enlaces hover	A:hover {atributos}

Atributos: sin-subrayado text-decoration:none color:tucolor

Ejemplos:

HTML	CSS
<pre><html> <head> <title>Ejemplo de utilización de enlaces</title> </head> <body> enlace nuevo
 enlace visitado Pulsa y ponte encima para ver los colores. </body> </html></pre>	<pre>A.nuevo:link{text-decoration:none;color:#0000cc;} A:.nuevo:visited{text-decoration:none;color:#ffcc33;} A.nuevo:active{text-decoration:none;color:#ff0000;} A.nuevo:hover {text-decoration:underline;color:#999999;font-weght:bold;} a.menus:link { text-decoration:none; color: #ffffff; background: #990000; } a.menus:visited {text-decoration:none; color:#cccccc} a.menus:active {text-decoration:none; color: #003399; background: #993333; } a.menus:hover {text-decoration:underline; color: #003399; background: #993333; }</pre>

6. Css con listas

Por defecto, los navegadores muestran los elementos de las listas no ordenadas con una viñeta formada por un pequeño círculo de color negro. Los elementos de las listas ordenadas se muestran por defecto con la numeración decimal utilizada en la mayoría de países.

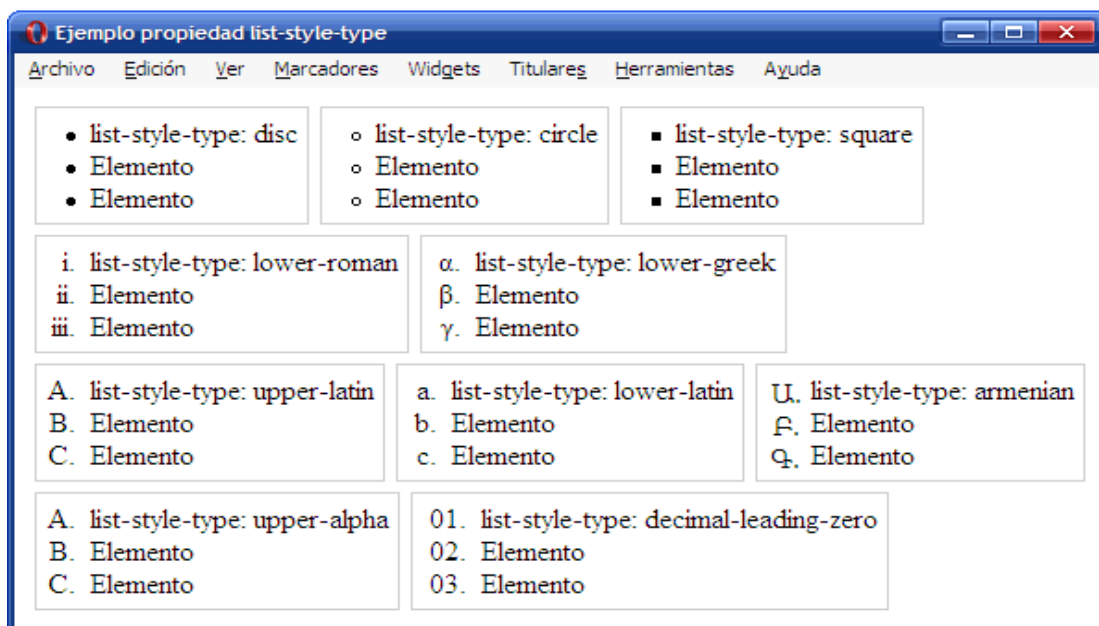
No obstante, CSS define varias propiedades para controlar el tipo de viñeta que muestran las listas, además de poder controlar la posición de la propia viñeta. La propiedad básica es la que controla el tipo de viñeta que se muestra y que se denomina `list-style-type`.

Propiedad	<code>list-style-type</code>
Valores	<code>disc</code> <code>circle</code> <code>square</code> <code>decimal</code> <code>decimal-leading-zero</code> <code>lower-roman</code> <code>upper-roman</code> <code>lower-greek</code> <code>lower-latin</code> <code>upper-latin</code> <code>armenian</code> <code>georgian</code> <code>lower-alpha</code> <code>upper-alpha</code> <code>none</code> inherit

El resto de valores de la propiedad `list-style-type` se dividen en tres tipos: gráficos, numéricos y alfabéticos.

- Los valores gráficos son `disc`, `circle` y `square` y muestran como viñeta un círculo relleno, un círculo vacío y un cuadrado relleno respectivamente.
- Los valores numéricos están formados por `decimal`, `decimal-leading-zero`, `lower-roman`, `upper-roman`, `armenian` y `georgian`.
- Por último, los valores alfanuméricos se controlan mediante `lower-latin`, `lower-alpha`, `upper-latin`, `upper-alpha` y `lower-greek`.

La siguiente imagen muestra algunos de los valores definidos por la propiedad `list-style-type`:



Ejemplo de propiedad list-style-type

Ejemplos:

```
<ul style="list-style-type: square">
```

```
<li>list-style-type: square</li>
```

```
<li>Elemento</li>
```

```
<li>Elemento</li>
```

```
</ul>
```

```
<ol style="list-style-type: lower-roman">
```

```
<li>list-style-type: lower-roman</li>
```

```
<li>Elemento</li>
```

```
<li>Elemento</li>
```

```
</ol>
```

```
<ol style="list-style-type: decimal-leading-zero; padding-left: 2em;">
```

```
<li>list-style-type: decimal-leading-zero</li>
```

```
<li>Elemento</li>
```

```
<li>Elemento</li>
```

```
</ol>
```

Utilizando las propiedades anteriores (`list-style-type` y `list-style-position`), se puede seleccionar el tipo de viñeta y su posición, pero no es posible personalizar algunas de sus características básicas como su color y tamaño.

Cuando se requiere personalizar el aspecto de las viñetas, se debe emplear la propiedad `list-style-image`, que permite mostrar una imagen propia en vez de una viñeta automática.

Propiedad	list-style-image
Valores	url none inherit

Las imágenes personalizadas se indican mediante la URL de la imagen. Si no se encuentra la imagen o no se puede cargar, se muestra la viñeta automática correspondiente (salvo que explícitamente se haya eliminado mediante la propiedad `list-style-type`).

La siguiente imagen muestra el uso de la propiedad `list-style-image` mediante tres ejemplos sencillos de listas con viñetas personalizadas:



Ejemplo de propiedad list-style-image

Ejemplo par que en toda la página la lista tenga el mismo estilo:

```
ul { list-style-image: url("cuadrado_rojo.gif"); }
```

Ejemplo con clases para cambiar el estilo de lista en la página:

html	CSS
<pre><ul class="vinieta1"> Elemento lista 1 <ul class="vinieta2"> Elemento lista 2 <ul class="vinieta3"> Elemento lista 3 </pre>	<pre>.vinieta1{ list-style-image: url('list1.gif'); } .vinieta2{ list-style-image: url('list2.png'); } .vinieta3{ list-style-image: url('list3.png'); }</pre>

7. CSS con imágenes

La propiedad `background-image` se emplea para establecer la imagen de fondo de cada elemento. La imagen se muestra en la zona que ocupan el contenido y el [relleno](#) del elemento, justo hasta su [borde](#).

Para indicar la imagen que se muestra como fondo de un elemento, se utiliza una URL. Los navegadores no restringen el tipo de URL que se pueden utilizar, por lo que pueden ser URL externas/internas y URL relativas/absolutas.

La URL de la imagen que se quiere mostrar se incluye dentro de los paréntesis de la palabra `url()`. De forma opcional, el valor de la URL se puede encerrar entre comillas simples o entre comillas dobles.

Ejemplos:

```
body {background-image:url(http://www.librosweb.es/website/css/images/logo.gif);}
```

```
h1, h2, h3 { background-image: url("imagenes/fondo.png");}
```

Si la imagen de fondo es más grande que el sitio disponible para mostrarla, el navegador sólo muestra la parte visible que resulta de colocar la esquina superior izquierda de la imagen en la esquina superior izquierda del espacio que ocupa el elemento:

```
body { background-image: url("images/imagen.png"); }
```

Por el contrario, si la imagen es más pequeña que el sitio disponible, el navegador repite la misma imagen en todas las direcciones hasta cubrir completamente el espacio que ocupa el elemento:

```
body { background-image: url("images/imagen_pequena.png"); }
```

Sobre un mismo elemento se puede establecer de forma simultánea una imagen de fondo y un [color de fondo](#). Este comportamiento es habitual en algunos elementos, como por ejemplo el fondo de la propia página web.

Si un mismo elemento tiene definida una imagen y un color de fondo, la imagen tiene prioridad. El color de fondo se muestra en las zonas transparentes de la imagen y en las zonas que no están cubiertas por la imagen.

Se recomienda establecer un color de fondo siempre que se utilice una imagen de fondo. Como las imágenes se indican mediante URL y el navegador las tiene que descargar, es posible que se produzcan errores y la descarga de la imagen no se pueda realizar.

Por lo tanto, si se establece un color de fondo junto con la imagen de fondo y se produce un error que impide mostrar la imagen, el navegador mostrará el color de fondo establecido. Por este motivo, el color de fondo debe ser similar a la imagen utilizada.

El estándar CSS no permite modificar el tamaño con el que se muestra la imagen de fondo. Sin embargo, es posible modificar opciones como si la imagen se repite o no ([background-repeat](#)), si la imagen permanece fija o no ([background-attachment](#)) y modificar su posición por defecto ([background-position](#)).

Por último, no se pueden establecer dos o más imágenes diferentes sobre un mismo elemento. Si se quieren mostrar dos imágenes diferentes como fondo de una página web, se puede establecer la primera imagen sobre el selector `html` y la segunda imagen sobre el selector `body`:

Ejemplos:

```
html { background-image: url("imagenes/primera_imagen.png"); }
```

```
body { background-image: url("imagenes/segunda_imagen.png"); }
```

Para que este último ejemplo sea realmente útil, es imprescindible utilizar al menos la propiedad [background-repeat](#).

8. CSS con tablas

Vamos a ver un uso de las CSS que puede ser muy interesante para hacer tablas con títulos que tengan un cierto estilo. Lo bueno de las CSS es que podemos definir el estilo una vez y se puede utilizar en múltiples elementos de la página. Con todo ello vamos a tratar la definición de un estilo y la aplicación para hacer distintos tipos de decoración de tablas.

Definición de los estilos

Vamos a definir un estilo por defecto para las celdas (etiqueta <td>) y un par de clases, la primera para las tablas y la segunda para las celdas titular (las que tienen color de fondo).

Ejemplo:

HTML	CSS
<pre><table width=280 height=18 cellpadding=2 cellspacing=2 class="estilotabla"> <tr> <td class="estilocelda">Título de sección</td> </tr> <tr> <td>Este es un texto de lo que podría ser la parte de abajo de la tabla con el contenido relacionado con este título.</td> </tr> </table></pre>	<pre>td { font-family:verdana,arial; font-size:8pt; } .estilotabla{ background-color:ffffff; border-style:solid; border-color:666666; border-width:1px; } .estilocelda{ background-color:ddeeff; color:333333; font-weight:bold; font-size:10pt; }</pre>

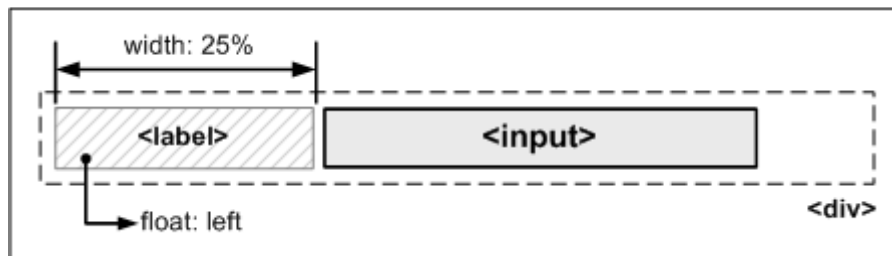
Los atributos de estilos se pueden conocer en el manual de CSS. En este caso, para la clase **estilotabla** estamos definiendo un color de fondo, un borde sólido, un color del borde y un ancho del borde, por ese orden. Para la clase **estilocelda** estamos definiendo un color de fondo, un color del texto, un grosor de la fuente y un tamaño de la fuente.

Lo único que tiene de especial es que utiliza las clases que se han definido previamente. En la etiqueta <table> se utiliza la clase **estilotabla** y en la etiqueta <td> que queremos que sea el titular se utiliza la clase **estilocelda**. La otra celda tendrá el estilo definido para todas las celdas en general.

En este caso utilizamos dos tablas para hacer el efecto. La tabla de arriba tiene un borde y la de abajo no. Para ello aplicamos la clase definida para la tabla y la celda solamente en la tabla de arriba, así el borde definido en la declaración de estilos sólo afecta a la tabla de arriba.

9. CSS CON FORMULARIOS

Para mostrar un formulario tal y como aparece en la imagen anterior no es necesario crear una tabla y controlar la anchura de sus columnas para conseguir una alineación perfecta. Sin embargo, sí que es necesario añadir un nuevo elemento (por ejemplo un `<div>`) que encierre a cada uno de los campos del formulario (`<label>` y `<input>`). El esquema de la solución propuesta es el siguiente:



Esquema de la técnica de alineación de etiquetas label y campos de formulario

Por tanto, en el código HTML del formulario anterior se añaden los elementos `<div>`:

```
<form>
  <fieldset>
    <legend>Alta en el servicio</legend>
    <div>
      <label for="nombre">Nombre</label>
      <input type="text" id="nombre" />
    </div>
    <div>
      <label for="apellidos">Apellidos</label>
      <input type="text" id="apellidos" size="35" />
    </div>
    ...
  </fieldset>
</form>
```

Y en el código CSS se añaden las reglas necesarias para alinear los campos del formulario:

```
div {
  margin: .4em 0;
}
div label {
  width: 25%;
  float: left;
}
```