M4. LLENGUATGE DE MARQUES I SISTEMES DE GESTIÓ DE LA INFORMACIÓlínea horizontal  
ÀLVARO MOLINA

alvaro.molina@iesjoandaustria.org

**Estructuració del curs**

| UF 1. Programació amb XML (45h) |
| --- |
| UF 2. Àmbits d’aplicació de l’XML (27h) |
| UF 3. Sistemes de gestió empresarial (27h) |

**Avaluació**

**Sublime bracket o notepad++ al windows**

UF1. PROGRAMACIÓ AMB XML

header line

1.1 Història

* 1989, Tim Berners-Lee → el CERN el problema més gran que tenia era l’organització de la informació.
* Va crear **Information Management: A proposal;**  lo que en el futur sería el HTML.
* 1990. Poc desprès, es va crear el primer navegador web, on emmagatzemen aquests documents. (WorldWideWeb)
* 1993. El CERN ho va posar de domini públic, garantint la gratuïtat.
* Actualment, el navegador més utilitzat es el chrome.

# 1.1.1 Història dels navegadors

### L’hipertext i l’hipermèdia

Hipertext: conjunt estructurat de textos, gràfics, etc. units entre si per enllaços.

Multimèdia: imatges, sons, text (Inputs digitals no escrits)

Hipermèdia: Juntar qualsevol contingut multimèdia.

### El nom de domini

Quan es connectem a google, posem google no la seva IP.

Google: Nom del domini

IP: Adreça

*A la primera práctica, ens crearem un domini.*

### HTML i XML

Les principals etiquetes son: (mirar pwp)

* **HTML:**
  + Va ser desenvolupat a partir del SGML
  + Desde els 90’ ha anat evolucionant.

### Conceptes bàsics

Necessitarem un editor de textos per crear una pàgina web.

* No es pot trencar cap cosa.
* Si hi ha algún error, es pot tornar a configurar.

HTML i XML únicament **defineixen estructura i contingut.**

La base del llenguatge HTML son les **etiquetes**. <etiqueta> </etiqueta>.

L’**element** és el component bàsic d’una web.

* Etiqueta inicial
* Contingut
* Etiqueta de tancament

Les **etiquetes**:

* Poden portar atributs **només** en la etiqueta principal
* Poden ser un valor o diversos.
* No es mostren tal qual, defineixen el comportament de la pàgina.

L'estructura **correcta** es <!DOCTYPE html> defineix el contingut del fitxer (en aquest cas que es contingut HTML).

<html> etiqueta primera

<head> son les fitxes, miniatures..

<title> Dins la capçelera

<body> Contra tot el contingut

Totes les etiquetes s’han de **tancar** i amb l’ordre **correcte**.

[www.000webhost.com](http://www.000webhost.com) servidor web públic

→ Activitat 1

# 2. CSS

## 2.1 XHTML y CSS

Inicialmente, HTML era fácil de aprender y muy limitado en términos de marcas y estructura.

Todo cambió cuando aparecieron los primeros navegadores modernos, eran capaces de representar recursos gráficos y animaciones.

El número de sitios web comenzó a crecer y con él el número de etiquetas HTML. El objetivo principal era crear sitios web más atractivos, y para eso, HTML tenía que incluir nuevas etiquetas para los efectos visuales.

Para evitar que (X) HTML fuera el responsable de la parte estética y visual de la web, se diseñó el lenguaje CSS (Cascading Style Sheets).

Usar hojas de estilo (**stylesheet)** nos brinda los siguientes beneficios:

- Posibilidad de mantener claro el código HTML.

- Para el diseño, CSS es más poderoso que (X) HTML.

- CSS es un lenguaje simple.

- Las hojas de estilo se pueden reutilizar de diferentes documentos (X) HTML.

El gran problema de las hojas de estilo es que no todos los navegadores se comportan de la misma manera frente a la misma hoja de estilo; porque algunos navegadores no siguen los estándares establecidos, lo que obliga a los programadores a codificar diferentes hojas de estilo (una para cada navegador). Sin embargo, en los últimos años, los navegadores están más cerca de los estándares propuestos.

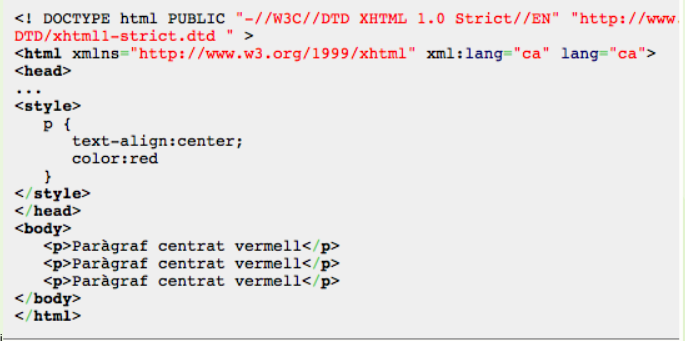
### Estilo XHTML:

- Dentro de la etiqueta XHTML, usando el atributo de estilo**. style attribute**



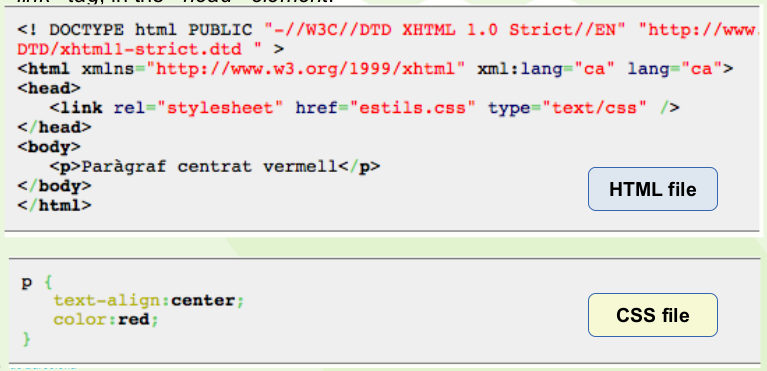
### CSS & HTML

- En el encabezado XHTML, agregando diferentes propiedades CSS dentro del elemento <style> ubicado en el elemento <head>.



### CSS

- En un documento externo, agregando todas las propiedades dentro del documento con extensión .css. Desde el documento HTML vinculamos esta hoja de estilo con la etiqueta <link>, en el elemento <head>.



### 

### 

### 

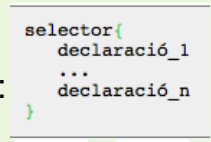
### 

### 

### 

### Reglas CSS

Sintaxis generica:



### AT-Rules

Hay un conjunto de reglas especiales llamadas **at-rules.**

Estas reglas se caracterizan porque comienzan con el carácter @.

#### @import:

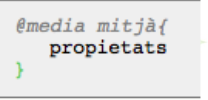
se utiliza para incluir **stylesheet** externas en nuestra hoja de estilo. Si, por ejemplo, queremos incluir en nuestra hoja de estilo todas las propiedades escritas en otra (mesestils.css), debemos escribir la siguiente línea:



#### @media:

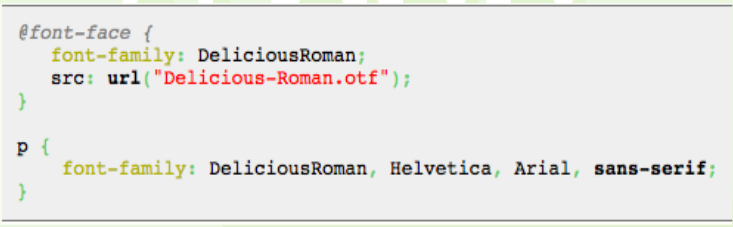
Se utiliza para indicar con qué dispositivo se ofrecerán las propiedades.

Debe ser un print (para imprimir en papel el contenido) o screen (en diferentes tamaños).



#### @font-face:

Especifica un nuevo tipo de fuente no incluido en el navegador que el usuario descargará.



font-family: la\_fuente;  
src: url(“link\_dela\_font”);

### Cascade & Inheritance

Dos de las características que hacen que las hojas de estilo sean tan poderosas son la **cascade** y la **inheritance**(herencia).

La cascada se refiere a la posible combinación de diferentes hojas de estilo.

La herencia se refiere a la capacidad de los elementos del documento HTML (X) para heredar propiedades de sus elementos predecesores.

### Cascade - Jerarquia propiedades

Si hay propiedades en conflicto, ¿qué propiedad tiene prioridad?

En CSS se ha establecido una jerarquía en las propiedades:

1) Propiedades en el **atributo** de estilo dentro de un elemento.

2) Propiedades en el elemento <**style**> del documento HTML (X).

3) Propiedades en una stylesheet **externa**.

4) Propiedades definidas por el **usuario**. Los navegadores permiten al usuario establecer diferentes propiedades de estilo.

5) Propiedades definidas por el **navegador**. Los navegadores tienen un estilo predeterminado para cada uno de los elementos HTML (X). Si no se ha establecido ninguna propiedad en los elementos anteriores, las propiedades del navegador son las predominantes

### Características Inheritance

* **No** todas las propiedades se heredan, esta característica se describe en la especificación CSS correspondiente.
* Si queremos **forzar** la herencia en una propiedad podemos ingresar el valor heredado.
* Si ponemos un nuevo valor en una propiedad, ese valor prevalece sobre el valor heredado.

## 2.2 Propiedades de formato básico

### Medidas

Algunas de las propiedades que podemos configurar en la hoja de estilo tienen una unidad de medida que indica el tamaño de la propiedad:

* Tamaños de fuente
* anchos (width)
* márgenes (margin)

Estas unidades pueden ser relativas o absolutas:

* unidades relativas: podemos medir con **em** y **%.**
* unidades absolutas: **px** (píxeles) **cm** (centímetros), **mm** (milímetros), **in** (pulgadas) o **pt** (puntos).

#### 

#### Relative mesures

* Si usamos la medida **px** (píxeles), una distancia definida **no** **cambia** si el usuario cambia el tamaño del texto en su **navegador** o si el usuario cambia el tamaño de la **ventana** del navegador, pero **sí** cambia si el usuario cambia el **resolución de la pantalla.**
* En cambio, una distancia definida en **em** o **porcentaje** es **proporcional** al tamaño de la ventana del navegador.

### Colores & fondo

* **color:** color de la lletra
* **background-color:** color de fons
* **background-image:** treu la imatge de fons, o la posa.
* **background-attachment:** Estableix si una imatge es fixa o es mou. fixed o scroll
* **background-repeat**: Estableix si la imatge de fons, volem que es repeteixi (repeat), o si no ho volem (no-repeat)
* **background-position:** Especifica la posició de la imatge de fons.
  + El primer valor es la X (left, center, o right)
  + El segon es la Y (top, center, bottom)
* **background:** permet ajuntar totes les propietats anteriors.



### Box Model - modelo de caja

Para entender este modelo tenemos que pensar en cada elemento del documento HTML (X) como si fuera una caja.

De cada una de estas casillas podemos distinguir los siguientes elementos:

* Content: El **contenido** del elemento: texto u otros elementos.
* Padding: El **acolchado**: espacio entre el contenido y el borde de la caja.
* Border: El **borde** (borde): borde que delimita el cuadro.
* Margin: El **margen**: espacio entre el borde de la caja y el resto de elementos.

Además, cada caja tiene cuatro segmentos: el superior (**top**), el inferior (**bottom**), el izquierdo (**left**) y el derecho (**right**).

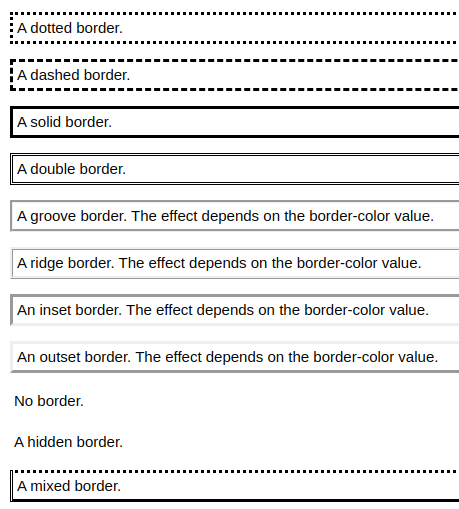


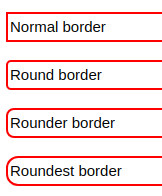
#### Propiedades Box

Resum de les més rares:

#### Border

* dotted - define un borde punteado
* dashed - define un borde discontinuo
* solid - define un borde sólido
* double - define un borde doble
* groove - define un borde acanalado 3D. El efecto depende del valor del color del borde.
* ridge - define un borde estriado en 3D. El efecto depende del valor del color del borde.
* inset - define un borde recuadro 3D. El efecto depende del valor del color del borde.
* outset - define un borde de inicio 3D. El efecto depende del valor del color del borde.
* none - no define ningún borde
* hidden - define un borde oculto



La propiedad border-radius se usa para agregar bordes redondeados a un elemento:

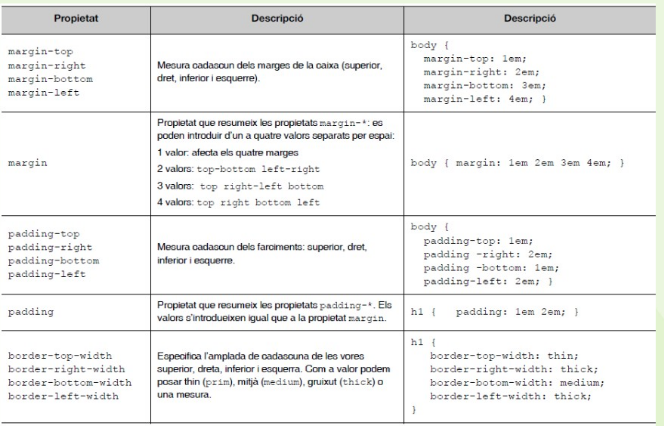
p {

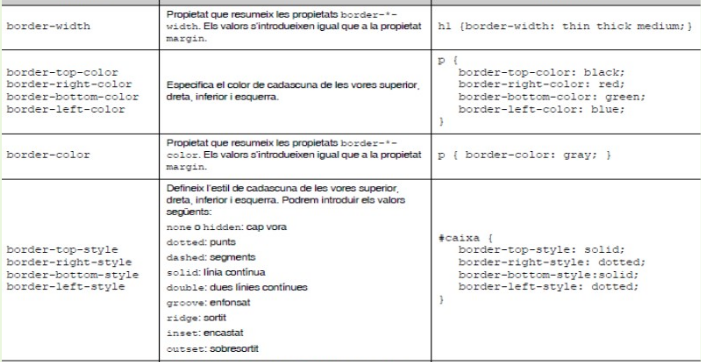
border: 2px solid red;

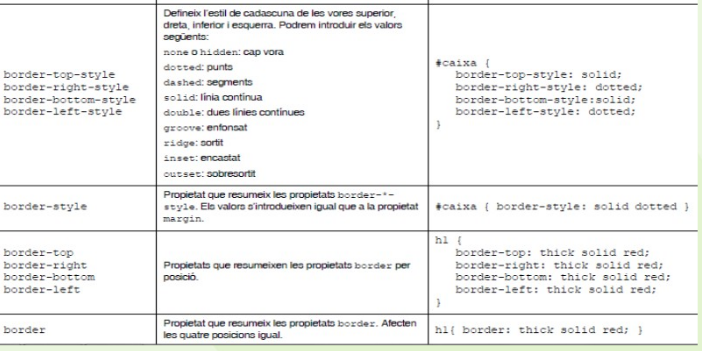
border-radius: 5px;

}

* **border-width**: el ancho de los bordes con valores **thin**(delgado), **medium**(medium), **thick**(gordo) o una medida.







### 

### 

### 

### 

### 

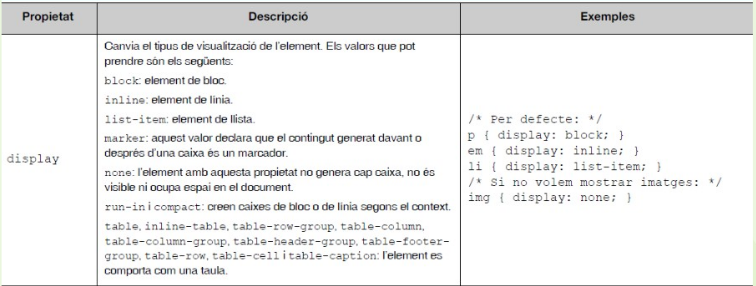
### 

### Formato visual de los elementos

Define cómo son los cuadros (es decir, los elementos HTML (X) vistos como cuadros) en el documento y el comportamiento de estos cuadros en relación con los otros cuadros de la página.

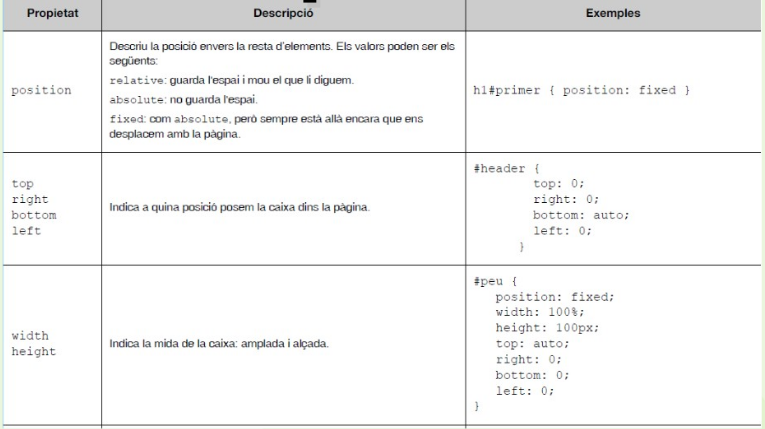
**display:**

* **block:** elemento de bloque
* **inline:** elemento en línea.
* **list-item:** elemento en lista
* **marker:** declara que el contenido delante o detras es un marcador.
* **none**: el elemento no genera una caja, no es visible ni ocupa espacio
* **run-in** i **compact:**  crean cajas de bloque o en línea según el contexto
* **table- :** se comporta como una tabla

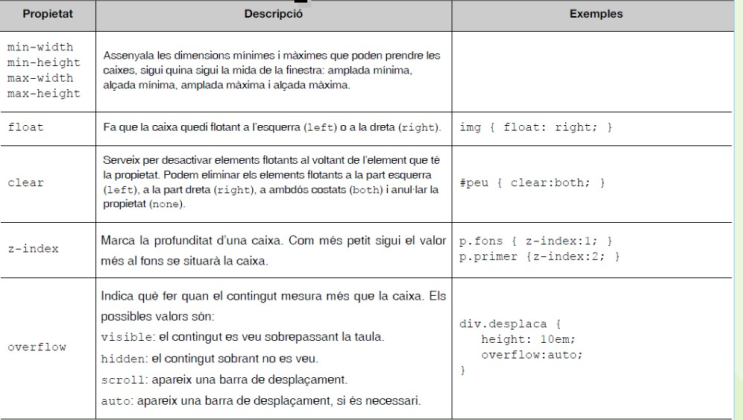


### Propiedades

* **position:** la posición respecto al resto de elementos.
  + **relative:** guarda el espacio y mueve el resto
  + **absolute:**  no guarda el espacio
  + **fixed:**  fijada en un sitio, aun moviendo la pagina.
  + **top,right,bottom,left:**  indica en que posicion se pone la caja en la pagina.
  + **width, height:** indica la medida de la caja

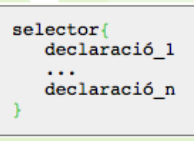


* **min o max-widht, height:** tamaño máximo o mínimo de ancho o alto de las cajas.
* **float:**  queda flotando a la derecha o izquierda (left, o right)
* **clear:**  sirve para desactivar elementos flotantes en una propiedad.
* **z-index:** marca la profundidad de una caja. (delante o atras)
* **overflow:**  indica que hacer cuando el contenido es mas grande que la caja.
  + **visible:**  el contenido sobrepasara la caja
  + **hidden:** no se vera
  + **scroll:**  aparece una barra de desplazamiento
  + **auto:**  automatico.



### Selectores

Un selector es la parte de la regla CSS que identifica el elemento o elementos del documento (X) HTML cuyas propiedades queremos especificar.

Hay diferentes tipos: 

* De tipo
* De clases
* Identificador
* Atributos
* Universal
* De niños
* De hermanos adyacentes

#### Tipo de selector

El elemento se selecciona mediante el **nombre de etiqueta** HTML (X). Las propiedades definidas afectan a todos los elementos HTML (X) con ese nombre de etiqueta.

h1{

color:blue;

}

#### Clases de selectores

Se utilizan para **asociar** **propiedades** a todos los elementos que tienen el **atributo de clase** de igual valor:

| <tag class="ValueClass">...</tag> | tag.**Value Class**{  declaraciones  } | **.ValueClass** {  declaraciones  } |
| --- | --- | --- |
|  | p{  color: green;  } | p.destacat{  color: red;  } |

#### Identificador de selectores

* Se utilizan para **asociar** propiedades con el elemento que tiene un determinado **identificador**.
* El identificador de un elemento HTML (X) se declara con el atributo **id** del elemento.
* Este identificador debe **ser único.**

<tag id="ValueId">...</tag>

| tag**#**ValueId { declaracions } | **#**ValueId {  declaracions  } |
| --- | --- |

#### 

#### Atributo de selectores

Se utilizarán para aplicar las propiedades de la regla a elementos que tengan un **atributo específico.**

| <tag attr="valor">...</tag> | tag[attr] {  declaracions  } |
| --- | --- |
|  | img[title] {  margin-right:10px;  } |

Para **más** **de** **un** atributo:

img [título] [clase = "preferencial"] {  
margen derecho: 10px;

Para el **valor** del atributo:

[attribute~="value"]

Se utiliza para seleccionar elementos que tienen u**na lista de palabras separadas por espacios** como atributo, uno de los cuales es exactamente valor.

| img[alt~="logo"] { border: solid } |
| --- |

[attribute|="value"]

Se utiliza para seleccionar elementos cuyo atributo **es una lista de palabras separadas por guiones,** comenzando con "valor".

| p[lang|="en"] {  font-family: "Times New Roman", Serif  } |
| --- |

#### 

#### Selector universal

Afecta a todos los items de un documento:

| \* {  declaraciones  } |
| --- |

#### Selector de descendientes

Nos permite seleccionar todos los elementos que están contenidos en otro.

| Tag TagDescendant {  declaracions  } |
| --- |
| p em{  color:blue;  } |

#### Selector de niños

* Un elemento es hijo de otro si es **descendiente de primer nivel.**
* Los selectores de niños nos permiten seleccionar niños para un elemento en particular.
* El símbolo que debemos usar es **>**.

| Tag>ChildTag  {  declarations  } |
| --- |
| body>p { font-weight: bold; } |
| div ol>li p { text-decoration : underline; } |

En el segundo caso, el selector afecta a todos los <p> elementos que descienden de un elemento <li>, donde el elemento <li> debe ser hijo (descendiente directo) de un elemento <ol> que debe descender de un elemento. <div>. y su puta madre que no entiendo na.

#### Selector de hermanos adyacentes

Nos permiten, dado un elemento, conferir propiedades a su hermano **inmediatamente después.**

El símbolo utilizado es +:

| Tag + TagAdjacentSibling {  declarations  } |
| --- |
| h1 + h2 { margin-top: -5mm } |

Si queremos reducir el espacio vertical entre un <h1> y un <h2> que lo sigue inmediatamente.

#### Agrupación de selectores

Si varios selectores tienen las mismas propiedades podemos agruparlos separando los selectores por una coma:

| selector1, selector2, selector3 {  declaracions  } |
| --- |
| h1, h2, h3 { font-family: sans-serif } |

### Pseudoclasses

**:first-child.** : Selecciona un elemento que sea el primer elemento secundario de otro elemento.

**:link.** : Le permite definir el estilo de los enlaces en nuestra página, cuando aún no se han visitado.

**:visited:** Le permite definir el estilo de los enlaces en nuestra página, cuando han sido visitados.

**:active.** Permite definir el estilo de los elementos cuando se activan (cuando hacemos clic en ellos).

**:hover.** Le permite definir el estilo de los elementos a medida que los revisamos.

**:focus.** Defina el estilo de los elementos cuando reciben el foco.

### Pseudoelements

**:first-line.** Defina el estilo de la primera línea del artículo.

**:first-letter.** Le permite agregar propiedades de estilo a la primera letra del elemento.

**:after.**  Le permite ingresar contenido al final del elemento. Debemos darle al elemento la propiedad contenta con el valor deseado.

**:before.** Le permite ingresar contenido al comienzo del elemento. Debemos darle al elemento la propiedad contenta con el valor deseado.

# 

## 

3. HTML

### Inicio

La primera línea <**! DOCTYPE html**> es la declaración del tipo de documento. Indica al navegador web que este documento es un documento HTML5. No distingue entre mayúsculas y minúsculas.

El elemento **<head>** es un contenedor para las etiquetas que proporciona información sobre el documento, por ejemplo, la etiqueta <title> define el título del documento.

El elemento <**body**> contiene el contenido real del documento (párrafos, enlaces, imágenes, tablas, etc.) que se representa en el navegador web y se muestra al usuario.

La etiqueta <**P**>, y la etiqueta <p> define lo mismo en HTML que es un párrafo.

### Elementos vacíos

Los elementos vacíos (también llamados elementos de cierre automático o nulos) no son etiquetas de contenedor, es decir, no puede escribir <hr> algún contenido </hr> o <br> algún contenido </br>.

Un ejemplo típico de un elemento vacío es el elemento <**br**>, que representa un **salto de línea.** Algunos otros elementos vacíos comunes son <img>, <input>, <link>, <meta>, <hr>, etc.

### Escribir comentarios en HTML

Los comentarios generalmente se agregan con el propósito de hacer que el código fuente sea más fácil de entender. Puede ayudar a otros desarrolladores (o a usted en el futuro cuando edite el código fuente) a comprender lo que estaba tratando de hacer con el HTML. Los comentarios no se muestran en el navegador.

Un comentario HTML comienza con <! - y termina con ->, como se muestra en el siguiente ejemplo:

<!-- This is an HTML comment -->

<!-- This is a multi-line HTML comment

that spans across more than one line →

### Tipos de elementos HTML

Los elementos se pueden colocar en dos grupos distintos: elementos de nivel de bloque y elementos de nivel en línea. Los primeros componen la estructura del documento, mientras que los segundos visten el contenido de un bloque.

Además, un elemento de bloque ocupa el 100% del ancho disponible y se representa con un salto de línea antes y después. Considerando que, un elemento en línea ocupará solo el espacio que necesita.

Los elementos de **nivel de bloque** más utilizados son <div>, <p>, <h1> a <h6>, <form>, <ol>, <ul>, <li>, etc.

Considerando que, los elementos de **nivel en línea** comúnmente utilizados son <img>, <a>, <span>, <strong>, <b>, <em>, <i>, <code>, <input>, <button>, etc.

### ¿Qué son los atributos?

Los atributos definen características o propiedades adicionales del elemento, como el ancho y el alto de una imagen. Los atributos siempre se especifican en la etiqueta de inicio (o etiqueta de apertura) y generalmente constan de pares de nombre / valor como nombre = "valor". Los valores de los atributos siempre deben ir entre comillas.

Además, se requieren algunos atributos para ciertos elementos. Por ejemplo, una etiqueta <img> debe contener los atributos src y alt. Echemos un vistazo a algunos ejemplos de usos de atributos:

Ejemplo\_

**<img src** = "images / smiley.png" width = "30" height = "30" alt = "Smiley">

<a href="https://www.google.com/" title="Search Engine"> Google </a>

<abbr title = "Lenguaje de marcado de hipertexto"> HTML </abbr>

<input type = "text" value = "John Doe">

En el ejemplo anterior, **src** dentro de la etiqueta <img> es un **atributo** y la ruta de la imagen proporcionada es su valor. De manera similar, **href** dentro de la etiqueta <a> es un atributo y el enlace proporcionado es su valor, y así sucesivamente.

### Atributos de propósito general

Hay algunos atributos, como id, título, clase, estilo, etc. que puede utilizar en la mayoría de los elementos HTML. La siguiente sección describe sus usos.

#### El atributo id

El atributo **id** se utiliza para dar un **nombre** o **identificador** **único** a un elemento dentro de un documento. Esto facilita la selección del elemento mediante CSS o JavaScript.

Ejemplo

<input type = "text" id = "firstName">

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

<p id = "infoText"> Este es un párrafo. </p>

#### El atributo de clase

Al igual que el atributo id, el atributo **class** también se utiliza para identificar elementos. Pero a diferencia de id, el atributo de clase **no tiene que ser único** en el documento. Esto significa que puede aplicar la misma clase a varios elementos en un documento, como se muestra en el siguiente ejemplo:

Ejemplo<input type = "text" class = "resaltar">

<div class = "box highlight"> Contenido </div>

<p class = "highlight"> Este es un párrafo. </p>

#### El atributo del título

El atributo de **title** se utiliza para proporcionar texto de asesoramiento sobre un elemento o su contenido.

Ejemplo

<abbr title = "Consorcio World Wide Web"> W3C </abbr>

<a href="images/kites.jpg" title="Haga clic para ver una imagen más grande">

<img src = "images / kites-thumb.jpg" alt = "kites">

</a>

#### El atributo de estilo

El atributo de estilo le permite especificar reglas de estilo CSS como color, fuente, borde, etc. directamente dentro del elemento. Veamos un ejemplo para ver cómo funciona:

Ejemplo

<p style = "color: blue;"> Este es un párrafo. </p>

<img src = "images / sky.jpg" style = "width: 300px;" alt = "Cielo nublado">

<div style = "border: 1px solid red;"> Algo de contenido </div>

### 

### 

### 

### 

### Crear saltos de línea

La etiqueta **<br>** se utiliza para insertar un salto de línea en la página web.

Dado que <br> es un elemento vacío, no hay necesidad de la etiqueta </br> correspondiente.

Ejemplo

<p> Este es un párrafo <br> con salto de línea. </p>

<p> Este es <br> otro párrafo <br> con saltos de línea. </p>

### Crear reglas horizontales

Puede utilizar la etiqueta <**hr**> para crear reglas o líneas horizontales para separar visualmente secciones de contenido en una página web. Como <br>, la etiqueta <hr> también es un elemento vacío. He aquí un ejemplo:

<p>This is a paragraph.</p>

<hr>

<p>This is another paragraph.</p>

### Gestionar espacios en blanco

Normalmente, el navegador mostrará los múltiples espacios creados dentro del código HTML presionando la tecla de barra espaciadora o la tecla de tabulación en el teclado como un solo espacio. Los saltos de línea múltiples creados dentro del código HTML al presionar la tecla Intro también se muestran como un solo espacio.

Los siguientes párrafos se mostrarán en una sola línea sin ningún espacio adicional:

Ejemplo

<p> Este párrafo contiene varios espacios en el código fuente. </p>

<p>

Este párrafo

contiene varias pestañas y saltos de línea

en el código fuente.

</p>

Insertar &nbsp; para crear espacios consecutivos adicionales, mientras que insertar etiqueta <br> para crear saltos de línea en sus páginas web.

### Sintaxis del enlace HTML

Los enlaces se especifican en HTML mediante la etiqueta **<a>.**

Un enlace o hipervínculo puede ser una palabra, un grupo de palabras o una imagen.

<a href="url"> Texto del enlace </a>

Cualquier cosa entre la etiqueta de apertura <a> y la etiqueta de cierre </a> se convierte en la parte del vínculo que el usuario ve y hace clic en un navegador.

Ejemplo

<a href="https://www.google.com/"> Búsqueda de Google </a>

<a href="https://www.tutorialrepublic.com/"> Tutorial Republic </a>

<a href="images/kites.jpg">

<img src = "kites-thumb.jpg" alt = "kites">

</a>

El atributo **href** especifica el destino del enlace. Su valor puede ser una URL absoluta o relativa.

Una URL absoluta es la URL que incluye todas las partes del formato de la URL, como el protocolo, el nombre de host y la ruta del documento, por ejemplo, https://www.google.com/, https://www.example.com /form.php, etc. Mientras que las URL relativas son rutas relativas a la página, por ejemplo, contact.html, images / smiley.png, etc. Una URL relativa nunca incluye el prefijo http: // o https: //.

#### Establecer los destinos de los enlaces

El **atributo** de **destino** le dice al navegador **dónde** abrir el documento vinculado. Hay cuatro objetivos definidos, y cada nombre de objetivo comienza con un carácter de subrayado (\_):

**\_blank:** abre el documento vinculado en una nueva ventana o pestaña.

\_**parent:** abre el documento vinculado en la ventana principal.

**\_self:** abre el documento vinculado en la misma ventana o pestaña que el documento de origen. Este es el valor predeterminado, por lo que no es necesario especificar explícitamente este valor.

**\_top:** abre el documento vinculado en la ventana completa del navegador.

#### Crear anclajes de marcadores

También puede crear anclajes de marcadores para permitir que los usuarios salten a una **sección específica** de una **página web.** Los marcadores son especialmente útiles si tiene una página web muy larga.

La creación de marcadores es un proceso de dos pasos: primero **agregue el atributo id** en el elemento al que desea saltar, luego use el valor del atributo id precedido por el signo de almohadilla (#) como el valor del atributo href de la etiqueta <a> , como se muestra en el siguiente ejemplo:

Ejemplo

<a href="#sectionA"> Vaya a la sección A </a>

<h2 id = "sectionA"> Sección A </h2>

#### Creación de enlaces de descarga

También puede **crear el enlace de descarga** del archivo exactamente de la misma manera que colocando enlaces de texto. Simplemente apunte la URL de destino al archivo que desea que esté disponible para descargar.

En el siguiente ejemplo, hemos creado los enlaces de descarga para archivos ZIP, PDF y JPG.

Ejemplo

<a href="downloads/test.zip"> Descargar archivo zip </a>

<a href="downloads/masters.pdf"> Descargar archivo PDF </a>

<a href="downloads/sample.jpg"> Descargar archivo de imagen </a>

### Formatear texto con HTML

HTML proporciona varias etiquetas que puede usar para hacer que el texto de sus páginas web aparezca de manera diferente al texto normal, por ejemplo, puede usar la etiqueta

* <**b**> para poner el texto en **negrita**,
* la etiqueta <**i>** para poner el texto en **cursiva**,
* etiqueta <**mark**> para **resaltar** el texto,
* etiqueta <**code**> para mostrar un fragmento de código de computadora
* etiquetas **<ins> y <del>** para marcar **inserciones** y **eliminaciones** editoriale

#### Dar formato a las citas

Puede formatear fácilmente los bloques de citas de otras fuentes con la etiqueta HTML <**blockquote**>.

Las citas en bloque generalmente se muestran con márgenes derecho e izquierdo con sangría, junto con un poco de espacio adicional agregado arriba y abajo.

<blockquote>

<p> Aprende del ayer, vive el hoy, espera el mañana. Lo importante es no dejar de cuestionar. </p>

<cite> - Albert Einstein </cite>

</blockquote>

#### Mostrar abreviaturas

Una abreviatura es una forma abreviada de una palabra, frase o nombre.

Puede utilizar la etiqueta <**abbr**> para indicar una abreviatura. El atributo de título se usa dentro de esta etiqueta para proporcionar la expansión completa de la abreviatura, que los navegadores muestran como información sobre herramientas cuando el cursor del mouse se desplaza sobre el elemento.

Ejemplo

<p> El <abbr title = "World Wide Web Consortium"> W3C </abbr> es la principal organización de estándares internacionales para la <abbr title = "World Wide Web"> WWW o W3 </abbr>. Fue fundada por Tim Berners-Lee. </p>

#### Marcado de direcciones de contacto

Las páginas web a menudo incluyen direcciones postales o de calles. HTML proporciona una etiqueta especial <**address**> para representar la información de contacto (física y / o digital) de una persona, personas u organización.

Esta etiqueta debería usarse idealmente para mostrar información de contacto relacionada con el documento en sí, como el autor del artículo. La mayoría de los navegadores muestran un bloque de direcciones en cursiva.

Ejemplo

<address>

Fundación Mozilla <br>

331 E. Evelyn Avenue <br>

Mountain View, CA 94041, EE. UU.

</address>

<https://www.tutorialrepublic.com/html-tutorial/html-images.php>

### Tablas

4. XML

Utilitzar sempre el mateix Chartset. UTF-8

# 1. La codificació o el conjunt de caràcters (Charset)

### A. Dades binàries:

* En un ordinador, tota la informació és un número binari.
* És possible que trobem moltes possibilitats de codificació per a imatges, vídeos, text ... i fitxers en general.
* Hem de conèixer el programari de descodificació per accedir a la informació dins dels fitxers.

### B. Dades de text:

* Codi ASCII. El 1967. 7 bits. 128 símbols. Alfabet anglès i personatges especials.
* ASCII ampliat. 8 bits. 128 (ASCII) + 128 a un altre llengües personatges.
* Una taula ASCII ampliada per zona mundial Ampliada: 8869-1 (Europa Occidental), 8859-2 (Europa Central i Oriental), etc.

### C. **UNICODI**:

* Incloeu tots els caràcters d'idiomes.
* Tots els personatges superen un byte.
* Els primers 256 caràcters provenen de la taula ISO-8859-1.
* 3 sistemes de codificació:
  + **UTF-8**. Tots els caràcters ocupen entre 1 i 4 caràcters. 1 caràcter per taula ASCII, 2 per llengües llatines, 3 per caràcters japonesos i xinesos i 4 per símbols matemàtics o llengües de la mort.
    - **Codificació més comuna.**
  + **UTF-16**. Tots els caràcters ocupen 2 o 4 caràcters.
  + **UTF-32**. Tots els caràcters ocupen 4 caràcters. Gairebé no s’utilitza

### D. Quin és el millor conjunt de caràcters per evitar problemes?

Si és possible, utilitzeu sempre **UTF-8 sense BOM.**

Si no és possible, un altre recomanat és: **ISO-8859-1,** també conegut com a **Latin-1**.

Després **d’UTF-8**, aquesta és la millor opció.

### E. UTF-8 sense BOM

BOM Byte Order Mark

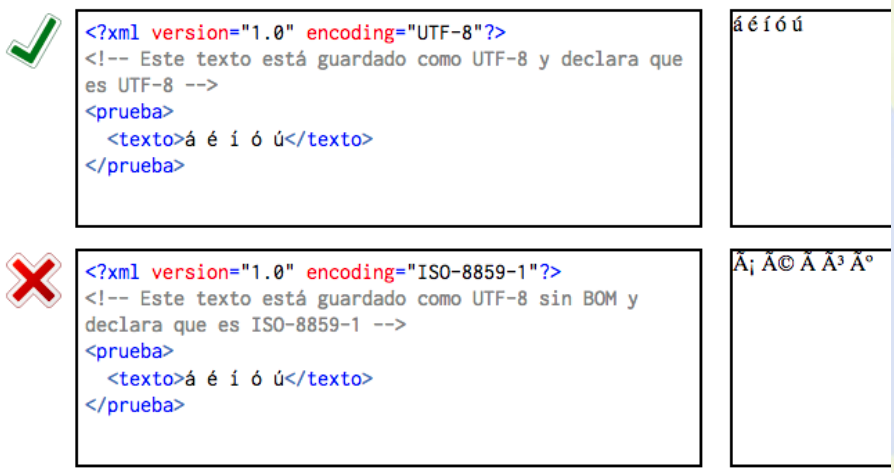
El BOM és un caràcter **unicode** amb una mida de **3 bytes** i s’utilitza per indicar l’ordre de bytes intern al començament del text del fitxer.

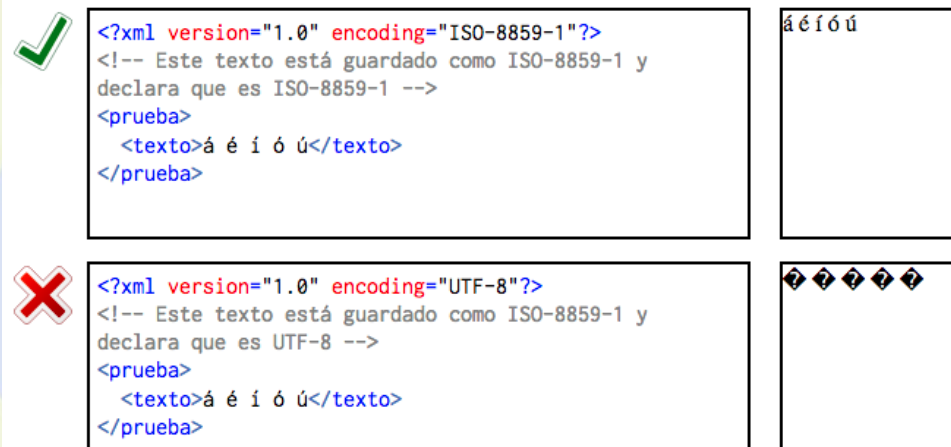
La millor opció és utilitzar **UTF-8**, però per defecte té el BOM, i alguns programes no llegeixen correctament aquests 3 bytes, com per exemple PHP5 i versions anteriors. En aquests casos, hem de tenir cura i fer servir UTF-8 sense BOM, o bé alguns dels nostres serveis web no es mostraran correctament.

Per escriure correctament els nostres fitxers, hem d'aplicar l'UTF-8 sense codificació BOM a **l'editor** (no només declarar-lo al codi).

Per comprovar si un fitxer té BOM, l’hem d’obrir amb un editor hexadecimal i parar atenció a la primera línia per veure:

ef bb bf -------- ï » ¿

Guardar el ISO-8859-1 com UTF-8 sense BOM -> No funciona 



Guardar un UTF8 com ISO-8859-1 no funciona.

## 1.1 Codificació o charset

Com podem indicar el conjunt de caràcters en un fitxer HTML?

Amb l'etiqueta <meta> (dins de la secció <head>).

Com aquests

| <meta http-equiv=''Content-Type'' content=''text/html; charset=''utf-8''/>  <meta http-equiv=''Content-Type'' content=''text/html; charset=''utf-8''>  <meta http-equiv=''Content-Type'' content=''text/html; charset=''utf-8''></meta> |
| --- |

Però, per als fitxers XML, la declaració de codificació hauria d'aparèixer a la **primera etiqueta com a atribut**, com aquest exemple:

| <?xml version=''1.0''encoding=''UTF-8''?> |
| --- |

# 2. XML

* XML és un subgrup SGML (Standard Generalized Mark-up Language), simplificat i adaptat a Internet.
* XML no és un llenguatge específic.
* XML és un **meta-llenguatge** que permet a l'usuari definir els seus propis llenguatges de marcatge per a usos específics.

#### Què no és XML?

* No és una millora de l'HTML.
* No és un idioma per crear pàgines web noves.
* No és difícil d’entendre.

#### Per què XML?

* És un estàndard d'ús internacional.
* No és un producte de l'empresa.
* Permet un ús eficaç d'Internet per a tots els programes.

#### És útil?

**Per mostrar informació estructurada** a través de navegadors web (tots els documents), permetent emmagatzemar, transmetre, processar, visualitzar, imprimir ... per diferents aplicacions i dispositius.

## 2.1 Estructura XML

En un fitxer XML podem trobar els components següents:

* **Elements**: peça bàsica en el llenguatge de marques, representada com en HTML. Pot existir elements buits (<br/>) i els elements poden contenir atributs.
* **Instruccions**: comandes especials per al navegador. Comencen pels personatges <? i en per?>.

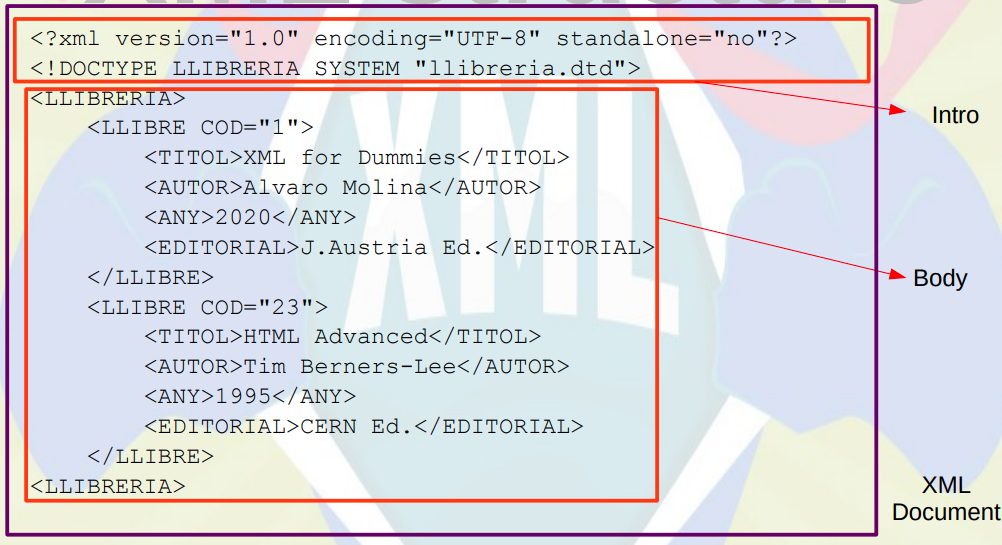
| <? xml-stylesheet type = "text / css" href = "estil.css"?> |
| --- |

* **Comentaris:** informació no llegida pels navegadors. Comenceu amb **<!** - i acabeu amb **->**.
* **Declaracions**: especificació de la informació del document:

| <! DOCTYPE persona SYSTEM "persona.dtd"> |
| --- |

* **Seccions** **CDATA**: obre una secció que els navegadors no llegeixen:

| <! [CDATA [Aquí és pot ficar qualsevol tipus de caràcter, com <, &, ->, ... Sense que siguin interpretats com a marcació]]> |
| --- |

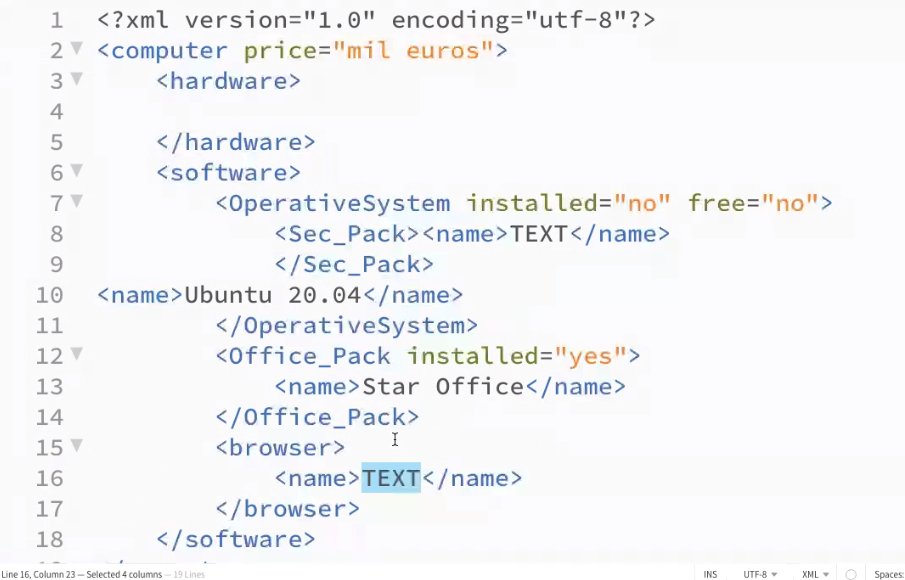


### Validació i ben format:

Un document XML pot:

* Estar **ben format (bona estructura)**: segueix la sintaxi XML. (Obrir o tancar)
* O **ser vàlid**: significa que el document està ben format i segueix regles específiques escrites en un fitxer DTD o XMLSchema.

Per **comprovar** si el nostre XML està ben format, només cal obrir-lo en un navegador o utilitzar una eina específica que ens pugui ajudar amb aquesta tasca.



5. GRAMMARS

# 5.1 Grammars & DTD’s

Las gramáticas XML y HTML son un conjunto de reglas para definir qué es correcto y qué no es correcto.

Si se selecciona una gramática (como HTML5), es muy recomendable seguir sus reglas.

* En HTML 5, esta es una página web válida:



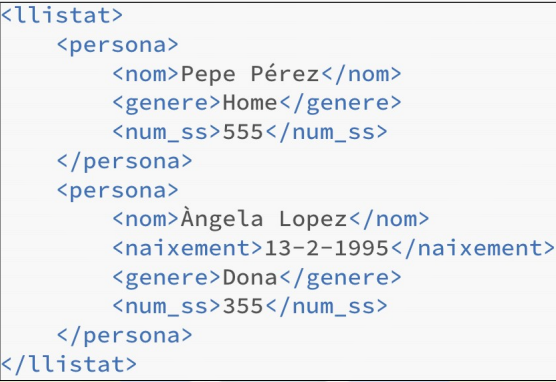
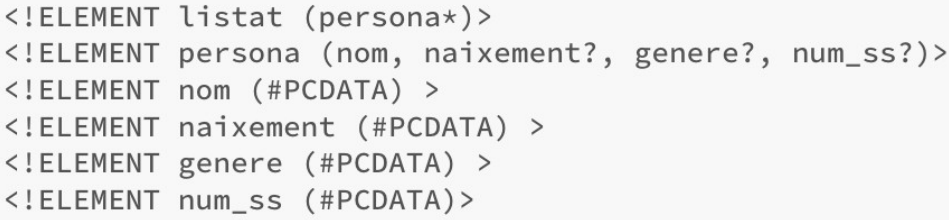
* En HTML 5, esto no es correcto:



Durante los últimos años, es posible que encontremos diferentes versiones de HTML (con distintas gramáticas). Estos documentos deben tener en su cabeza la indicación estándar especifica que sigue el código.

* **HTML4**: muy tolerante, dificulta la interpretación por parte de los navegadores.
* **XHTML**: es HTML con las estrictas reglas XML. Este hecho ayudó al desarrollo y permitió la expansión de Internet.
* **HTML5**: una revisión XHTML con mejoras como las etiquetas <àudio> y <video> (definición del código en la etiqueta). Ayuda mucho a la integración de JavaScript.

# 5.2 DTD : <https://www.abrirllave.com/dtd/entidades-generales-internas-analizables.php>



# 5.3 XML SCHEMA

Ha sido desarrollado y recomendado por W3C.

* Define el contenido y la estructura XML.
* Nos permite definir nuestro propio lenguaje de etiquetas basado en XML.
* Incluye muchas mejoras contra DTD.
* Los más usados y populares en realidad.

Para crear este tipo de archivos, podemos usar nuestro editor favorito o herramientas en línea como estas:

[www.utilities-online.info/xsdvalidation/](http://www.utilities-online.info/xsdvalidation/) <https://www.freeformatter.com/xml-validator-xsd.html> <https://www.topbestalternatives.com/best-xml-editor-software/>

## A.- INDICACIONES GENERALES

* Un XML-Schema sigue las reglas del XML pero:
  + El elemento raíz siempre: <schema>
  + Es obligatorio usar el espacio de nombre dentro de los elementos.
  + Los archivos se almacenarán con esta nueva extensión: .xsd

## B.- NAME SPACE

Es la dirección URL o URN que contiene el esquema DTD, que se utiliza para validar los elementos del documento XML.

* Evita conflictos en el documento XML.
* Existen dos formas de declarar el espacio de nombres, como predefinido o explícito.

**Predefinido**

* El espacio de nombres se declara como un nombre no calificado, sin prefijo:

| xmlns = "http: // ....." <! - declaració amb URL -> xmlns = "urn: ...." <! - declaració amb URN -> |
| --- |

Un nombre no calificado hace referencia a los elementos sin prefijo: <elemento>

**Explicito**

El espacio de nombres se ha declarado mediante un prefijo. Los elementos y atributos deben usar el nombre completo:

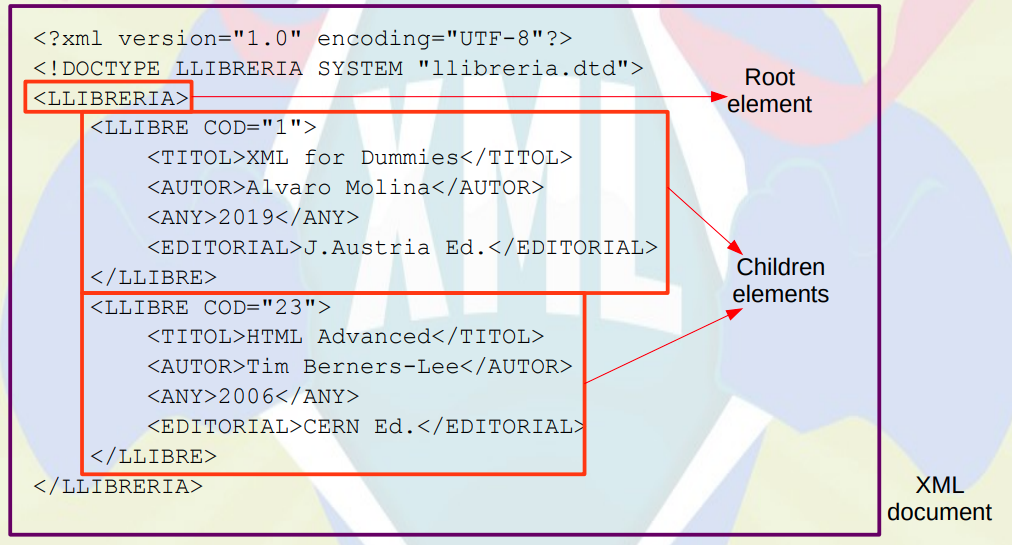
| xmlns: prefix = "http: // ..." xmlns: prefix = "urn: ..." |
| --- |

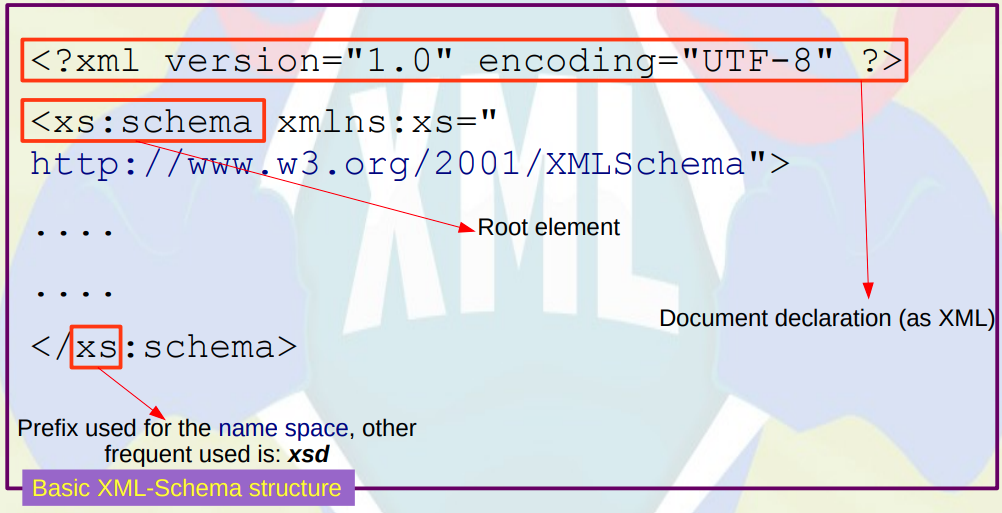
Los nombres completos usan el prefijo: <prefijo: elemento>

## 

## 

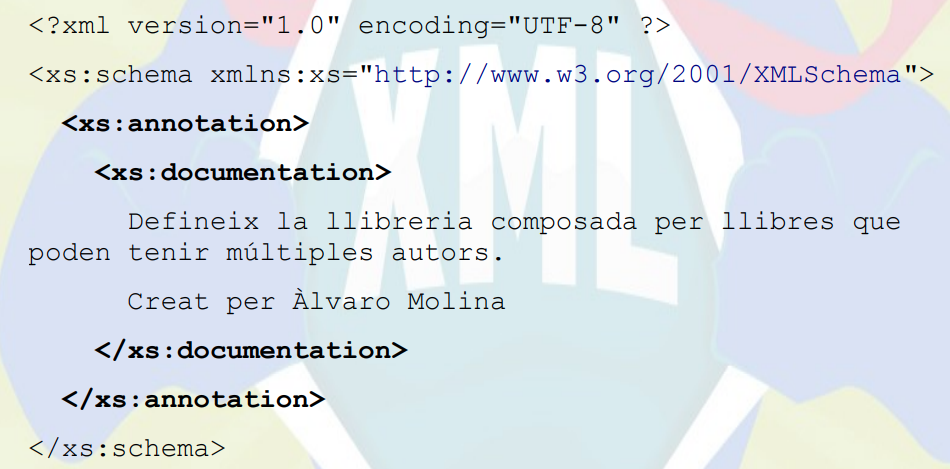
## C. ESTRUCTURA SCHEMA

****

****

XML-Schema permite dos formas de insertar **comentarios**:

* SIEMPRE dentro del elemento: xs:annotation
  + xs:documentation→ para agregar información sobre el archivo (datos de creación, derechos de autor ... comentarios para los usuarios ...).
  + xs: appinfo → orientado a transferir información adicional a las aplicaciones que usarán el esquema.
* Marcas de comentarios comunes generales: - <! - Comentari ->

****

## D. DATA TYPES

Para definir un elemento XML usamos:

| <xs: element name = ""> </ xs: element> |
| --- |

**Elemento simple**: solo datos de contenido:

| <xs: element name = "tel" type = "xs: integer" /> |
| --- |

**Elemento complejo**: puede contener atributos, elementos dentro o mezclar contenido.

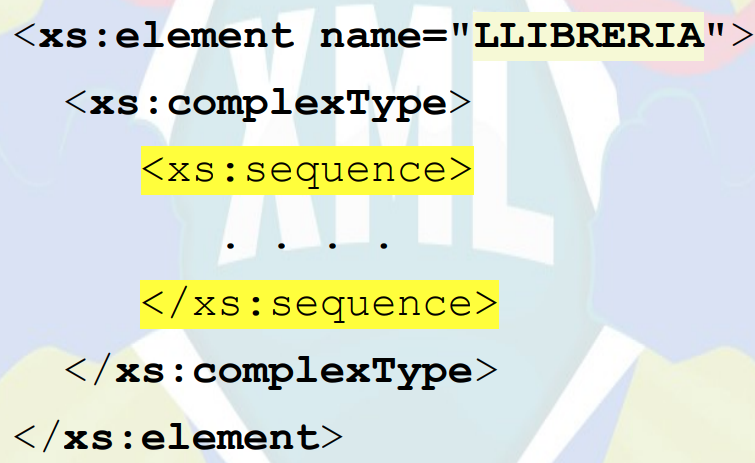
| <xs: nombre del elemento = "persona">  <xs: complexType>  </ xs: complexType> </ xs: elemento> |
| --- |

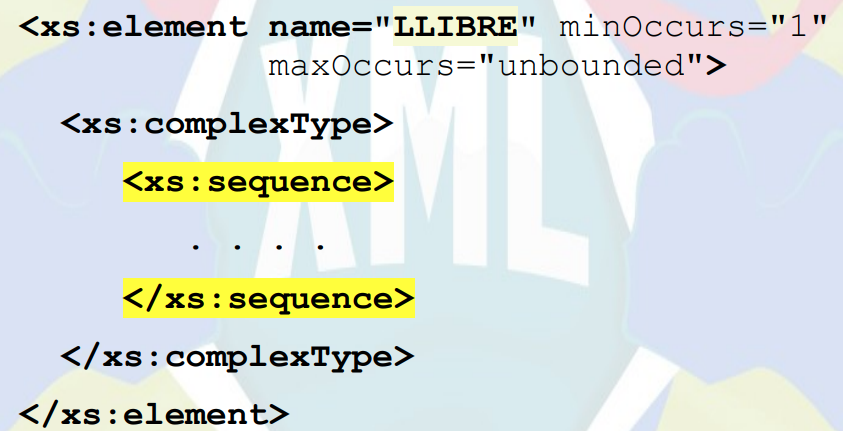
Opción <xs: sequence />:

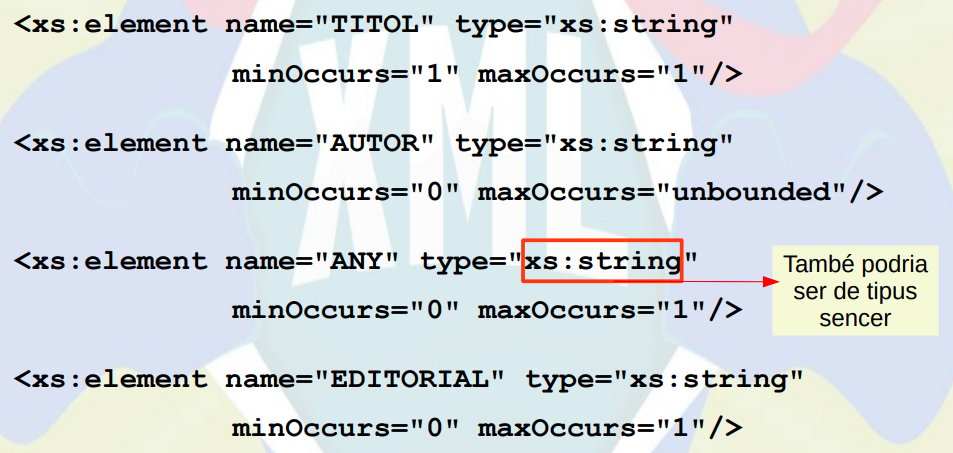
* Define una secuencia ordenada de elementos en el XML.
* Permite definir también: minOccurs, maxOccurs

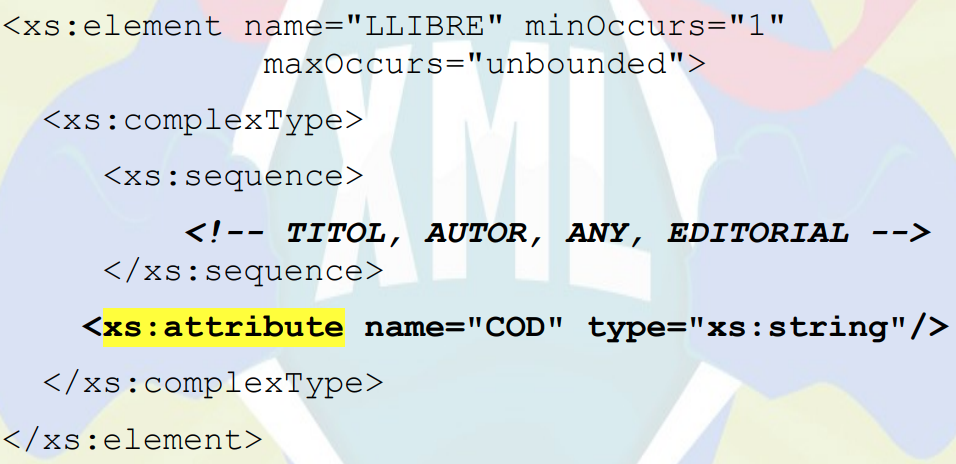
Opción <xs: choice />: define un grupo de elementos excluyentes, lo que permite que también defina un mínimo y un máx.

Opción <xs: all />: define un grupo de elementos no ordenados, cada elemento puede aparecer (en el XML) solo una o cero veces.









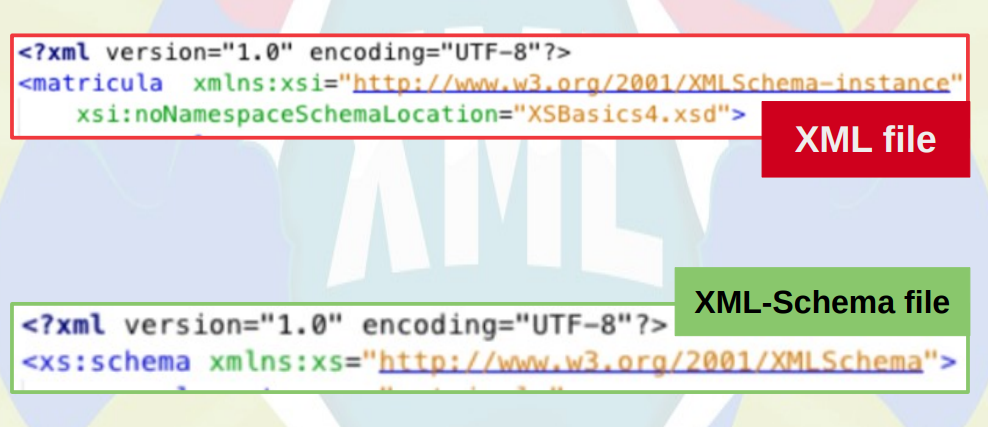
## 

## 

## 

## 

## E. COMO LINKAR

****

## F. NORMAS PARA ATRIBUTOS

Solo pueden aparecer después de la declaración de subelementos.

* ¡No tiene ningún elemento secundario!
* Son como elementos simples (una línea).
* Su declaración no define su orden de uso.
* Su tipo predeterminado es anySimpleType; representa cualquier carácter permitido en el XML.
* Un atributo solo puede aparecer una vez declarado.
* De forma predeterminada, los atributos son opcionales en el XML (si no existe un definición opuesta).
* Existen 3 opciones adicionales para definirlos: use, default i fixed.

| ● **use** |
| --- |

- required → uso obligatorio en el XML.

- opcional → uso opcional en XML.

- prohibited → uso no permitido en XML.

| <xs: atributo name = "lang" type = "xs: string" use = "required" /> |
| --- |

| ● **default** |
| --- |

Valor predeterminado del atributo.

Si el atributo no está presente en el XML, la aplicación tomará este valor.

| <xs: attribute name = "lang" type = "xs: string" default = "EN" /> |
| --- |

| ● **fixed** |
| --- |

- Fijar valor para un atributo.

- El atributo puede aparecer o no, pero si está presente, debe contener su valor.

| <xs: attribute name = "lang" type = "xs: string" fixed = "EN" /> tiempo declarado. |
| --- |

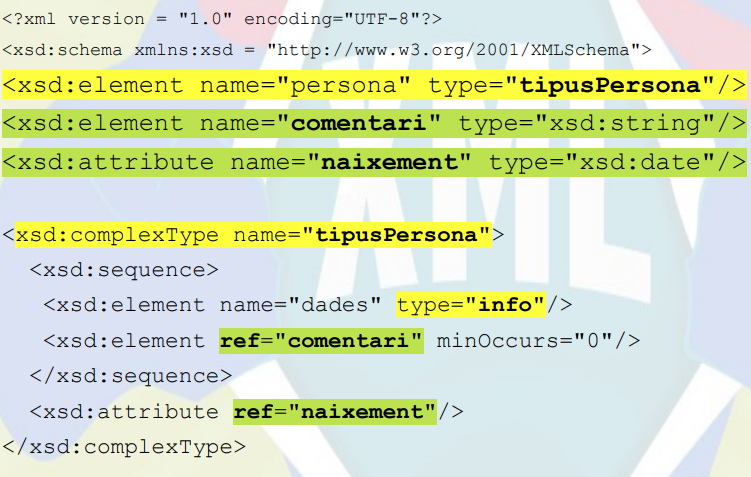
Todas las declaraciones (elementos y atributos) pueden ser:

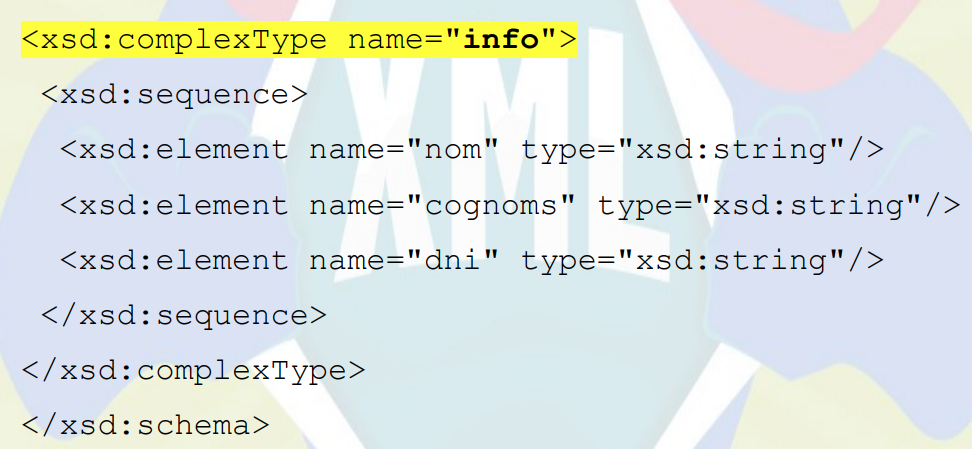
| ● **Global** |
| --- |

- Hijos directos de: <xs: schema />

- Otro tipo complejo puede usar esta declaración con una referencia.

Estos elementos y declaraciones de atributos pueden ser reutilizados por otros elementos o atributos, reduciendo mucho los códigos de línea y ayudando a la comprensión del archivo.





| ● **Local** |
| --- |

- Declaraciones dentro de los elementos (como el ejemplo en lides anteriores).