

ISSD

—*Asc*—
Analista de
Sistemas

TABLAS Y
ESTILOS

I2

Informática 2

Módulo didáctico - 2015



Unidad 1 y 2



Clase 4



HTML y CSS3



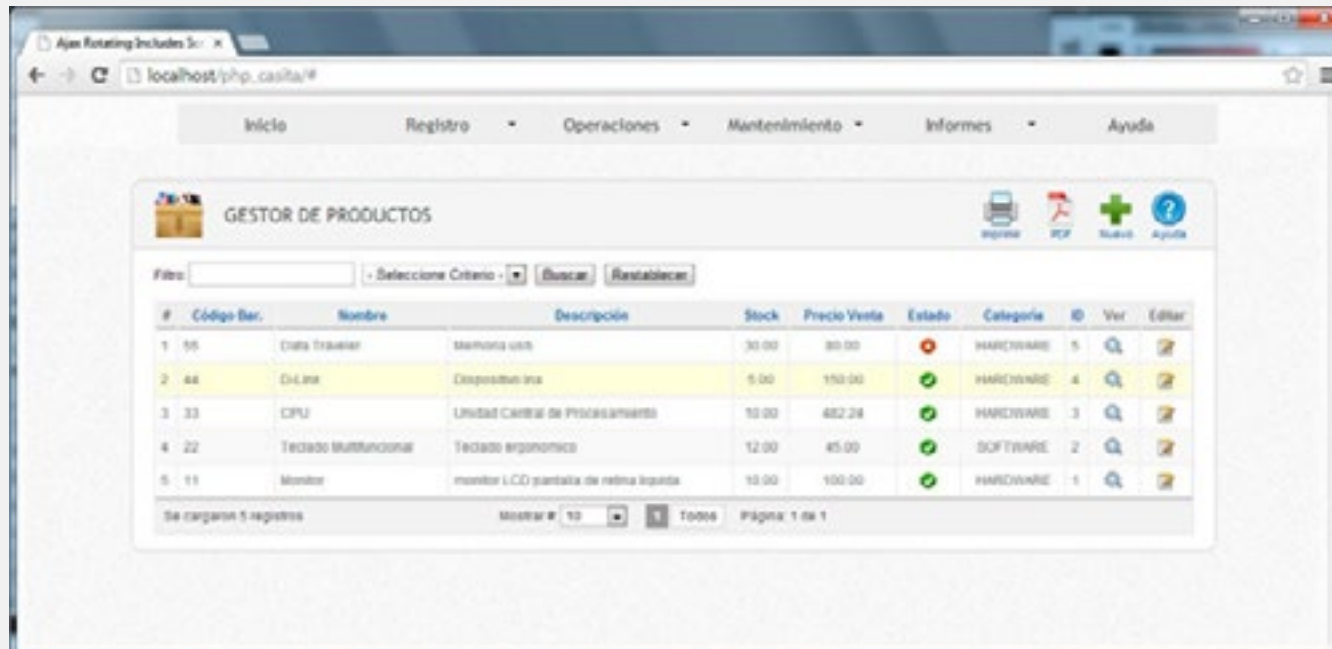
Al finalizar esta clase ya dominarás dos temas de gran importancia:

| Uno está referido a las tablas, es decir, a los datos tabulados para una mejor lectura y comprensión del usuario.

| El otro remite a los estilos CSS3.

Un cliente te solicita que mediante una página web pueda ver las transacciones de su negocio en el último mes. ¿Cómo colocarías los datos en filas y en columnas?

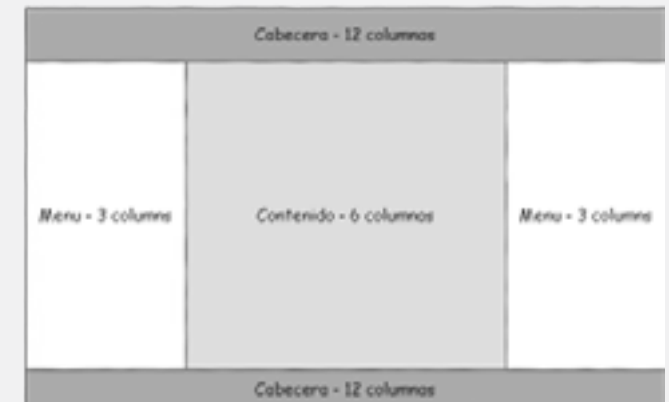
Muchos programadores al comienzo se valían del artilugio de las filas y las columnas para maquetar un sitio. Pero se dieron cuenta que no tenían tanto control sobre los datos y es por eso que implementaron el uso de contenedores, que pasaremos a describir en la próxima clase. ¡No te la pierdas!



A screenshot of a web application titled "GESTOR DE PRODUCTOS". The interface includes a navigation menu with "Inicio", "Registro", "Operaciones", "Mantenimiento", "Informes", and "Ayuda". Below the menu is a search bar with a "Filtro" input, a "Selección Criterio" dropdown, and "Buscar" and "Restablecer" buttons. The main content area displays a table of products with the following data:

#	Código Bar.	Nombre	Descripción	Stock	Precio Venta	Estado	Categoría	Ver	Editar
1	55	Crata Traveler	Memoria USB	30.00	80.00	✖	HARDWARE	5	🔍 🗑️
2	44	D-Link	Dispositivo Ina	5.00	150.00	✔	HARDWARE	4	🔍 🗑️
3	33	CPU	Unidad Central de Procesamiento	10.00	482.24	✔	HARDWARE	3	🔍 🗑️
4	22	Teclado Multifuncional	Teclado ergonómico	12.00	45.00	✔	SOFTWARE	2	🔍 🗑️
5	11	Monitor	monitor LCD pantalla de retina ligera	10.00	100.00	✔	HARDWARE	1	🔍 🗑️

At the bottom of the table, it says "Se cargaron 5 registros", "Mostrar 10", "10000", and "Página 1 de 1".





Tablas y Estilos

¡Bienvenido!

Durante el transcurso de esta clase aprenderás a implementar tablas para tabular información y entenderás de qué manera se definen los estilos y cuál es la ventaja de su uso.

Además vas a comprender como HTML implementa CSS3 para lograr sitios de calidad profesional y como se establecen las distintas formas de definir estilos, las diferentes unidades de medidas que se definen en CSS3. También como se realiza la lectura de varias propiedades a la vez.

Por último comprenderás el concepto de selectores y su uso y aprenderás los estilos básicos mediante ejercicios comentados paso a paso.

La idea para comenzar con esta clase es armar una tabla y aplicarle estilos desde la zona `<style>`. De ahora en más todos los ejercicios y ejemplos tendrán definidos estilos, en sus diferentes variantes.

Te deseo lo mejor para esta clase y recordá lo importante que es desarrollar los ejercicios o ejemplos. Mientras más practiques más dominio tendrás del lenguaje.

¡Mucha Suerte!



Desempeño de Exploración

En este punto te proponemos debatir sobre los conceptos expuestos en clases y también que busques ejemplos y relaciones con otras materias. La intención es que comiences a perfilarte todas las metas de comprensión y tópicos de la materia.

Desempeño de Investigación

También realizaremos investigaciones sobre los temas expuestos en clase en forma grupal para debatir en las clases subsiguientes. Armaremos galerías de imágenes, trabajaremos con herramientas auxiliares como editores fotográficos, paleta de colores, fuentes etc.

Sobre las Tablas

1

<table>

Artículo	Cantidad
Zapatillas	1.500
Gorras	12.200
Pantalones	3.800
Camisetas	7.100

<td>

<tr>

Tablas básicas

Una tabla es un conjunto de celdas organizadas dentro de las cuales podemos alojar distintos contenidos.

En un principio nos podría parecer que las tablas son raramente útiles y que pueden ser utilizadas principalmente para listar datos como agendas, resultados y otra información de una forma organizada. ¡Nada más lejos de la realidad!

Hoy, gran parte de los diseñadores de páginas basan su maquetación en este tipo de artilugios. En efecto, una tabla nos

permite organizar y distribuir los espacios de la manera más óptima. Nos puede ayudar a generar texto en columnas como los periódicos, prefijar los tamaños ocupados por distintas secciones de la página o poner de una manera sencilla un pie de foto a una imagen.

Puede que en un principio nos resulte un poco complicado trabajar con estas estructuras pero, si deseamos crear una página de calidad, tarde o temprano tendremos que vernoslas con ellas y nos daremos cuenta de las posibilidades que nos ofrecen. Para empezar, nada más sencillo que hacerlo por

el comienzo: las tablas son definidas por las etiquetas <table> y </table>. Dentro de estas dos etiquetas colocaremos todas las otras etiquetas, textos e imágenes que darán forma y contenido a la tabla.

Las tablas son descritas por líneas de izquierda a derecha. Cada una de estas líneas es definida por otra etiqueta y su cierre: <tr> y </tr>. Asimismo, dentro de cada línea, habrá diferentes celdas. Cada una de estas celdas será definida por otro par de etiquetas: <td> y </td>. Dentro de estas etiquetas será donde coloquemos nuestro contenido (1).

align	Justifica el texto de la celda del mismo modo que si fuese el de un párrafo.
valign	Podemos elegir si queremos que el texto aparezca arriba (top), en el centro (middle) o abajo (bottom) de la celda.
bgcolor	Da color a la celda o línea elegida.
bordercolor	Define el color del borde.

background	Nos permite colocar un fondo para la celda a partir de un enlace a una imagen.
height	Define la altura de la celda en pixels o porcentaje.
width	Define la anchura de la celda en pixels o porcentaje.
colspan	Expande una celda horizontalmente.
rowspan	Expande una celda verticalmente.

Veamos cómo podemos construir una tabla:

```
<table>
<tr>
<td>Directivas</td><td>Atributos?</td>
<td>Directiva nueva?</td>
</tr>
<tr>
<td>aside</td><td>no</td><td>si</td>
</tr>
<tr>
<td>p</td><td>no</td><td>no</td>
</tr>
<tr>
<td>img</td><td>src</td><td>no</td>
</tr>
<tr>
<td>a</td><td>href</td><td>no</td>
</tr>
</table>
```

Atributos para filas y celdas

Observemos a continuación algunos atributos útiles para la construcción de nuestras tablas. Empecemos viendo atributos que nos permiten modificar una celda en concreto o toda una línea (1).

Otros atributos que pueden ser únicamente asignados a una celda y no al conjunto de celdas de una línea son los de la imagen (2).

Estos últimos cuatro atributos descriptos son de gran utilidad. Concretamente, *height* y *width* nos ayudan a definir las dimensiones de nuestras celdas de una forma absoluta (en pixels o puntos de pantalla) o de una forma relativa, es decir, por porcentajes referidos al

tamaño total de la tabla.

<td width="80"> : Dará una anchura de 80 pixels a la celda.

<td width="80%"> : Dará una anchura a la celda del 80% de la anchura de la tabla.

Los atributos *rowspan* y *colspan* son también utilizados frecuentemente. Gracias a ellos es posible expandir celdas fusionando éstas con sus vecinas. El valor que pueden tomar estas etiquetas es numérico. El número representa la cantidad de celdas fusionadas.

Por lo tanto diremos que:

| Rowspan: indica el número de filas que ocupará la celda. Por defecto ocupa una sola fila.

Atributos para filas y celdas

Observemos a continuación algunos atributos útiles para la construcción de nuestras tablas. Empecemos viendo atributos que nos permiten modificar una celda en concreto o toda una línea (1).

Otros atributos que pueden ser únicamente asignados a una celda y no al conjunto de celdas de una línea son los de la imagen (2).

Estos últimos cuatro atributos descriptos son de gran utilidad. Concretamente, *height* y *width* nos ayudan a definir las dimensiones de nuestras celdas de una forma absoluta (en pixels o puntos de pantalla) o de una forma relativa, es decir, por porcentajes referidos al tamaño total de la tabla.

<td width="80"> : Dará una anchura de 80 pixels a la celda.

<td width="80%"> : Dará una anchura a la celda del 80% de la anchura de la tabla.

Los atributos `rowspan` y `colspan` son también

Países europeos	España	Madrid
	Francia	Paris
	Reino Unido	Londres
Países americanos	EEUU	Washington
	Canada	Toronto
	México	México

Código	Resultado						
<code><TABLE BORDER=1></code> <code><CAPTION></code> Ejemplo de tablas <code></CAPTION></code> ... <code></TABLE></code>	Ejemplo de tablas <table><tr><td>1,1 y 1,2</td><td>1,3</td></tr><tr><td>2,1 y 3,1</td><td>2,2 2,3</td></tr><tr><td></td><td>3,2 3,3</td></tr></table>	1,1 y 1,2	1,3	2,1 y 3,1	2,2 2,3		3,2 3,3
1,1 y 1,2	1,3						
2,1 y 3,1	2,2 2,3						
	3,2 3,3						

```
<table width="80%" border="1"
cellspacing="0" cellpadding="0">
<tr>
  <td rowspan="3">
    <p>Países Europeos</p>
  </td>
  <td><p>España</p></td>
  <td><p>Madrid</p></td>
</tr>
<tr>
  <td><p>Canada</p></td>
  <td><p>Toronto</p></td>
  <td><p>Francia</p></td>
  <td><p>Paris</p></td>
</tr>
<tr>
  <td><p>Reino Unido</p></td>
  <td><p>Londres</p></td>
</tr>
<tr>
  <td rowspan="3">
    <p>Países Americanos</p>
  </td>
  <td><p>EEUU</p></td>
  <td><p>Washington</p></td>
  <td><p>México</p></td>
  <td><p>México</p></td>
</tr>
</table>
```


utilizados frecuentemente. Gracias a ellos es posible expandir celdas fusionando éstas con sus vecinas. El valor que pueden tomar estas etiquetas es numérico. El número representa la cantidad de celdas fusionadas.

Por lo tanto diremos que:

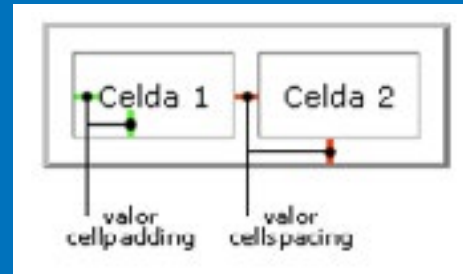
| **Rowspan:** indica el número de filas que ocupará la celda. Por defecto ocupa una sola fila.

| **Colspan:** indica el número de columnas que ocupará la celda. Por defecto ocupa una sola columna.

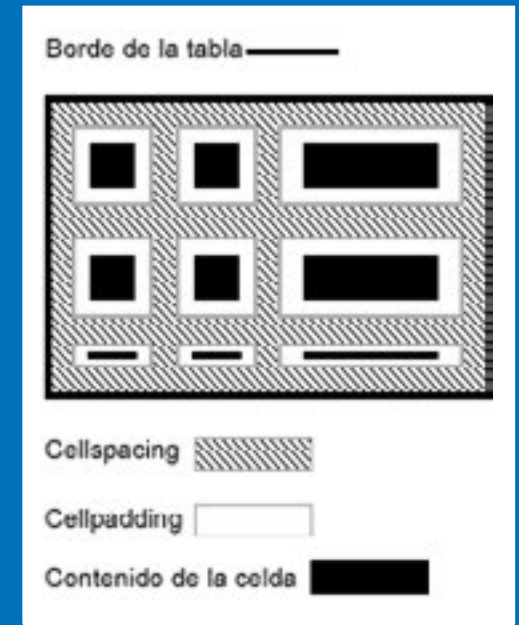
De esta forma si ponemos `<td colspan=2>`, quiere decir que la celda actual se extiende en el ancho de dos celdas. Algo parecido ocurre si ponemos `<td rowspan=3>`, la celda ocupará el alto de 3 celdas normales. Veamos un ejemplo (1).

Si queremos que la tabla tenga título, lo haremos por medio de la etiqueta `CAPTION`. Para ver cómo funciona, vamos a incluirlo en la declaración de la tabla anterior tal como lo muestra la imagen (2). Además, para establecer los márgenes de las celdas utilizaremos los siguientes atributos:

1



2



3

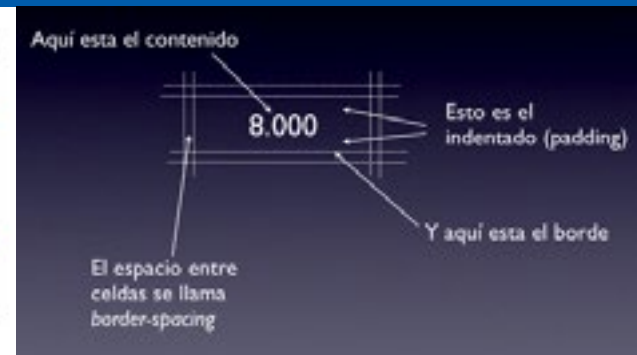
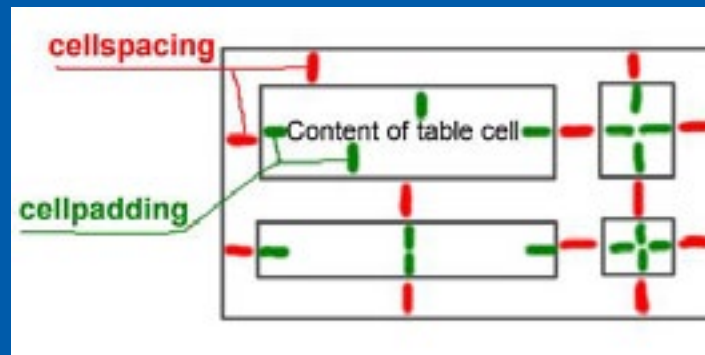


Tabla con 10 pixels de margen
(CELLPADDING=10)

Fila1,Columna1	Fila1,Columna2
Fila2,Columna1	Fila2,Columna2

Tabla con márgenes nulos

Fila1,Columna1	Fila1,Columna2
Fila2,Columna1	Fila2,Columna2

```
<TABLE CELLPADDING=8 CELLSPACING=0
BACKGROUND="foto.gif">
<TD BACKGROUND="images/ex.gif" WIDTH="60"
HEIGHT="35">
```

| Cellspacing: especifica cuánto espacio debería dejar el agente de usuario entre el lado izquierdo de la tabla y el lado izquierdo de la columna que está más a la izquierda, entre la parte superior de la tabla y el lado superior de la fila que está más arriba, y lo mismo para los lados derecho e inferior. El atributo también especifica la cantidad de espacio entre celdas **(1)**.

| Cellpadding : Este atributo especifica la

cantidad de espacio entre el borde de la celda y sus contenidos. Si el valor de este atributo es una longitud en píxeles, los cuatro bordes deberían estar a esta distancia de los contenidos. Si el valor del atributo es una longitud porcentual, los bordes superior e inferior deberían estar igualmente separados del contenido según un porcentaje del espacio vertical disponible, y los bordes izquierdo y derecho deberían estar igualmente separados de los contenidos según un porcentaje del

espacio horizontal disponible **(2)**.

Estos dos atributos controlan el espacio entre y dentro de las celdas. La imagen **(3)** explica cuál es la relación entre ellos.

En el ejemplo de la página siguiente, el atributo *cellspacing* especifica que las celdas deberían estar separadas entre sí y hasta el marco de la tabla por veinte píxeles. El atributo *cellpadding* especifica que el margen superior

Bordes y colores

Para mejorar el aspecto de nuestras tablas, lo primero que queremos hacer es jugar con la anchura del borde. Para ello se ha definido el parámetro *border* de la etiqueta `<table>`, que define la anchura de todos los bordes de la tabla en pixels. Si se especifica con valor 0, el borde no se visualiza.

El color del borde lo controla el parámetro *bordercolor*, que se aplica al tag `<td>`. Este parámetro toma valores hexadecimales o nombres de colores predefinidos, según el alfabeto inglés. La forma en que se indica un color con código hexadecimal es la misma que se utiliza para otros elementos de una página, por ejemplo, para el color de fondo.

Finalmente, también podemos modificar el color de fondo de cada celda de forma independiente, aplicando el parámetro *bgcolor* al tag `<td>`. Los colores se especifican de igual forma. Con todo esto, podemos mejorar el aspecto del ejemplo anterior de la siguiente forma:

```
<table width="80%" cellspacing="0" cellpadding="5"
border="4">
```

```
<tr>
```

```
<td width="80%" height="70" align="right"
valign="bottom">Fila 1, celda 1</td>
```

```
<td width="20%" height="70">Fila 1, celda 2</td>
```

```
</tr>
```

```
<tr>
```

```
<td width="80%" bgcolor="#FFCCCC">Fila 2, celda
1</td>
```

```
<td width="20%" bordercolor="red">Fila 2, celda
2</td> </tr> </table>
```

Agrupar columnas

El agrupamiento en columnas es más potente que el que acabamos de ver para las filas. Nos permite, entre otras cosas, definir individualmente las propiedades de cada columna, estableciendo por ejemplo la anchura o la alineación. La etiqueta para definir un grupo de columnas es **<colgroup>**, y dentro de esta podemos definir subgrupos de una o más columnas con la etiqueta **<col>**. Los atributos opcionales para estas etiquetas son los mismos que para **<tr>**, y dos propios de estas que son **span** para indicar el número de columnas, y **width** para indicar la anchura de las columnas. Estos grupos de columnas han de definirse inmediatamente después de la etiqueta **<table>** y antes de definir las filas.

```
<HTML>
<HEAD><TITLE>TABLAS ...</title></HEAD>
<BODY>
<H2 align="center"> TABLAS: (Básicas - Agrupadas)</H2>
<H3 align="center">Ejemplo de tablas básicas con algunos de los elementos y parámetros más utilizados</H3>
<TABLE BORDER>
  <TR>    <TD><TH ROWSPAN=2></TH>
          <TH COLSPAN=2>Valores</TH></TD>
        </TR>
  <TR>    <TD><TH>Normal</TH><TH>Lectura</TH></TD>
        </TR>
  <TR>    <TH ROWSPAN=2>Tipos de Medidores</TH>
          <TH>Familiar</TH><TD>190</TD><TD>385</TD>
        </TR>
  <TR>    <TH>Comercial</TH><TD>400</TD><TD>8860</TD>
        </TR>
</TABLE>
<H3 align="center">Ejemplo de tablas agrupando filas y columnas con algunos de los elementos y parámetros más utilizados</H3>
<TABLE width="100%" cellpadding="5" cellspacing="10" rules="groups" bordercolor="red">
  <CAPTION>Listado de Empresas</CAPTION>
  <COLGROUP span=2 align="left" width="40%" >
  </COLGROUP>
  <COL align="right" width="20%">
  <THEAD align="left">
    <TR>
      <TH>Empresa
      <TH>Contacto
      <TH>Teléfono
    </THEAD>
  <TBODY align="left" bgcolor="#358391">
    <TR>
      <TD>Televisión
      <TD>Productora
      <TD>999999999
    </TR>
    <TR>
      <TD>Internet
      <TD>Anónimo
      <TD>124597897
    </TBODY>
  </TABLE>

</BODY>
</HTML>
```

También podemos encontrar tablas avanzadas como la que se observa en la imagen (1).

Otros atributos

Pero eso no es todo, también existen los atributos:

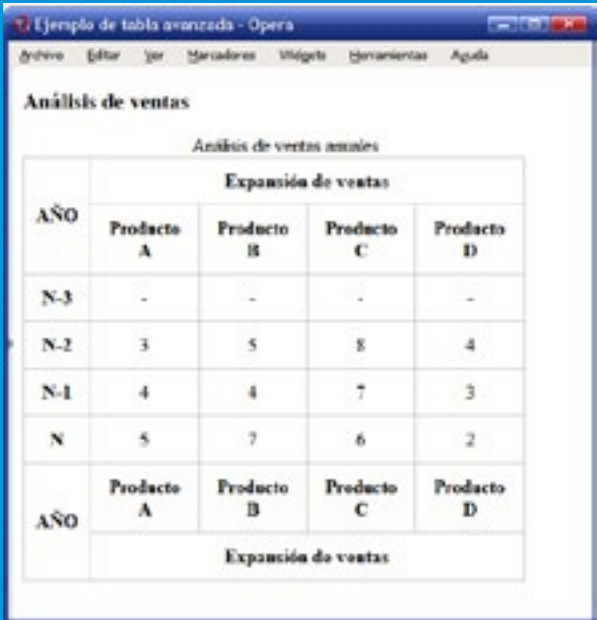
| **summary:** resumen del contenido de la tabla para sistemas acústicos.

| **Frame (marco):** a qué lados de la tabla deben ser añadidos bordes:

- frame="void" (void = nada)
- frame="above" (above = superior)
- frame="below" (below = inferior)
- frame="hsides" (horizontal sides = lados horizontales)
- frame="vsides" (vertical sides = lados verticales)
- frame="lhs" (left hand side = lado izquierdo)
- frame="rhs" (right hand side = lado derecho)

| **Rules (reglas):** permite especificar en la etiqueta de apertura <table> las líneas horizontales o reglas para las líneas de la cuadrícula:

- rules="none" (none = ningún) no son visualizadas las líneas de la cuadrícula, el borde exterior de la tabla es sin embargo visualizado.
- rules="rows" (rows = filas) son visualizadas las líneas entre todos las filas de la tabla, pero no las líneas entre las columnas de la tabla.
- rules="cols" (cols = columnas) son visualizadas todas las líneas entre todas las columnas de la tabla, sin embargo no las líneas entre las filas.
- rules="groups" (groups = grupos) son visualizadas todas las líneas entre la cabeza, cuerpo y pie de una tabla
- rules="all" (all = todo) son visualizadas todas las líneas entre todas las celdas de la tabla. Es el valor preajustado por lo que equivale a no utilizar este atributo.



Ejemplo de tabla avanzada - Opera

Análisis de ventas

Análisis de ventas anuales

AÑO	Expansión de ventas			
	Producto A	Producto B	Producto C	Producto D
N-3	-	-	-	-
N-2	3	5	8	4
N-1	4	4	7	3
N	5	7	6	2
AÑO	Producto A	Producto B	Producto C	Producto D
	Expansión de ventas			

Tablas anidadas

Muy útil también es el uso de tablas anidadas. De la misma forma que podíamos incluir listas dentro de otras listas, las tablas pueden ser incluidas dentro de otras. Así, podemos incluir una tabla dentro de la celda de otra. El modo de funcionamiento sigue siendo el mismo aunque la situación puede complicarse si el número de tablas embebidas dentro de otras es elevado. Podemos observar ejemplos de esto en las imágenes 1, 2 y 3.

El siguiente código corresponde a la imagen 1

1

Celda de la tabla principal	Tabla anidada, celda 1	Tabla anidada, celda 2
	Tabla anidada, celda 3	Tabla anidada, celda 4

2

```
<HTML>
<BODY>
<table border="1">
<tr>
<td>Celda1.1</td>
<td>
Celda1.2
<table align="center" border="1">
<tr>
<td>a</td>
<td>b</td>
</tr>
</table>
</td>
</tr>
<tr>
<td>Celda2.1</td>
<td>Celda2.2</td>
</tr>
</table>
</BODY>
</HTML>
```

Lo que escribimos en el editor de texto (Wordpad) y guardamos como fichero TablasAnidadas.html

Lo que vemos en el navegador (Internet Explorer) al abrir el fichero TablasAnidadas.html

3

Logotipo	Fecha		Tabla 1
	Menú		Tabla 2
Secciones del Sitio	Foto	Contenido 1	Tabla 4
			Tabla 3
	Contenido 2		
Pie de Página			Tabla 5

Estilos CSS

¿Qué es CSS?

CSS son las siglas de Cascading Style Sheets (en español, hojas de estilo en cascada), y es un lenguaje que define el estilo de la presentación de páginas de Internet. Se usa comúnmente en conjunto con HTML, donde la parte de HTML define la estructura de la página, y la parte de CSS define la presentación de la página, esto es, aspectos como colores, tipos de letra, márgenes, dimensiones, entre otros.

¿Para qué sirve?

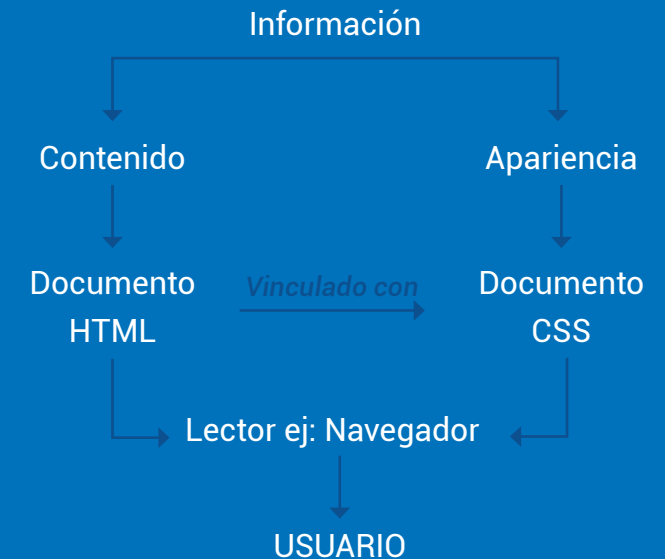
CSS se utiliza para dar formato a documentos HTML y XML, separando el contenido de la presentación. CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento.

¿Qué beneficios ofrece?

CSS fue toda una revolución en el mundo del diseño web. Entre sus beneficios concretos encontramos:

- | Control de la presentación de muchos documentos desde una única hoja de estilo.
- | Control más preciso de la presentación.
- | Aplicación de diferentes presentaciones a diferentes tipos de medios (pantalla, impresión, etc.).

Navegador	Motor	CSS 1.0	CSS 2.1	CSS 3.0
Internet Explorer	Trident	Completo desde la versión 6.0	Completo desde la versión 8.0	Prácticamente nulo
Firefox	Gecko	Completo	Casi completo	Selectores, pseudo-clases y algunas propiedades
Safari	WebKit	Completo	Casi completo	Todos los selectores, pseudo-clases y muchas propiedades
Opera	Presto	Completo	Casi completo	
Google Chrome	WebKit	Completo	Casi completo	



Soporte de CSS en los navegadores

El trabajo del diseñador web siempre está limitado por las posibilidades de los navegadores que utilizan los usuarios para acceder a sus páginas. Por este motivo es imprescindible conocer el soporte de CSS en cada uno de los navegadores más utilizados del mercado.

Internamente los navegadores están divididos en varios componentes. La parte del navegador que se encarga de interpretar el código HTML y CSS para mostrar las páginas se denomina motor. Desde el punto de vista del diseñador CSS, la versión de un motor es mucho más importante que la versión del propio navegador.

La tabla (1) muestra el soporte de CSS 1, CSS 2.1 y CSS 3 de los cinco navegadores más utilizados por los usuarios.

Comenzando a Definir Estilos

Una vez que aprendimos a crear la estructura en HTML5, haremos hincapié en el formato del sitio, para ello comenzaremos a estudiar CSS en su versión 3. A continuación te indico tres conceptos que debes manejar de ahora en más:

Atributos: Son las palabras que usaremos para indicar cual estilo queremos modificar, por ejemplo, si queremos cambiar el tipo de

letra, usamos el atributo *font*, si es el fondo, el atributo *background*, etc.

| **Valores:** Son para definir cómo vamos a modificar el atributo, o la propiedad que le daremos. Por ejemplo, si queremos que un tipo de letra sea rojo, usamos el atributo *font* y el valor *red*.

| **Selectores:** Se usan para definir sobre cuales elementos HTML vamos a aplicar los estilos, si queremos definir un estilo para toda la página, debemos usar el selector *body* que se refiere a la etiqueta `<body>` del documento HTML.

Hay tres tipos de selectores:

| Los selectores de etiquetas HTML, se utilizan escribiendo el nombre de la etiqueta a la que le aplicaremos el estilo.

| Los selectores de identificador, se usan para aplicar estilos solo a las etiquetas identificadas con un nombre.

| El tercer selector es el de clase, se escribe en el documento CSS comenzando con un punto

seguido del nombre que le queramos poner a la clase, de esta forma: `.mi_clase`.

Obviamente que veremos cada una de estas formas a lo largo de la materia. Comenzaremos por la parte más fácil incrementando el grado de dificultad. Para ello te pido que desarrolles los ejemplos y ejercicios que se muestran en cada una de las clases, esto te ayudará a tomar destreza y experiencia en el código y sobre todo a la hora de depurar errores.

Formas de implementación

Hay varias formas de implementar código CSS a un documento HTML:

1 | **Hoja de estilos externa:** Esta es la forma más versátil y la única que recomiendo; un fichero externo contiene todo nuestro código CSS de manera que es completamente independiente el código propio de la página web de su código de formato. Esta última metodología es la que dejaremos para el final de todo.

<Selectores>

<Atributos>

<Valores>

<Selectores>

<Atributos>

<Valores>





Tengo una duda...en caso de usar estos tres tipos de implementación para CSS simultáneamente ¿cuál tiene más prioridad?



¡Buena pregunta! el navegador interpreta nuestro código (tanto HTML como CSS) linealmente de izquierda a derecha y de arriba a abajo (tal y como leemos nosotros). Por eso, si lo primero que hacemos es incluir una hoja de estilos externa, luego implementamos una hoja de estilos interna y posteriormente le damos un estilo a una etiqueta HTML concreta, el estilo que visualizaremos será el de la etiqueta HTML por ser el último que el navegador ha leído.

2 | Hoja de estilos interna (la etiqueta style):

Esta forma es a veces utilizada; consiste en incluir el código CSS en la cabecera del sitio (etiquetas head del documento HTML), concretamente dentro de las etiquetas <style> y </style>

```
<html>
  <head>
    <title>Ejemplo</title>
    <style type="text/css">
      body {background-color:
#FF0000;}
    </style>
```

```
</head>
<body>
  <p>Esta es una página con
fondo rojo</p>
</body>
</html>
```

3 | Estilos en línea (el atributo style): Esta es, quizá, la forma menos adecuada, versátil y recomendable de todas ya que vincula directamente los estilos de formato con el código HTML; cuando una de las principales ventajas del CSS es precisamente separar el código HTML del formato del mismo (hablaremos más tarde de esto),

aunque ayuda bastante a comprender el funcionamiento de los estilos en cascada. Su implementación se hace directamente en las etiquetas HTML con la propiedad style.

```
<html>
  <head>
    <title>Ejemplo</title>
  </head>
  <body style="background-color:
#FF0000;">
    <p>Esta es una página con
fondo rojo</p>
  </body>
</html>
```

Declaración CSS	Archivos afectados	SS se aplica a...
En línea	Uno, aquel donde se realiza la declaración	Una línea de código
Interna	Uno, aquel en cuya cabecera se realiza la declaración	Todos los elementos del archivo en los que resulte de aplicación el estilo
Externa	Todos los archivos que invoquen el archivo CSS	Todos los elementos de los archivos afectados en que resulte de aplicación el estilo



¿Qué tipo de CSS usar?

En la tabla (1) hacemos un resumen de las características de los distintos tipos de CSS que hemos visto hasta el momento.

Estilo en línea

Una de las formas más simples e intuitivas de dotar de estilos al código HTML es usando el atributo style que admiten la mayoría de las etiquetas HTML. Supongamos que sobre el código de ejemplo que estamos utilizando en el curso deseamos que el texto de los elementos del menú se muestre de color

verde y el texto de los párrafos, de color azul. Para aplicar un estilo en línea utilizaremos esta sintaxis:

<nombreDeEtiqueta style = "propiedadCSS : valorEstablecido;" > ... </nombreDeEtiqueta>

En nuestro caso para un párrafo usaríamos <p style = "color: blue;" > ... </p>

Para aplicar el color verde a los elementos del menú, que están en una lista, podemos probar aplicarle el atributo style y la propiedad color a la etiqueta El código quedaría como sigue y su resultado sería el que observamos en la imagen (2)

```
<ul style = "color: green;">
<li><a href="#" >Inicio</a></li>
<li> <a href="libros.html">Libros de
programación</a> </li>
<li> <a href="cursos.html">Cursos de
programación</a> </li>
<li> <a href="humor.html">Humor
informático</a> </li>
</ul>
```

```
<ul style = "color: red;">
<li><a href="#" style = "color: green; text-decoration:
none;">Inicio</a></li>
<li> <a href="libros.html" style = "color: green; text-decoration:
none;">Libros de programación</a> </li>
<li> <a href="cursos.html" style = "color: green; text-
decoration: none;">Cursos de programación</a> </li>
<li> <a href="humor.html" style = "color: green; text-
decoration: none;">Humor informático</a> </li>
</ul>
```



Nos planteamos ahora dejar las viñetas en color rojo y el texto del menú en color verde pero sin subrayado. Para ello tendremos que añadir una propiedad al link que elimine el estilo subrayado que por defecto incorpora el navegador. Esta propiedad será text-decoration cuyos posibles valores son none (ninguno), underline (subrayado), overline (línea superior), line-through (tachado). El código será el de la imagen (1) y el resultado el se observa en (2).

Otro ejemplo:

```
<p style="color:#990000"> Esto es un párrafo
de color rojo.</p>
<p style="color:#000099">Esto es un párrafo
de color azul. </p>
```

```
<HTML>
<HEAD>
<TITLE>Ejemplo de CSS - DIRECTOS</title>
</HEAD>
<BODY>
```

```
<H1 STYLE="font-family:Verdana,sans-
serif;font-size:14pt;color:red"> Prueba de
definición de estilo directo en una etiqueta</
H1>
</BODY>
</HTML>
```

Como se puede apreciar en este ejemplo solamente aplicamos el etilo sobre la etiqueta <H1>, en donde tenemos definido tipo de letra, tamaño y color para el texto que se encuentra encerrado dentro de esta etiqueta.



Definir los estilos para toda una página es una manera cómoda y potente de darle forma al documento y nos permitirá ahorrarnos muchas etiquetas HTML.

Estilo definido para toda una página

Podemos definir, en la cabecera del documento, estilos para que sean aplicados a toda la página. Es una manera muy cómoda de darle forma al documento y muy potente, ya que estos estilos serán seguidos en toda la página y nos ahorraremos así muchas etiquetas HTML que apliquen forma al documento. Además, si deseamos cambiar los estilos de la página lo haremos de una sola vez. Este ejemplo es más complicado, puesto que se utiliza la sintaxis CSS de manera más avanzada.

```
<html>
<head>
  <title>Ejemplo de estilos para toda una
  página</title>
  <STYLE type="text/css">
    H1 {text-decoration: underline; text-
    align:center}
    P {font-Family:arial,verdana; color: white;
    background-color: black}
    BODY {color:black;background-color:
    #cccccc; text-indent:1cm}

  </STYLE>
</head>
<body>
```

```
<h1>Página con estilos</h1>
Bienvenidos...
<p>Este es un Importante Ejemplo, para
aplicar Estilo</p>
</body>
</html>
```

En el ejemplo vemos que se utiliza la etiqueta **<STYLE>** colocada en la cabecera de la página para definir los distintos estilos del documento. A grandes rasgos, entre de **<STYLE>** y **</STYLE>**, se coloca el nombre de la etiqueta con que queremos definir los estilos y entre llaves **{}** colocamos en sintaxis CSS las características de estilos (1).



Unidades de medida

Las medidas en CSS se emplean, entre otras, para definir la altura, anchura y márgenes de los elementos y para establecer el tamaño de letra del texto. Todas las medidas se indican como un valor numérico entero o decimal seguido de una unidad de medida (sin ningún espacio en blanco entre el número y esta unidad).

Unidades Absolutas

Son medidas de un tamaño fijo, como por ejemplo, un centímetro. Solo deben usarse cuando sepamos con certeza las dimensiones del medio de salida (monitor, impresora, etc)

lo cual no siempre es posible. Las distintas Unidades Absolutas que podemos usar son las siguientes:

- | **in**: son pulgadas (en inglés "inches") y equivalen a 2.5 centímetros aproximadamente.
- | **Cm** y **mm**.
- | **pt**: el Punto equivale a 0.014 pulgadas.
- | **pc**: Las picas, equivalen a 12 puntos.

Unidades Relativas

Adoptan el valor relativo al tamaño del documento, del monitor, impresora, etc, según el caso y son las siguientes:

- | **px**: la más utilizada es el Pixel.
- | **ex**: se refiere a la altura de la "x"
- | **em**: también llamada "cuadratín" es relativa al tamaño de letra definido por font-size.

Ahora bien, una vez que ya tenemos todos los conceptos principales vistos vamos a ver como se definen y en qué zona de la página se implementan los estilos.

La sintaxis de la estructura es la que podrás observar si seguís el código de la página siguiente y la imagen (1) de esta página.

HTML	CSS	Tipo de selector
<body>	body {}	selector de etiqueta
<h1>	h1 {}	selector de etiqueta
<section>	section {}	selector de etiqueta



Selectores Básicos

El selector indica a qué hay que aplicarle la regla y la declaración especifica de qué es lo que hay que hacerle. Dentro de la declaración nos encontraremos con las propiedades y sus valores, que son los que al final marcarán el estilo de nuestro sitio: colores, tipografías, márgenes, etc. El primero al que recurrimos es al selector de etiqueta, porque aprovechamos las etiquetas que ya están escritas en HTML y le creamos una regla que la modifique.

Por ejemplo: si en el HTML tenemos la etiqueta <body>, para crear una regla en CSS simplemente escribo el nombre de la etiqueta body {} tal como se observa en la imagen (1)

Ahora veamos con detenimiento cada parte del selector. Podés irte guiando con las imágenes (2) y (3)

| Selector: El selector le dice al navegador web qué directiva o directivas de una página tienen que someterse al estilo. En la imagen, el selector (h1) se refiere a la directiva <h1>, que hace que los buscadores web den formato a todas las etiquetas <h1> usando las direcciones de formato recogidas en este estilo.

| Bloque de declaración: El código que sigue al selector incluye todas las opciones de formato que se pueden aplicar al selector. El bloque comienza con una llave de apertura ({) y termina con una llave de cierre (}).

Como veras, en el ejemplo están codificadas todas las secciones de html5 y los estilos se han definido en la zona de <Style>. A continuación pasaremos a definir algunos componentes en la definición de estilos.

| Declaración: Entre las llaves de apertura y de cierre de la declaración, se añade una o más declaraciones o instrucciones de formato. Cada declaración tiene dos partes, una propiedad y un valor, y además acaban en un punto y coma.

| Propiedad: CSS ofrece un amplio rango de opciones de formato, que se llaman propiedades. Una propiedad es una palabra que indica un cierto estilo de efecto. La mayoría de las propiedades tienen nombres sencillos como `font-size` (tamaño de fuente), `margin-top` (margin superior), etc.

| Valor: Finalmente se puede expresar la creatividad asignando un valor a una propiedad CSS (como hacer un fondo azul, por ejemplo). Las diferentes propiedades CSS requieren tipos específicos de valores: un color (como rojo o `#ff0000`), una longitud (como 16px, o 5em), o una clave específica (como `top`, `center` o `bottom`), como muestra la imagen (1)

Podemos aplicar el mismo estilo en un conjunto de etiquetas. Para ello, indicamos las etiquetas seguidas por comas y luego, entre llaves, los atributos que deseamos definir (2)

En la próxima clase retomaremos este tema para continuar con el resto de los selectores.

Propiedades del Texto

Font-family

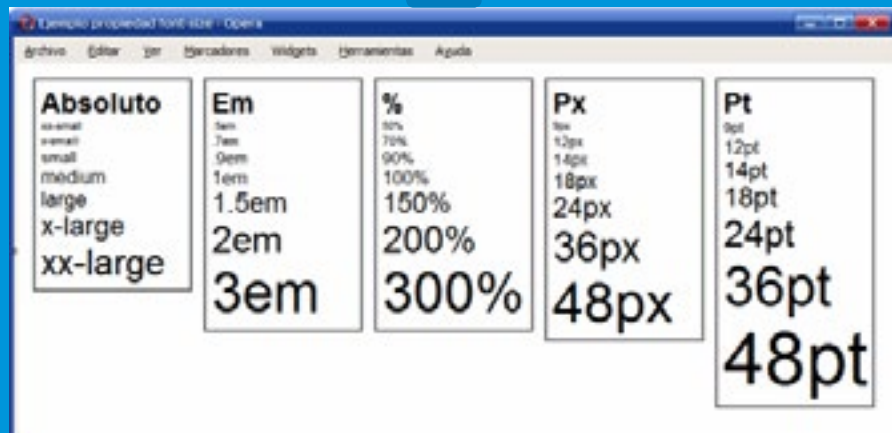
font-family se utiliza para indicar el tipo de letra con el que se muestra el texto.

El nombre genérico *familia tipográfica* no se refiere a ninguna fuente en concreto, sino que hace referencia al estilo del tipo de letra. Las familias genéricas definidas son *serif* (tipo de letra similar a Times New Roman), *sans-serif* (tipo Arial), *cursive* (tipo Comic Sans), *fantasy* (tipo Impact) y *monospace* (tipo Courier New). Un ejemplo puede ser:

```
p {
  color:blue;
  font-size:20;
  font-weight:bold;
  font-family:Arial;
}
```

```
p, h1, h5 {
  color:#ffcc03;
  font-family:Arial;
}
```

font-family:{ Georgia, "Times New Roman", Times, serif};



Valores

normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900

Font-size

Una vez seleccionado el tipo de letra, se puede modificar su tamaño mediante la propiedad *font-size*. Además de todas las unidades de medida relativa y absoluta y el uso de porcentajes, CSS permite utilizar una serie de palabras clave para indicar el tamaño de letra del texto (1).

| **tamaño_absoluto**: indica el tamaño de letra de forma absoluta mediante alguna de las palabras clave, tales como *xx-small*, *x-small*, *small*, *medium*, *large*, *x-large*, *xx-large*.

| **tamaño_relativo**: indica de forma relativa el tamaño de letra del texto mediante dos palabras clave (*larger*, *smaller*) que toman

como referencia el tamaño de letra del elemento padre.

CSS recomienda indicar el tamaño del texto en la unidad **em** o en porcentaje (%). Además, es habitual indicar el tamaño del texto en puntos (pt) cuando el documento está específicamente diseñado para imprimirlo. Por defecto los navegadores asignan los siguientes tamaños a los títulos de sección:

<h1> = xx-large, <h2> = x-large, <h3> = large, <h4> = medium, <h5> = small, <h6> = xx-small.

Font-weight

Una vez indicado el tipo y el tamaño de letra, es habitual modificar otras características como su grosor (texto en negrita) y su estilo (texto en cursiva). La propiedad que controla la anchura de la letra es **font-weight**. (2)

Los valores que normalmente se utilizan son *normal* (el valor por defecto) y *bold* para los textos en negrita. El valor *normal* equivale al valor numérico 400 y el valor *bold* al valor numérico 700.

1

Valores

normal | italic | oblique

3

Valores

left | right | center | justify

5

```
p { line-height: 1.2; font-size: 1em; }
p { line-height: 1.2em; font-size: 1em; }
p { line-height: 120%; font-size: 1em; }
```

2

Valores

normal | small-caps

4

Valores

normal | numero | unidad
de medida | porcentaje

Font-style

Además de la anchura de la letra, CSS permite variar su estilo mediante la propiedad *font-style* (1).

Normalmente la propiedad *font-style* se emplea para mostrar un texto en cursiva mediante el valor *italic*.

Font-variant

Por último, CSS permite otra variación en el estilo del tipo de letra, controlado mediante la propiedad *font-variant*. Esta propiedad

no se suele emplear habitualmente, ya que sólo permite mostrar el texto con *letra versal* (mayúsculas pequeñas)(2).

Alineación e interlineado

La propiedad que define la alineación del texto se denomina *text-align* (3).

Los valores definidos por CSS permiten alinear el texto según los valores tradicionales: a la izquierda (*left*), a la derecha (*right*), centrado (*center*) y justificado (*justify*).

Por su parte, el interlineado de un texto se controla mediante la propiedad *line-height*,

que permite controlar la altura ocupada por cada línea de texto(4).

Además de todas las unidades de medida y el uso de porcentajes, la propiedad *line-height* permite indicar un número sin unidades que se interpreta como el múltiplo del tamaño de letra del elemento. Por lo tanto, estas tres reglas CSS son equivalentes (5).

1

Valores

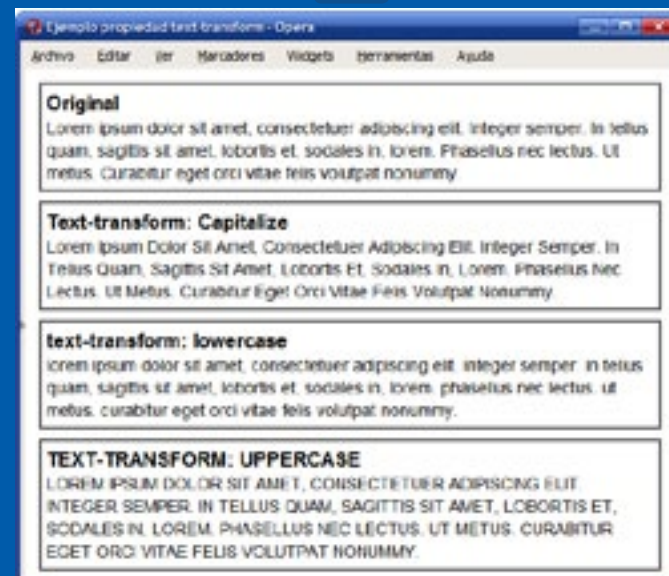
none | (underline || overline
|| line-through || blink)

2

Valores

capitalize | uppercase |
lowercase | none

3



Estilos para el texto en su conjunto

Text-decoration

Además de la decoración que se puede aplicar a la tipografía que utilizan los textos, CSS define otros estilos y decoraciones para el texto en su conjunto. La propiedad que decora el texto se denomina *text-decoration* (1).

El valor *underline* subraya el texto, por lo que puede confundir a los usuarios haciéndoles creer que se trata de un enlace. El valor *overline* añade una línea en la parte superior

del texto, un aspecto que raramente es deseable. El valor *line-through* muestra el texto tachado con una línea continua, por lo que su uso tampoco es muy habitual. Por último, el valor *blink* muestra el texto parpadeante y se recomienda evitar su uso por las molestias que genera a la mayoría de usuarios.

Text- transform

Una de las propiedades de CSS más desconocidas y que puede ser de gran utilidad en algunas circunstancias es la propiedad

text-transform, que puede variar de forma sustancial el aspecto del texto (2).

La propiedad *text-transform* permite mostrar el texto original transformado en un texto completamente en mayúsculas (*uppercase*), en minúsculas (*lowercase*) o con la primera letra de cada palabra en mayúscula (*capitalize*). Podés observar un ejemplo de esto en la imagen (3).

Valores

baseline | sub | super | top | text-top | middle | bottom |
text-bottom | porcentaje | unidad de medida

Valores

unidad de medida |
porcentaje

Valores

normal | unidad de medida



Vertical-align

Uno de los principales problemas del diseño de documentos y páginas mediante CSS consiste en la alineación vertical en una misma línea de varios elementos diferentes como imágenes y texto. Para controlar esta alineación, CSS define la propiedad *vertical-align* (1).

El valor por defecto es **baseline** y el valor más utilizado cuando se establece la propiedad **vertical-align** es **middle**.

Text-indent

En muchas publicaciones impresas suele ser habitual tabular la primera línea de cada párrafo para facilitar su lectura. CSS permite controlar esta tabulación mediante la propiedad *text-indent* (2).

Letter-spacing y word-spacing

CSS también permite controlar la separación entre las letras que forman las palabras y la separación entre las palabras que forman los textos. La propiedad que controla la

separación entre letras se llama *letter-spacing* y la separación entre palabras se controla mediante *word-spacing* (3).

La imagen (4) muestra la comparación entre un texto normal y otro con las propiedades *letter-spacing* y *word-spacing* aplicadas.

Cuando se utiliza un valor numérico en las propiedades *letter-spacing* y *word-spacing*, se interpreta como la separación adicional que se añada (si el valor es positivo) o se quita (si el valor es negativo) a la separación por defecto entre letras y palabras respectivamente.

Valores	normal pre nowrap pre-wrap pre-line
---------	---

Valor	Respetar espacios en blanco	Respetar saltos de línea	Ajusta las líneas
normal	no	no	si
pre	si	si	no
nowrap	no	no	no
pre-wrap	si	si	si
pre-line	no	si	si



White-space

CSS también permite controlar el tratamiento de los espacios en blanco de los textos mediante la propiedad *white-space* (1).

El significado de cada uno de los valores es el siguiente:

- | **normal**: comportamiento por defecto de HTML.
- | **pre**: se respetan los espacios en blanco y las nuevas líneas (exactamente igual que la etiqueta <pre>). Si la línea es muy larga, se sale del espacio asignado para ese contenido.

| **nowrap**: elimina los espacios en blanco y las nuevas líneas. Si la línea es muy larga, se sale del espacio asignado para ese contenido.

| **pre-wrap**: se respetan los espacios en blanco y las nuevas líneas, pero ajustando cada línea al espacio asignado para ese contenido.

| **pre-line**: elimina los espacios en blanco y respeta las nuevas líneas, pero ajustando cada línea al espacio asignado para ese contenido.

En la tabla (2) se resumen las características de cada valor.

La imagen (3) muestra las diferencias entre los valores permitidos para *white-space*. El párrafo original contiene espacios en blanco y nuevas líneas y se ha limitado la anchura de su elemento contenedor.

Pseudo-elementos

Por último, CSS define unos elementos especiales llamados "**pseudo-elementos**" que permiten aplicar estilos a ciertas partes de un texto. En concreto, CSS permite definir estilos especiales a la primera frase de un texto y a la primera letra de un texto.

First-line

El pseudo-elemento **first-line** permite aplicar estilos a la primera línea de un texto. Las palabras que forman esta primera línea dependen del espacio reservado para mostrar el texto o del tamaño de la ventana del navegador, por lo que CSS calcula de forma automática las palabras que forman la primera línea de texto en cada momento. La regla CSS utilizada para los párrafos del ejemplo se muestra a continuación:

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

First-letter

De la misma forma, CSS permite aplicar estilos a la primera letra del texto mediante el pseudo-elemento **first-letter**.

Por ejemplo,

```
p:first-letter {
  font-size: 2.5em; font-weight: bold;
  line-height: .9em; float: left;
  margin: .1em;
}
```

Más pseudo-elementos

Otros pseudo-elementos pueden ser:

- :before.** Nos posiciona al inicio del contenido de un elemento.
- :after.** Nos posiciona al final del contenido de un elemento.

La imagen (1) muestra dónde apuntaría el selector en cada uno de los casos.

En la clase siguiente continuaremos con CSS3, profundizando este tema, ahora te dejo con una serie de ejercicios para que apliques lo aprendido, es decir, las tablas y formatos con estilo.



Código	Resultado
<pre><head> < style type="text/css"> p:first-line {color: #006699;} p:first-letter {font-size: xx-large;} < /style> < /head> < body> <p>Este es un ejemplo para la clase 4 de DAI1. Recuerda, no dejes de practicar los ejemplos.<p> < /body></pre>	Este es un ejemplo para la clase 4 de DAI1. Recuerda, no dejes de practicar los ejemplos.



Desempeños

Con lo visto hasta ahora, probá desarrollar los siguientes ejercicios.
Seguramente te falten aprender estilos, lo importante de esto es que
vayas codificando las estructuras y con el desarrollo de la clase siguiente
puedas terminarlos ¡Mucha suerte!



Desempeño 1

Animales en peligro de extinción			
Nombre	Cabezas	Previsión 2010	Previsión 2020
Ballena	6000	4000	1500
Oso Pardo	50	0	
Lince	10		
Tigre	300	210	

Desempeño 2

Desempeño 3

Climas de América del Sur

Parte de arriba de América del Sur.

Países como:

Venezuela
Colombia
Ecuador
Perú

Parte de abajo de América del Sur.

Países como:

Argentina
Chile
Uruguay
Paraguay

Bosque tropical, clima de sabana, clima marítimo con inviernos secos.

Climas marítimos con veranos secos, con inviernos secos, climas fríos, clima de estepa, clima desértico.

DIFERENCIAS ENTRE EL PERRO Y EL HOMBRE

DIFERENCIAS	PERRO		HOMBRE
	PEQUEÑO	GRANDE	
Duración crecimiento	10 meses	18 a 24 meses	16 años
Tiempo de gestación	58 a 63 días		9 meses
Duración de vida del pelo/cabello	1 año		2 a 7 años

Desempeño 4

Desempeño 5

<u>Gastronomía</u>	Restaurantes	
	Sidrerías y asadores	
	Bodegas de vino y tsakoli	
	Productos típicos	
	Escuelas de hostelería	
<u>Cultura</u>	archivos y bibliotecas	
	museos	
	palacios de congresos	
	ferias de muestras	
<u>Ocio</u>	Entretenimiento y diversión	
	Ocio cultural	
	Excursiones y deporte	

A. S. I. R.

	HORAS	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES
1	8.30-9.25	PAR	LMSGI	PAR	LMSGI	ISO
2	9.25-10.20	LMSGI			FH	
3	10.20-11.10	ISO	PAR	FOL	PAR	PAR
-	11.10-11.40	RECREO				
4	11.40-12.35	GBD	ISO	ISO	PAR	FOL
5	12.35-13.30	GBD	FOL		GBD	GBD
6	13.30-14.20	FH	FH	GBD	ISO	

LEYENDA DE ABREVIATURAS

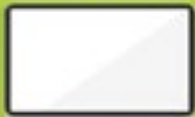
Abreviatura	Descripción
FH	Fundamentos de Hardware
LMSGI	Lenguajes de marca y sistemas de gestión de la información
PAR	Planificación y Administración de Redes
GBD	Gestión de Bases de Datos
FOL	Formación y Orientación Laboral
ISO	Implantación de Sistemas Operativos

Desempeño 6

Desempeño 7

Vermús

Una serie de talleres organizados por el Colectivo Mause enfocados al aprendizaje y disfruta en general.



HTML5 + CSS3 – Nivel 1

20h de HTML y CSS para superar todos tus miedos. Bueno, todos no, pero los referidos. Alto de picar código quedarán totalmente superados. Hazte tu web de una vez ya!


Impartido por Óscar Marí,
picador de código y colaga Muuu

Tabla de Clientes

CODIGO	NOMBRE	APELLIDOS	DIRECCION	TELEFONO	CREDITO	TOTAL COMPRA	CREDITO RESTANTE
	ILIAN ALBERTO	CARIO FERNANDEZ	Col. La Rabida	2278-4589	\$ 500.00		
	LIDIA DEL CARMEN	BLANCO GAITAN	Calle El Jocote	2859-1545	\$ 325.00		
	ANA MARIA	RIVAS BELTRAN	Col. Santa Anita	7789-9088	\$ 900.00		
	HUGO ALEXANDER	DIAZ ANZORA	Col. Malaga	7113-4545	\$ 450.00		
	CLAUDIA YANIRA	ALAS CASTRO	Calle EL Rosario	2225-6678	\$ 600.00		
	ERIKA VANESSA	BLANCO RIVAS	Col. Miramonte	2290-1845	\$ 425.00		
	CARLOS AARON	RIVAS VALDES	Col. El Paraíso	7598-5560	\$ 550.00		

	Detección (40%)	Rendimiento (30%)	Funciones y calidad (30%)	Rating final
Ad-Aware Free 10	9,47	6,64	7	7,88
avast! Free 7	9,09	5,01	10	8,14
AVG Free 2013	9,00	7,13	9	8,44
Avira Personal 2013	9,34	7,54	9	8,70
MSE 4.1	8,41	8,00	7	7,86
*Panda Cloud 2.1	9,42	(2,85)	7	6,72

- 1 | Ejemplificá los atributos de bordes para las tablas (rules y frames).
- 2 | ¿Cómo se controla el espacio entre la tabla y los bordes de las celdas?
- 3 | ¿Cómo pueden establecerse anchos diferentes para cada columna?
- 4 | Investigá cuál es el uso que se les da a las tablas en HTML5.
- 5 | ¿Cómo se coloca una imagen en el fondo de una celda de una tabla?



¡Felicitaciones, llegaste al final de esa cuarta clase! Tal vez te haya resultado difícil codificar estos últimos ejercicios, pero no te preocupes, en la clase siguiente veremos algunos estilos que te faltan aprender.

¡Hasta la próxima!