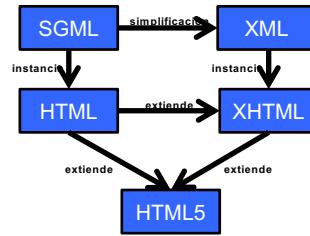


Sistemas Informáticos



Tema 2.2: Web Hipertexto

El lenguaje SGML



Standard Generalized Markup Language, SGML

- Estándar internacional para la definición de texto electrónico independiente de dispositivos, sistemas y aplicaciones.
- Metalenguaje para definir lenguajes de diseño descriptivos.
- Proporciona un medio de codificar documentos hipertexto cuyo destino sea el intercambio directo entre sistemas o aplicaciones.
- Estándar del ISO: ISO-8879-1/1986.
- Múltiples lenguajes definidos mediante SGML.
 - *Hypertext Markup Language, HTML.*
- Evolución de SGML: *Extended Markup Language, XML.*



SGML - Características

- SGML tiene cinco características principales
 - Permite crear lenguajes de codificación descriptivos.
 - Códigos incluidos en el propio texto definen los elementos que lo componen.
 - Define una estructura de documento jerárquica, con elementos y componentes interconectados.
 - Proporciona una especificación formal completa del documento.
 - Contenida en la *Document Type Definition, DTD*.
 - No tiene un conjunto implícito de convenciones de señalización. Soporta, por tanto, un conjunto flexible de juegos de etiquetas.
 - SGML es un metalenguaje. Definición de etiquetas para cada lenguaje en su DTD.
 - Los documentos generados con él son legibles por personas.



Ejemplo SGML – DTD de HTML 4

```
<!-- Parameter Entities -->

<!ENTITY % head.misc "SCRIPT|STYLE|META|LINK" -- repeatable head elements -->

<!ENTITY % heading "H1|H2|H3|H4|H5|H6">

<!ENTITY % list "UL | OL | DIR | MENU">

<!ENTITY % preformatted "PRE">

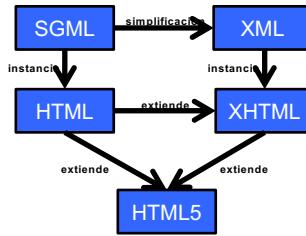
.....



<!ELEMENT FONT -- (%inline)* -- local change to font -->
<!ATTLIST FONT
  size   CDATA  #IMPLIED -- [+]nn e.g. size="+1", size=4 --
  color  CDATA  #IMPLIED -- #RRGGBB in hex, e.g. red: "#FF0000" --
  face   CDATA  #IMPLIED -- comma separated list of font names --
>
```



Sistemas Informáticos



Tema 2.2: Web Hipertexto

El lenguaje HTML



Hypertext Markup Language, HTML

- Aplicación del estándar ISO 8879, SGML.
- Lenguaje de definición de formato de documento.
 - Define elementos dentro del documento.
 - Se especifican mediante etiquetas (*Tags*) de comienzo y final de documento: Texto ASCII encerrado entre <*y*>.
 - Sólo se define el tipo de elemento, y no la forma de representarlo.
- Lenguaje estándar utilizado en la Web para crear documentos hipermedia, que incluyen:
 - Texto
 - Imágenes (Fotos, Vídeo)
 - Audio
 - Vínculos (Links)
- Estándar especificado por el *World Wide Web Consortium, W3C*:
<http://www.w3c.org/>



Estructura de un documento HTML

- Un documento HTML consta de las siguientes partes:
 - Identificación SGML.
 - Permite identificar la DTD adecuada para procesarlo.
 - Cabecera. Información asociada al documento.
 - Identificado por la etiqueta <HEAD>.
 - Contiene información descriptiva del documento y definiciones de formato para los elementos del resto del mismo.
 - Cuerpo del documento.
 - Identificado con la etiqueta <BODY>
 - Contiene los elementos de presentación del documento.



Elementos del Lenguaje HTML

- Ficheros HTML
 - usualmente terminados con el sufijo “html” (o bien “htm” en algunos servidores PC)
 - ASCII puro
 - Contienen:
 - información de propiedades de todo el documento
 - define el formato del documento
 - tags (Etiquetas)
 - identificación de vínculos a otros objetos/documentos
 - contenido



Elementos del Lenguaje HTML

- TAGS (Etiquetas)
 - No vacíos:
 - La mayoría de elementos comienzan por un tag, luego contenido, finalizando por un tag de terminación
 - <TAG> contenido </TAG>
 - Vacíos: por ejemplo ,
, etc.
- HTML es insensible a Mayúsculas / Minúsculas:
 - <TAG> es equivalente a <tag> o <TaG>
- Espacios en blanco, tabulaciones y retornos de carros no son significativos en HTML.



2. Estructura de un documento HTML

```
<HTML>
  <HEAD>
    Definiciones de
    la cabecera
  </HEAD>
  <BODY>
    Instrucciones
    HTML
  </BODY>
</HTML>
```



- Etiquetas de apertura y cierre de documento
- Documento dividido en dos partes: Cabecera y Cuerpo

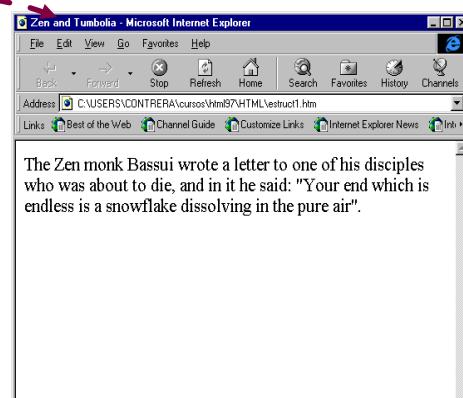
Comentarios

- Texto contenido entre <!-- y -->
- puede aparecer en cualquier lugar de un documento HTML
- útil para identificar un documento: Autor, fecha, etc.
- No se muestran en los navegadores
- <!-- Esto es un comentario -->



Cabecera (<HEAD>)

```
<HTML>
<HEAD>
<TITLE>Zen and Tumbolia</TITLE>
</HEAD>
<BODY>
The Zen monk Bassui wrote a letter
to one of his disciples who was
about to die, and...
</BODY>
</HTML>
```



- Contiene el título del documento
- Utilizado también en Bookmarks y History Lists

Depende del navegador !!!



Cuerpo (**<BODY>**)

- Cuerpo de un documento HTML delimitado **<BODY>** y **</BODY>**
- Se incluirán todas las instrucciones HTML y el texto que forman el documento
- Si un documento no presenta ninguna de las etiquetas de identificación de sus distintas partes (**<HTML>**, **<HEAD>** ó **<BODY>**) se considerará que todo lo que se defina pertenece al cuerpo del documento.



Atributos de **<BODY>**

- BACKGROUND (URL, imagen de fondo)
- BGCOLOR (color de fondo)
- TEXT (color del texto)
- LINK (color del texto de enlaces no visitados)
- VLINK (color del texto de enlaces visitados)
- ALINK (color del texto de enlaces mientras el usuario hace click en él)
- Ej: `<body bgcolor=white text=black link=red vlink=maroon alink=fuchsia>`



Imágenes

- Se puede incluir cualquier imagen en formato PNG, JPEG, GIF o XBM.
 - En un principio el más extendido era GIF
 - soportado por todos los navegadores
 - programas de tratamientos de gráficos que graban y convierten a GIF
 - utiliza algoritmo de compresión ⇒ ficheros pequeños
 - recomendado para iconos, gráficas
 - Problema: formato propietario.
 - Actualmente reemplazo por PNG.
 - También comprimido y permite mayor profundidad de color.
 - Es libre.



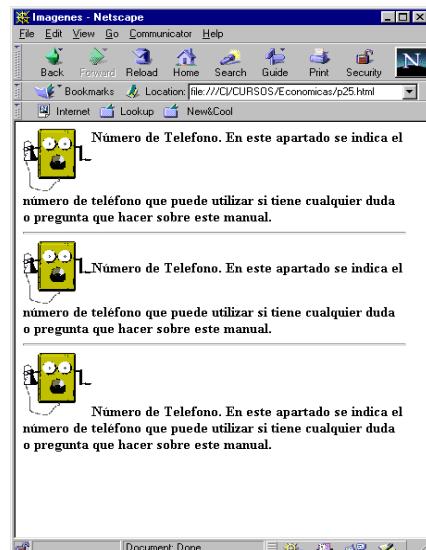
Etiqueta IMG

-
- Se pueden insertar imágenes
 - en cualquier parte del documento
 - dentro de cualquier estructura HTML (listas, tablas, etc)
- fichero debe tener extensión .gif o .jpg o .jpeg
- *fichero_imagen* puede estar en otro servidor ⇒ especificamos URL de la imagen



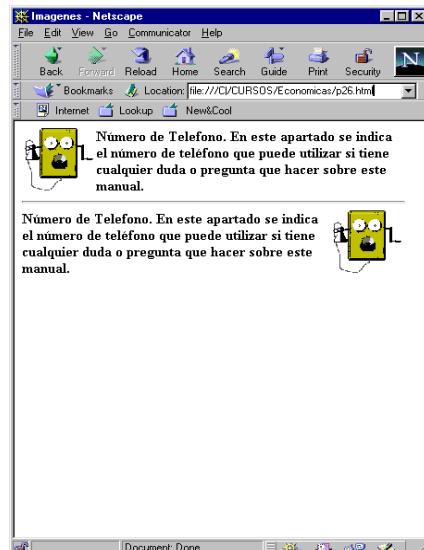
Atributos de IMG (II)

- ALIGN
 - TOP
 - MIDDLE
 - BOTTOM
 - indica dónde se coloca primera línea del texto que sigue a la imagen
 -



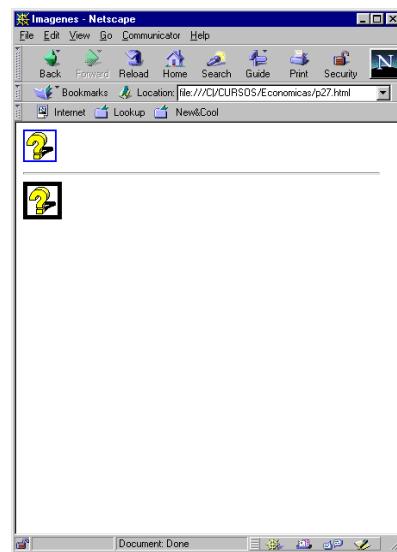
Atributos de IMG (III)

- ALIGN
 - LEFT
 - RIGHT
 - indica si la imagen queda a la izquierda o a la derecha del texto
 -



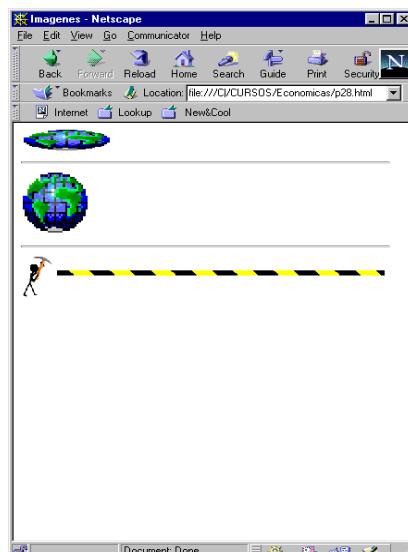
Atributos de IMG (IV)

- BORDER=n
 - indica tamaño del borde de la imagen
 - si dentro de hiperenlace, color apropiado
 - BORDER=0 ⇒ sin borde
 -



Atributos de IMG (V)

- HEIGHT y WIDTH
 - indican alto y ancho de imagen
 - se especifican en pixels o en porcentaje del tamaño del documento
 - si sólo se indica una, se ajusta la otra



Hiperenlaces

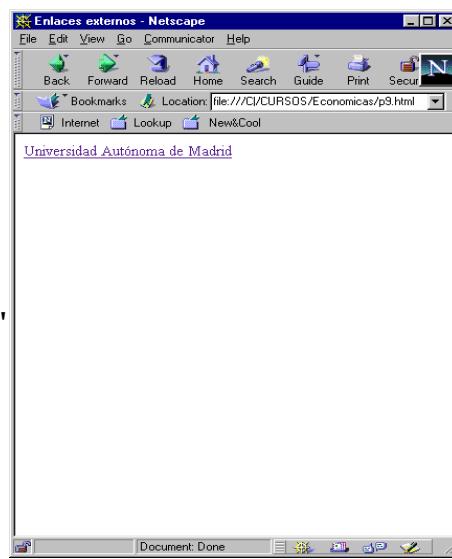
- Elemento que establece conexión con otras páginas o zonas de la misma página
- Se pueden definir en texto, cabeceras, imágenes, ...
- Texto en azul y subrayado
 - podemos cambiar color mediante atributos de BODY
- Evitar documentos largos
 - utilizar hiperenlaces para dividir
- Creación de enlaces
 - Enlaces a documentos HTML externos
 - Enlaces locales
 - Enlaces a recursos externos



Enlaces a documentos HTML externos

```
<A HREF="url">texto</A>

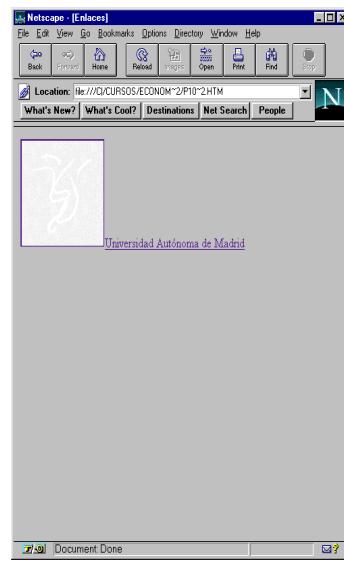
<HTML>
<HEAD>
<TITLE>Enlaces</TITLE>
</HEAD>
<BODY>
<A
    HREF="http://www.uam.es"
    >
    Universidad
    Aut&oacute;noma de
    Madrid</A>
</BODY>
</HTML>
```



Enlaces desde imágenes

```
<A HREF="url"><IMG SRC="imagen>texto</A>

<HTML>
<HEAD>
<TITLE>Enlaces</TITLE>
</HEAD>
<BODY>
<A HREF="http://www.uam.es">
    <IMG SRC="decor.gif">
    Universidad Aut&acute;noma de
    Madrid</A>
</BODY>
</HTML>
```



URL absolutas y relativas

- URL absoluta = URL completa
 $http://nombre_dominio/path/fichero.html$
- URL relativa = se puede omitir nombre de dominio y path
 - lo que se omite se toma de la URL del documento que contiene el hiperenlace
 - puede comenzar por
 - / → camino desde el directorio raíz
 - ../ → subir un nivel en estructura de directorios
 - nombre fichero o directorio → a partir de directorio actual
 - etiqueta BASE indica dirección base



Tablas

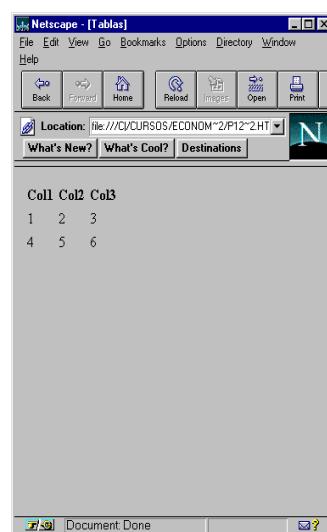
- Permite representación por filas y columnas
- Se calcula automáticamente el tamaño de celdas
- Se pueden introducir cualquier elemento: imágenes, enlaces, cabeceras, etc.
- No es necesario definir número de filas o columnas
- Definición
 - Inicio y fin de tabla
`<TABLE> </TABLE>`
 - Definición de filas
`<TR> </TR>`
 - Definición de celdas
 - Celdas de datos `<TD> </TD>`
 - Celdas cabecera `<TH> </TH>`



Ejemplo Tabla

```

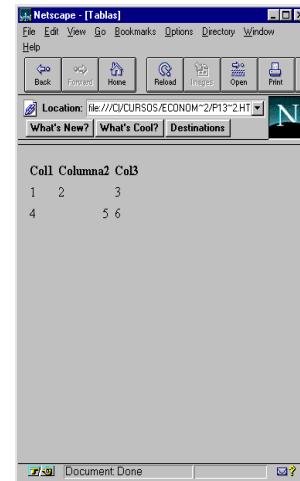
<HTML>
<HEAD>
<TITLE>Tablas</TITLE>
</HEAD>
<BODY>
<TABLE>
<TR> <TH>Col1</TH> <TH>Col2</TH>
      <TH>Col3</TH></TR>
<TR> <TD>1</TD> <TD>2</TD> <TD>3</TD>
      </TR>
<TR> <TD>4</TD> <TD>5</TD> <TD>6</TD>
      </TR>
</TABLE>
</BODY>
</HTML>
  
```



Atributos de TH y TD (I)

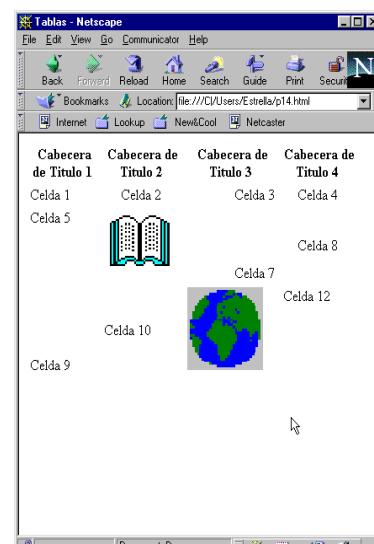
- **ALIGN** - alineación horizontal dentro de la celda

```
<HTML>
<HEAD><TITLE>Tablas</TITLE></HEAD>
<BODY>
<TABLE>
<TR> <TH>Col1</TH> <TH
    ALIGN=“CENTER”>Columna2</TH>
<TH>Col3</TH></TR>
<TR> <TD>1</TD> <TD ALIGN=“LEFT”>2</TD>
    <TD>3</TD> </TR>
<TR> <TD>4</TD> <TD ALIGN=“RIGHT”>5</TD>
    <TD>6</TD> </TR>
</TABLE>
</BODY>
</HTML>
```



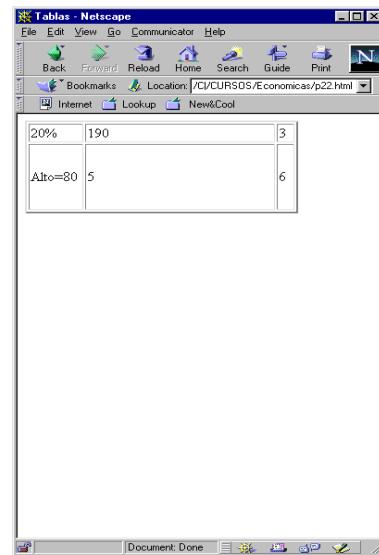
Atributos de TH y TD (II)

- **VALIGN** - alineación vertical dentro de la celda
 - TOP (Celda 12)
 - BOTTOM (Celda 9)
 - MIDDLE (Celda 10)



Atributos de TH y TD (III)

- **WIDTH** - ancho de la celda
 - ancho absoluto (pixels)
 - porcentaje del ancho de la tabla
- **HEIGHT** - alto de la celda
 - ancho absoluto (pixels)



The screenshot shows a table with three columns and two rows. The first column has a width of 20% and a height of 80 pixels. The second column has a width of 190 pixels and a height of 5 pixels. The third column has a width of 3 pixels and a height of 6 pixels.

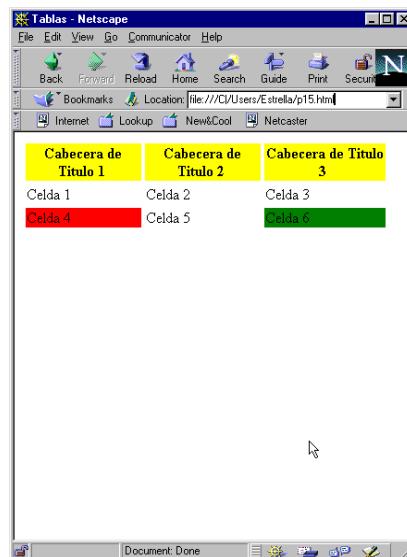
Alto=80	5	6
---------	---	---



Atributos de TH y TD (IV)

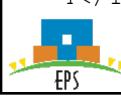
- **BGCOLOR**
 - color de fondo de la celda
 - independiente para cada celda
 - mismo formato que para atributos de BODY
 - RRGGBB
 - ejemplo

```
<TD BGCOLOR=RED>Celda  
4</TD>
```



The screenshot shows a table with three columns and four rows. The first row is a header row with the cells labeled 'Cabecera de Titulo 1', 'Cabecera de Titulo 2', and 'Cabecera de Titulo 3'. The second row contains cells 'Celda 1', 'Celda 2', and 'Celda 3'. The third row contains cells 'Celda 4' (red background), 'Celda 5', and 'Celda 6' (green background). The fourth row contains cells 'Celda 7', 'Celda 8', and 'Celda 9'.

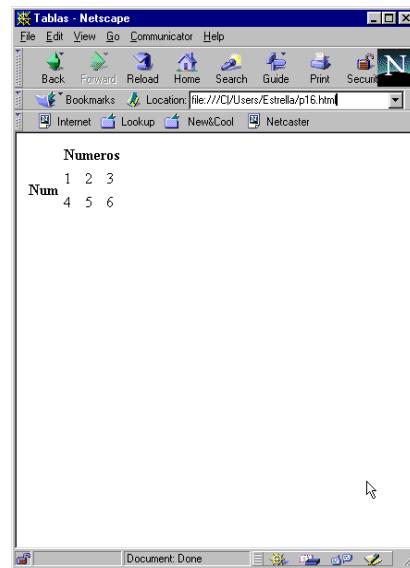
Cabecera de Titulo 1	Cabecera de Titulo 2	Cabecera de Titulo 3
Celda 1	Celda 2	Celda 3
Celda 4	Celda 5	Celda 6
Celda 7	Celda 8	Celda 9



Atributos de TH y TD (V)

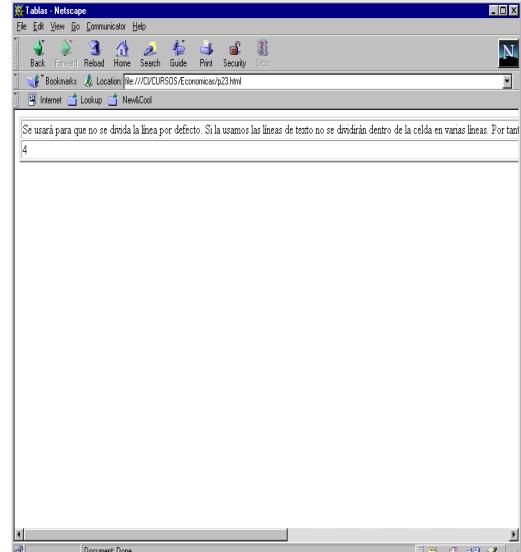
- ROWSPAN y COLSPAN
 - número de filas (o columnas) que ocupa la celda

```
<TABLE>
<TR>
<TD><TH COLSPAN=3>Numeros
<TR>
<TH
  ROWSPAN=2>Num<TD>1<TD>2<TD>3
<TR>
<TD>4<TD>5<TD>6
</TABLE>
```



Atributos de TH y TD (VI)

- NOWRAP
 - indica que las líneas de texto no se dividan dentro de la celda
 - puede causar celdas extremadamente anchas



Atributos de TR

- Afectan a todas las celdas de la fila
- ALIGN - alineación horizontal
 - LEFT, CENTER, RIGHT
- VALIGN - alineación vertical
 - TOP, MIDDLE, BOTTOM
- BGCOLOR -color de fondo



Título de la tabla

```
<CAPTION ALIGN=TOP|BOTTOM>título</CAPTION>
```

- El título aparecerá fuera de la tabla centrado horizontalmente
- ALIGN - indica si el título debe aparecer
 - arriba (TOP) o
 - debajo (BOTTOM) de la tabla

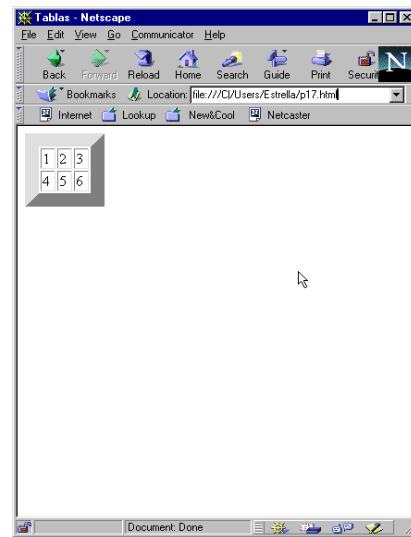


Atributos de TABLE (I)

- **BORDER=n**

- se dibuja borde alrededor de la tabla y separando celdas
- n indica grosor en pixels
- por defecto n = 0
- ejemplo

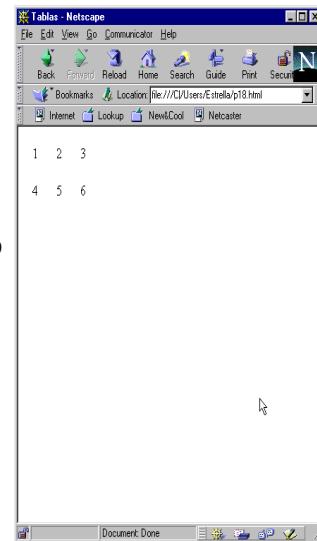
```
<TABLE BORDER=1>
```



Atributos de TABLE (II)

- **CELLSPACING=n**
 - espacio entre celdas
 - por defecto n = 2
- **CELLPADDING=n**
 - espacio entre borde de celda y contenido
 - por defecto n = 1

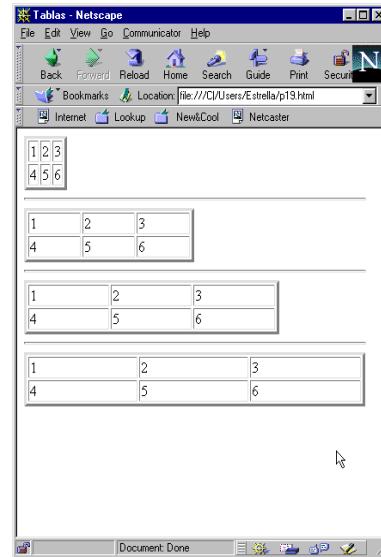
```
<TABLE CELLPADDING=10  
CELLSPACING=0>
```



Atributos de TABLE (III)

- **WIDTH=valor o porcentaje**
 - ancho de la tabla
 - valor absoluto en puntos
 - porcentaje del tamaño del documento
 - ejemplo

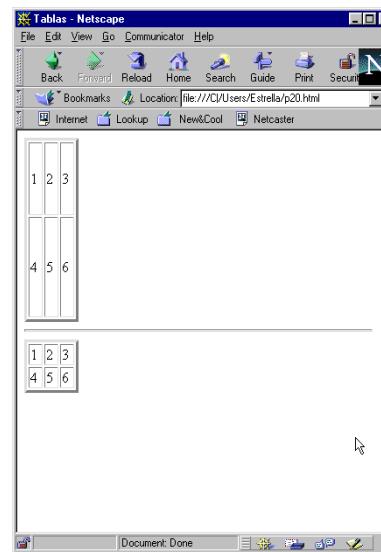
```
<TABLE WIDTH="50%">
```



Atributos de TABLE (IV)

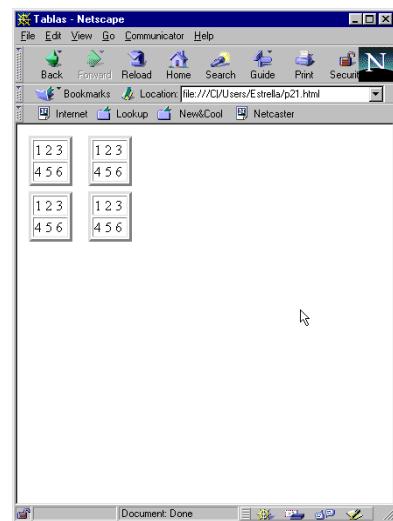
- **HEIGHT=valor o porcentaje**
 - alto de la tabla
 - valor absoluto en puntos
 - porcentaje del tamaño del documento
 - ejemplo

```
<TABLE HEIGHT=200>
```



Tablas anidadas

```
<TABLE>
<TR>
    <TD>tabla <TD>tabla
<TR>
    <TD>tabla <TD>tabla
</TABLE>
```



Frames (Marcos)

- Con los frames (iframes en HTML5) varios documentos HTML pueden ser mostrados en la misma ventaja del navegador.
 - Cada documento es llamado un frame y cada frame es independiente de los otros
- Difíciles de usar:
 - Por ejemplo, imprimir toda la página, *bookmarks*, accesibilidad.
- El desarrollo web se complica por los documentos HTML adicionales



Ejemplo de *iframe*

```
<html>
  <body>
    <p>Ejemplo de página con iframes</p>
    <table width="100%">
      <tr>
        <td>
          <iframe src="frame_a.htm" height="200px" width="100%"></iframe>
        </td>
        <td>
          <iframe src="frame_b.htm" height="200px" width="100%"></iframe>
        </td>
      </tr>
    </table>
  </body>
</html>
```

Ejemplo de página con iframes

Frame_a Frame_b



Layout en HTML

- Dos formas de organizar dónde va cada cosa:

Main Title of Web Page

Menu
HTML
CSS
JavaScript

Content goes here

Copyright © 2011 W3Schools.com

- Tablas
- Etiquetas <div>
- CSS



Layout utilizando <table>

- El único disponible en las primeras versiones de HTML
- Desventajas:
 - Obliga a crear ficheros muy grandes y complejos.
 - Obliga a descargar información de presentación con cada página que se descarga.
 - El rediseño de sitios y contenido existente es más laboriosos (y costoso).
 - Es difícil (y costoso) mantener consistencia visual en un sitio.
 - Las páginas basadas en tablas son menos accesibles a usuarios con necesidades especiales y a los que usan teléfonos móviles, PDAs o tabletas para acceder.



Layout utilizando <table> - Ejemplo

```
<html>
<body>
<table width="500" border="0">
<tr>
<td colspan="2" style="background-color:#FFA500;">
<h1>Main Title of Web Page</h1>
</td>
</tr>

<tr valign="top">
<td style="background-color:#FFD700; width:100px; text-align:left;>
<b>New<br/>HTML<br/>CSS<br/>Java</b>
</td>
<td style="background-color:#EEEEEE; height:200px; width:400px; text-align:top;">
Content goes here</td>
</tr>
<tr>
<td colspan="2" style="background-color:#FFA500; text-align:center;">
Copyright © 2011
W3Schools.com</td>
</tr>
</table>
```

DON'T DO IT.


Copyright © 2011 W3Schools.com

Layout utilizando <div>

```
<html>
<body>

<div id="container" style="width:500px">

<div id="header"
style="background-color:#FFA500;">
<h1 style="margin-bottom:0;">
Main Title of Web Page</h1></div>

<div id="menu"
style="background-color:#FFD700;
height:200px;width:100px;float:left;">

<b>Menu</b><br />
HTML<br />
CSS<br />
JavaScript</div>
```

```
<div id="content" style="background-
color:#EEEEEE;height:200px;width:400px;
float:left;">
Content goes here</div>

<div id="footer" style="background-
color:#FFA500;clear:both;text-
align:center;">
Copyright © 2011 W3Schools.com</div>

</div>

</body>
</html>
```

http://www.w3schools.com/tags/tag_div.asp



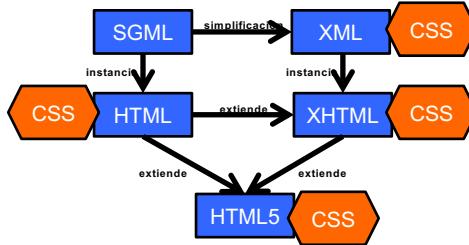
Más información

- <https://www.codecademy.com/tracks/web>
- <http://www.htmlcodetutorial.com/>
- Miles de libros
 - No hace falta que leáis todos!
- Como siempre, www.google.es ;)





Sistemas Informáticos



Tema 2.2: Web Hipertexto

El lenguaje CSS



Separación de responsabilidades

- Una regla fundamental del diseño de software es que funciones distintas deben ser llevadas a cabo por entidades distintas.
 - “*Separation of concerns*”
- HTML originalmente fue diseñado para definir los contenidos de documentos
 - <h1>This is a heading</h1>
 - <p>This is a paragraph.</p>
- Pero no para definir formato.
 - Cuando se agregaron etiquetas como (a partir de HTML 3.2) comenzó una pesadilla para los desarrolladores web.
 - Se mezcla el contenido (el qué) con el formato (el cómo)!!!



Cascade Style Sheets - CSS

- Usado para describir la semántica de presentación de un documento escrito en lenguaje de marcado
 - El uso más común es darle estilo a páginas escritas en HTML y XHTML.
- El objetivo es separar el contenido del documento (HTML o similar) de cómo se presenta (layout, colores, fuentes, etc.)
- Ventajas:
 - Ofrecer accesibilidad
 - Más control y flexibilidad de presentación
 - Compartir formato entre múltiples páginas
 - Distinta presentación para distintos dispositivos: ordenador, móvil, impreso, etc.



Heading 1

This is some text in a paragraph.

http://www.w3schools.com/Css/demo_default.htm

This is another paragraph.

Heading 2

Name	E-mail	Phone
Doe, John	jdoe@example.com	555-789-7222
Smith, Eva	esmith@example.com	555-324-3693

Heading 1

This is some text in a paragraph.

This is another paragraph.

Heading 2

Name	E-mail	Phone
Doe, John	jdoe@example.com	555-789-7222
Smith, Eva	esmith@example.com	555-324-3693

Heading 1

This is some text in a paragraph.

This is another paragraph.

Heading 2

Name	E-mail	Phone
Doe, John	jdoe@example.com	555-789-7222
Smith, Eva	esmith@example.com	555-324-3693

Heading 1

This is some text in a paragraph.

This is another paragraph.

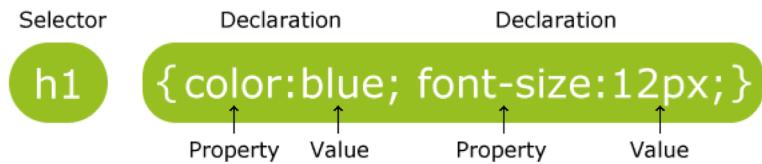
Heading 2

Name	E-mail	Phone
Doe, John	jdoe@example.com	555-789-7222
Smith, Eva	esmith@example.com	555-324-3693



Sintaxis CSS

- Una regla CSS tiene dos partes: un selector y una o más declaraciones



- El selector es normalmente un elemento HTML
- Cada declaración es una propiedad y un valor



Ejemplo CSS

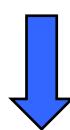
```
<html>
<head>
<link rel="stylesheet" type="text/css"
      href="ex2.css">
</head>

<body>
<h1>This is a header 1</h1>
<hr />

<p>You can see that the style
sheet formats the text</p>

<p><a href="http://www.w3schools.com"
target="_blank">This is a link</a></p>
</body>
</html>
```

```
body {background-color:tan;}
h1 {color:maroon;font-size:20pt;}
hr {color:navy;}
p {font-size:11pt;margin-left:15px;}
a:link {color:green;}
a:visited {color:yellow;}
a:hover {color:black;}
a:active {color:blue;}
```



This is a header 1

You can see that the style sheet formats the text
[This is a link](#)



Selector "id"

- Además de dar estilos a elementos HTML, CSS permite utilizar dos selectores propios: "id" y "class".
 - Combinan muy bien con <div>
- Selector "id":
 - Usado para especificar el estilo de un elemento con un id determinado.
 - Usa el atributo id de los elementos HTML, y se define con "#"
 - Ejemplo.
 - Cuidado: el ejemplo inserta el código CSS dentro del mismo fichero HTML.
 - Este método puede ser práctico cuando se quiere definir un estilo para una vez única.
 - Pero dificulta el mantenimiento de los ficheros.
 - No permite reutilizar el estilo en otros ficheros HTML



Ejemplo selector "id"

```
<html>
<head>
<style type="text/css">
#para1 {
text-align:center;
color:red;
}
</style>
</head>

<body>
<p id="para1">Hello World!</p>
<p>This paragraph is not affected by the style.</p>
</body>
</html>
```



Hello World!

This paragraph is not affected by the style.



El selector “class”

- Especifica el estilo de un grupo de elementos
 - En general se usa con muchos elementos.
- Permite especificar un estilo para todos los elementos HTML de una determinada clase
 - Utiliza el atributo *class* de HTML y se define con un “.”
 - Útil la etiqueta ** de HTML
 - Permite agrupar elementos y por sí sola no produce cambio visual. Ejemplo:

```
<p>My mother has <span class="blue">light blue</span> eyes.</p>
```

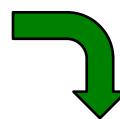
- Ejemplo: por simplicidad, otra vez utiliza el método de incluir el código CSS en el fichero HTML



Ejemplo selector “class”

```
<html>
<head>
<style type="text/css">
.center {
text-align:center;
}
</style>
</head>

<body>
<h1 class="center">Center-aligned heading</h1>
<p class="center">Center-aligned paragraph.</p>
</body>
</html>
```



Center-aligned heading

Center-aligned paragraph.



Combinaciones de selectores

- Finalmente, los selectores pueden combinarse para especificar una regla de selección en función de la relación entre selectores:
 - Espacio: selector de descendientes
 - >: selector de hijos
 - ~: selector de hermanos
 - +: selector de hermanos adyacentes
- Ejemplo
 - https://www.w3schools.com/css/css_combinators.asp



Resumiendo

- Tres formas de incorporar estilos (con CSS) a fichero HTML:
 - Hoja de estilos externa: **preferida** cuando los estilos se aplican a muchos ficheros (x ej, sitio entero).

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css" />
</head>
```

- Hoja de estilos interna: cambia el estilo a un único documento

```
<head>
<style type="text/css">
hr {color:sienna;}
p {margin-left:20px;}
body {background-image:url("images/back40.gif");}
</style>
</head>
```

- Estilos *inline*: directamente en el tag. Solo en casos muy específicos!

```
<p style="color:sienna;margin-left:20px">This is a paragraph.</p>
```



Cascada

- (lo que por fin explica el nombre)
- Como un determinado elemento (tag) puede tener estilos definidos en varios sitios, se impone un orden de precedencia que determina qué selector prevalece según tres criterios (de mayor a menor):
 - Importancia (!important)
 - Especificidad:
 - Estilo *inline*
 - Selector id
 - Selector class
 - Selector de elemento HTML
 - Orden

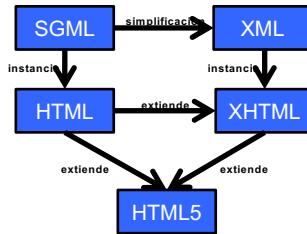


CSS – Más información

- <http://www.w3.org/Style/CSS/learning>
- <https://www.codecademy.com/tracks/web>
- Miles de libros (menos que HTML, pero no obstante ...)
- www.google.es
- Recomendado: a los usuarios de *Firefox* instalar las extensiones *Web developer* y *Firebug*



Sistemas Informáticos



Tema 2.2: Web Hipertexto

El lenguaje XML

Tema 2



61

Extensible Markup Language, XML

- Especificación del W3C para crear lenguajes descriptivos de propósito especial.
 - www.w3.org/XML
- Lenguaje de meta-marcado
- Diseñado para describir estructuras de datos.
- Utilizado para realizar intercambios de datos entre sistemas.
 - Intercambio de datos, gráficos vectoriales, serialización de objetos, XSL, etc.
 - Química, música, propiedades inmobiliarias, matemáticas, etc.
- Etiquetas, elementos y atributos con estructura de árbol



Extensible Markup Language, XML

- Las etiquetas de XML no están predefinidas. Se deben definir para cada caso concreto mediante una DTD o un esquema XML.
- Presenta los datos de forma legible por el usuario e interpretable por el ordenador de modo simultáneo
- Autodocumentado: Contiene simultáneamente la descripción de las estructuras de datos y sus valores.
- Su sintaxis estricta facilita su análisis y extracción de datos.
 - Parsers: SAX, DOM, JDOM
- **No comprimido.** Genera mensajes muy grandes comparado con otras técnicas.
- La manipulación de datos requiere el análisis del documento completo.
- Compatible con Web y otros protocolos de Internet.



Sintaxis XML

- El documento XML es de texto, normalmente Unicode.
- Cada documento XML debe tener un único elemento raíz.
- Todo elemento no vacío debe tener etiqueta de comienzo (`<elemento>`) y final (`</elemento>`).
- Todo elemento que no lleve contenido puede ser marcado con una etiqueta de elemento vacío (`<elemento/>`).
- Todos los valores de atributos deben ir entre comillas dobles.
- Los elementos se pueden anidar, pero no solapar con otros.
- Los nombres de los elementos son sensibles a mayúsculas y minúsculas.
- Permite el uso de distintos espacios de nombres en el mismo documento, tomados de distintos vocabularios (`namespace:nombre`).



Estándares relacionados

- XML procede de SGML
 - Standard Generalized Markup Language, IBM, 70's
 - Complejidad de SGML (150 páginas para su especificación)
- XHTML es una aplicación XML
 - HTML mezcla datos y formateo
 - XML se puede ver como una generalización de HTML que explicita estructuras de datos
- Aplicaciones XML: XSL, SVG, SMIL, MathML, RDF, SOAP...



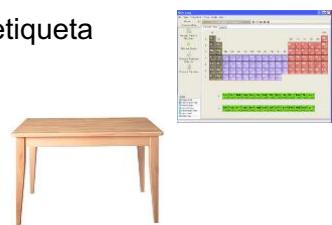
Validación con DTD

- Document Type Definition
- No es obligatorio
- Especifican un sublenguaje de XML
 - Lista de etiquetas permitidas
 - Etiquetas y atributos permitidos dentro de cada etiqueta
- Pueden ser muy complejos
 - Para SVG (Scalable Vector Graphics) más de 1000 líneas
 - Para XHTML 1.0 más de 1,500 líneas
 - Para DocBook más de 11.000 líneas
- Alternativa: XML Schema (versión 1.0, mayo 2001)



Extensible Stylesheet Language, XSL

- CSS dice cómo mostrar un documento HTML
 - HTML tiene un conjunto predefinido de etiquetas.
 - Decirle a un navegador cómo mostrar un elemento en una fuente o color especial es sencillo.
- XML en cambio permite definir cualquier etiqueta
 - ¿Qué significa TABLE?
 - ¿Cómo debe mostrarlo un navegador?
- XSL le dice al navegador cómo se debe mostrar el documento XML
- ES un documento XML



Ejemplo

Root
`<Person birth = "1912" death = "1954">` — Atributo
`<name>`
 `<firstName> Alan </firstName>` — Elemento
 `<lastName> Turing </lastName>`
 `</name>`
 `<profession> computer scientist </profession>`
 `<profession> mathematician </profession>`
 `<profession> cryptographer </profession>`
`</Person>`



Parsing de XML: DOM

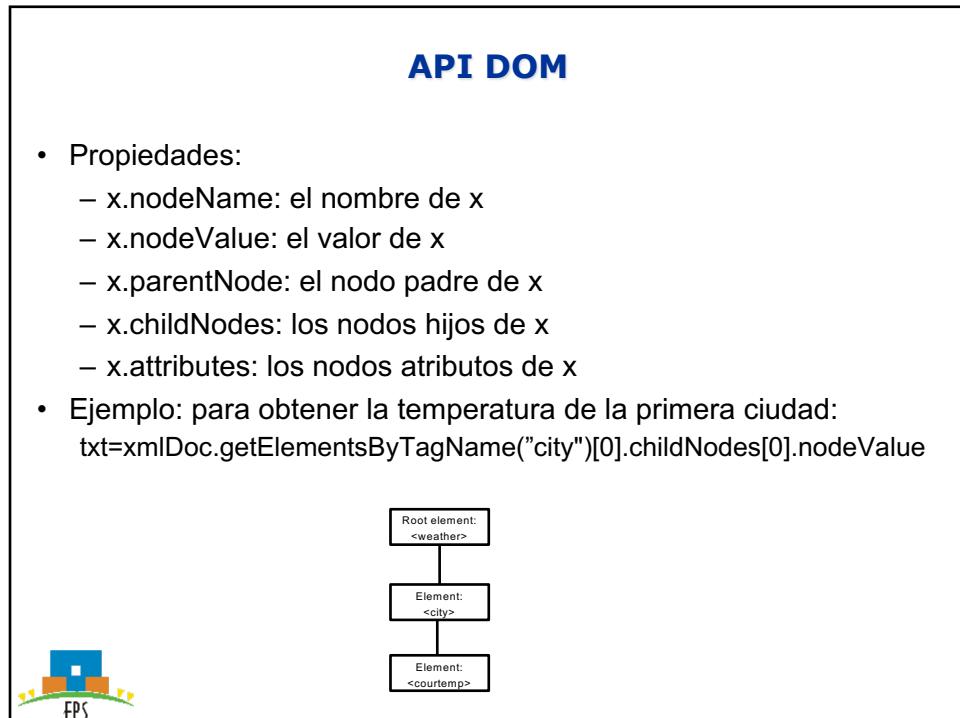
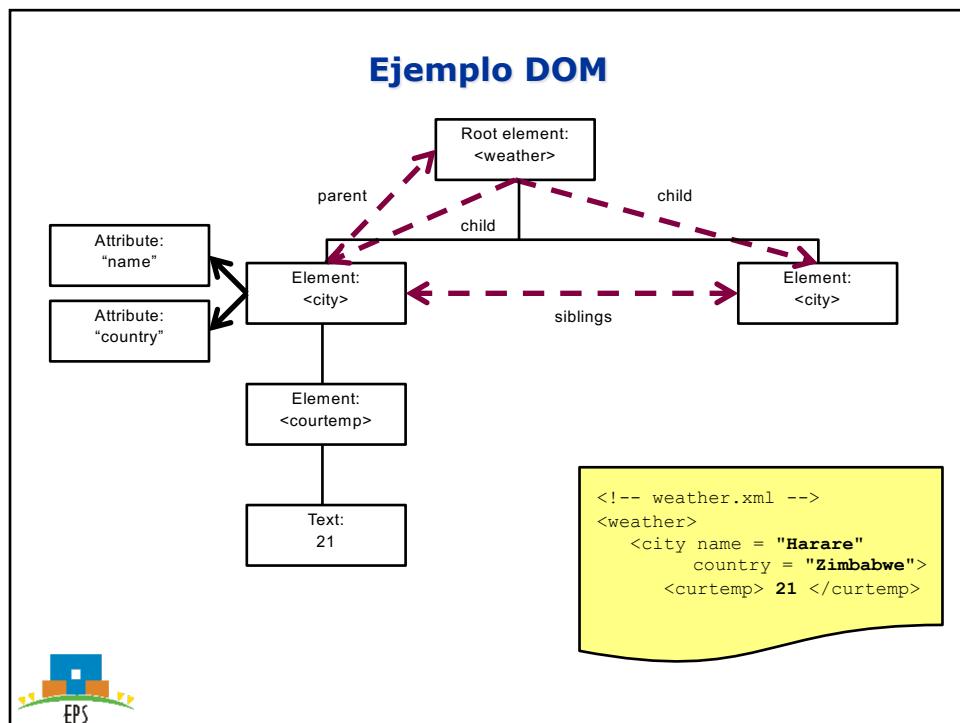
- Forma estándar para acceder y manipular documentos XML.
 - También sirve para otros lenguajes como HTML
- Estándar del W3C
- Los documentos son estructurados como árboles.
- 3 partes: Core DOM, HTML DOM y XML DOM
- XML DOM:
 - Modelo de objetos estándar para XML.
 - Una API estándar para XML
 - Independiente de la plataforma y del lenguaje de programación



Conceptos DOM

- Todo en DOM son nodos:
 - El documento completo es un *document node*
 - Cada elemento XML es un *element node*
 - El texto en elementos XML son *text nodes*
 - Cada atributo es un *attribute node*
 - Los comentarios son *comment nodes*





XPath

- Tiene 7 tipos de nodos:
 - *element, attribute, text, namespace, processing-instruction, comment, and document nodes.*
- Los documentos son tratados como árboles
 - El elemento superior es llamado raíz
- Relaciones entre nodos:
 - *Parent,*
 - *Children,*
 - *Siblings,*
 - *Ancestors,*
 - *Descendants.*



Ventajas y desventajas de DOM

- Fácil de usar
- Consume mucha memoria
 - Obliga a construir el árbol de todo el documento antes de empezar!
- Lento
 - Por lo mismo
- Más información:
 - <http://www.w3schools.com/dom/default.asp>



Parsing de XML: SAX

- Otro protocolo estándar para acceder documentos XML.
- Es orientado a eventos:
 - El usuario provee métodos *callback*, que son invocados por el parser a medida que lee el documento XML.
- Es rápido y eficiente, pero más difícil de usar que DOM.
 - Cada nodo es tratado independientemente de lo que había antes o después.
 - No se puede “volver atrás” en el procesado.



Sistemas Informáticos

Tema 2.2: Web Hipertexto

El lenguaje JSON



Características Principales

- Muy popular: es seguramente el formato más utilizado para intercambiar datos entre aplicaciones web.
- JSON es legible por personas e interpretable por algoritmos.
- JSON está basado en un subconjunto de JavaScript
 - su diseño se inspira en los objetos literales de JavaScript



¿Qué aspecto tiene un documento JSON?

```
{ "artists" : [ { "artistname" : "Deep Purple", "albums" : [ { "albumname" : "Machine Head", "year" : "1972", "genre" : "Rock" }, { "albumname" : "Stormbringer", "year" : "1974", "genre" : "Rock" } ] } ] }
```



Sintaxis JSON

- JSON almacena los datos en **objetos y arrays**.



Objetos JSON

- Un objeto JSON es un conjunto desordenado de pares nombre/valor, encerrado entre llaves ({}).

```
{ }

{ "artistname" : "Pink Floyd" }

{
  "artistname" : "Pink Floyd",
  "formed" : "1964"
}

{
  "artistname" : "Pink Floyd",
  "formed" : "1964",
  "origin" : "Cambridge, United Kingdom"
}
```



Sintaxis JSON

- En JSON, un nombre es una cadena (string). Su valor puede ser un objeto, un array, número, cadena, true, false o null.
- En los ejemplos anteriores, los valores son todos cadenas.



Arrays JSON

- Un array JSON es una colección ordenada de valores. Permite construir listas de valores.

```
{ "artists" : [ { "artistname" : "Pink Floyd", "formed" : "1964" }, { "artistname" : "Bruce Springsteen", "born" : "1949" }, { "artistname" : "U2", "formed" : "1976" } ] }
```



Datos anidados

```
{ "artists" : [ { "artistname" : "Pink Floyd", "formed" : "1964", "albums" : [ { "albumname" : "The Dark Side of the Moon", "year" : "1973", "genre" : "Rock" }, { "albumname" : "Wish You Were Here", "year" : "1975", "genre" : "Rock" } ] } ] }
```

