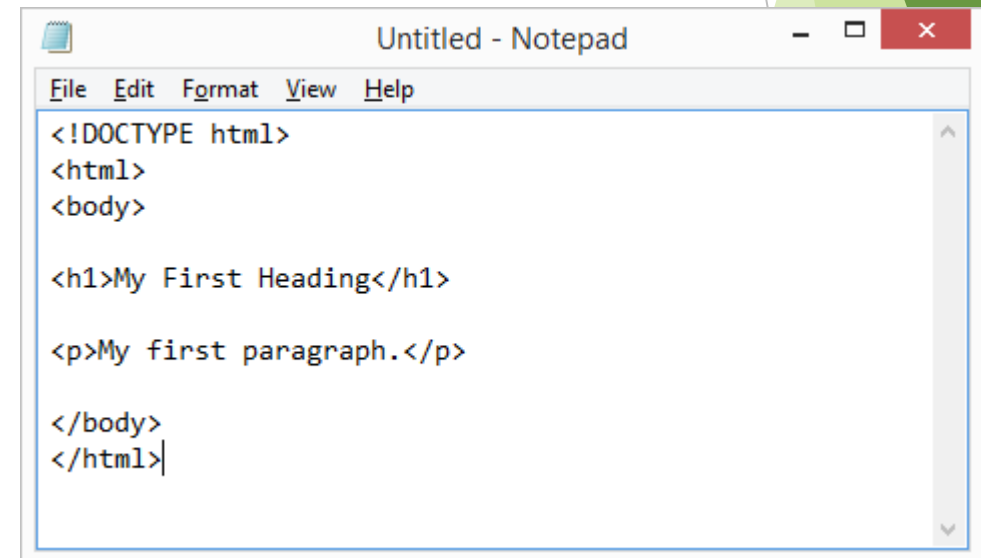


The background features abstract, overlapping green geometric shapes, primarily triangles and polygons, in various shades of green, creating a modern, layered effect. The shapes are concentrated on the left and right sides of the frame, leaving a central white area.

XML

Què és un llenguatge de marques?

- ▶ un llenguatge en el qual les seves parts es diferencien entre sí mitjançant senyals. És una forma de codificar un document que, juntament amb el text, incorpora etiquetes o marques que contenen informació addicional sobre l'estructura del text o la seva presentació.
- ▶ Quin és el llenguatge de marques més popular? **HTML**
(Hyper Text Markup Language)



```
Untitled - Notepad
File Edit Format View Help
<!DOCTYPE html>
<html>
<body>

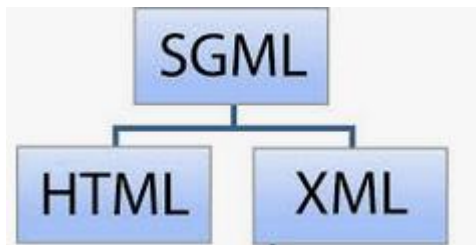
<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```

I on situem XML?

- ▶ XML (eXtensible Markup Language) és un metallenguatge.
- ▶ Un metallenguatge és un llenguatge que permet descriure altres llenguatges.
- ▶ XML és més aviat un format estàndard per a l'estructuració de dades que no pas un llenguatge de programació. Es defineix com a extensible perquè permet als usuaris crear les seves propies etiquetes (tags) i relacions estructurals entre elles.
- ▶ L'objectiu és definir, intercanviar i validar informació entre múltiples sistemes i aplicacions.



```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

HTML	XML
<p>Un document HTML formata i mostra la informació d'una pàgina web.</p> <p>Etiquetes (tags) ja predefinides pel llenguatge de marques.</p> <p>Pot no tindre etiqueta de tancament i es visualitza igual.</p> <p>No és sensible a majúscules i minúscules.</p> <p>És directament visible al navegador.</p>	<p>Un document XML porta dades exclusivament i la seva descripció.</p> <p>No té etiquetes predefinides. Pots crear i definir noves etiquetes si ho necessites.</p> <p>Ha de tancar les etiquetes sense excepció.</p> <p>És sensible a majúscules i minúscules.</p> <p>Només visible si es disposa de un programa que tracti el fitxer, o si fem servir css o si ho transformem a pdf per exemple.</p>

XML

```
<firstName>Maria</firstName>  
<lastName>Roberts</lastName>  
<dateBirth>12-11-1942</dateBirth>
```

HTML

```
<font size="3">Maria Roberts</font>  
<b>12-11-1942</b>
```

XML: estructura i sintaxi

- ▶ L'importància està en el contingut de les dades, no en com es veuran.
- ▶ A més, això ens permet diferenciar les dades de la seva presentació visual.
- ▶ Els fitxers XML estan formats per elements. Un element es compon de:
 - ▶ L'obertura d'una etiqueta (amb atributs o sense)
 - ▶ Un contingut (que pot ser un conjunt d'etiquetes)
 - ▶ Tancament d'etiqueta

```
<?xml version="1.0" encoding="utf-8" ?>
<books>
  <book>
    <bookid> B001 </bookid>
    <title> Understanding XML </title>
    <Price> $30 </Price>
    <Author>
      <FirstName> Lily </FirstName>
      <LastName> Hicks </LastName>
    </Author>
  </book>
  <book>
    <bookid> B002 </bookid>
    <title> .NET Framework </title>
    <Price> $45 </Price>
    <Author>
      <FirstName> Jasmine </FirstName>
      <LastName> Williams </LastName>
    </Author>
  </book>
</books>
```

XML: Parts d'un document

► Pròleg (opcional)

- Formada per instruccions de processament, que comencen per <? i acaben amb ?>
- Aquestes instruccions són marques, no elements (no són dades)
- La més habitual, és la que crida a un full d'estils:

```
<?xml-stylesheet href="persona.css" type="text/css" ?>  
<persona>  
    Carla Gómez  
</persona>
```

- També està la declaració XML, on especifiquem que es tracta d'un arxiu XML. Sempre ha de ser la primera línia de codi.

```
<?xml version="1.0" encoding="ASCII" standalone="yes" ?>  
<persona>  
    Carla Gómez  
</persona>
```

- L'atribut standalone indica al processador XML si un document XML és independent. Si no fa referencia a cap altre fitxer, ficarem standalone="yes". Per defecte, es basa en informació de fonts externes, i per aixó hauríem de ficar standalone="no". Però com és la opció per defecte, no cal ficar-ho.

XML: Parts d'un document

- ▶ Cos (les dades que un document XML ens ofereix)
- ▶ És un arbre únic d'elements marcats.

Exemple:

```
<?xml version="1.0"?>
<movies>
  <movie>
    <title>Casablanca</title>
    <director>Michael Curtiz</director>
    <actors>
      <actor>Humphrey Bogart</actor>
    </actors>
  </movie>
  <movie> ... </movie>
</movies>
```

En el nostre exemple, és tot el que va entre **<movie>** i **</movie>**, aquest element es coneix com **l'arrel de l'element**.

XML: Parts d'un document

- ▶ **Epíleg (opcional)**
- ▶ Format per 0 o més comentaris o instruccions de procés. Excepte declaracions XMLo de tipus de document.

- ▶ **Comentaris...**

Un document XML pot contenir anotacions en forma de comentari. Els comentaris no són part del contingut d'informació del document, i poden ser ignorats pels processadors XML. Els comentaris s'escriuen com:

```
<!-- ...text del comentari... -->
```

El text d'un comentari no pot contenir la seqüència —.

- ▶ **Entitats predefinides...**

Els símbols &, <, >, ' i " mai s'interpretarn com a marcat pel processador XML

XML: Elements

- ▶ Els elements XML poden tenir contingut (més elements i/o caràcters) o ser elements buits. Per exemple, un element amb contingut seria:

`<nom> Maria Roberts </nom>`

`<nom></nom>`

- ▶ Per diferenciar entre els diferents elements que componen un document XML fem servir etiquetes. Estan compostes per un nom i uns atributs, ha d'existir una etiqueta d'obertura i una altra de tancament.
- ▶ Com a regla, les etiquetes...
 - ▶ Han d'estar delimitades pels símbols `<` i `>`
 - ▶ El nom d'etiqueta ha de començar per una lletra o guió baix.
 - ▶ Majúscules i minúscules es diferencien. `<Persona>` és diferent de `<persona>` o `<PERSONA>`
 - ▶ Cada etiqueta d'obertura ha de tindre una de tancament.
 - ▶ El nom d'elements, atributs, etc. poden contenir guió baix, guió, punt, i caràcters alfanumèrics del tipus `a..z`, `A..Z`, `0..9`. No poden tenir espais en blanc!

XML: Atributs

- ▶ Igual que a HTML, a XML també tenim atributs. Un atribut s'afegeix a l'etiqueta d'inici d'element i té el format:

`<element nomAtribut="valor"> Contingut </element>`

- ▶ El valor dels atributs ha d'estar sempre entre cometes dobles o simples.
- ▶ Es poden definir tants atributs com calgui, i el seu ordre no importa.
- ▶ No es poden repetir noms d'atributs al mateix element.

Exemple 2:

```
<persona sexe="femení">  
  <nom>Carla</nom>  
  <cognom>Gómez</cognom>  
</persona>
```

Exemple 3:

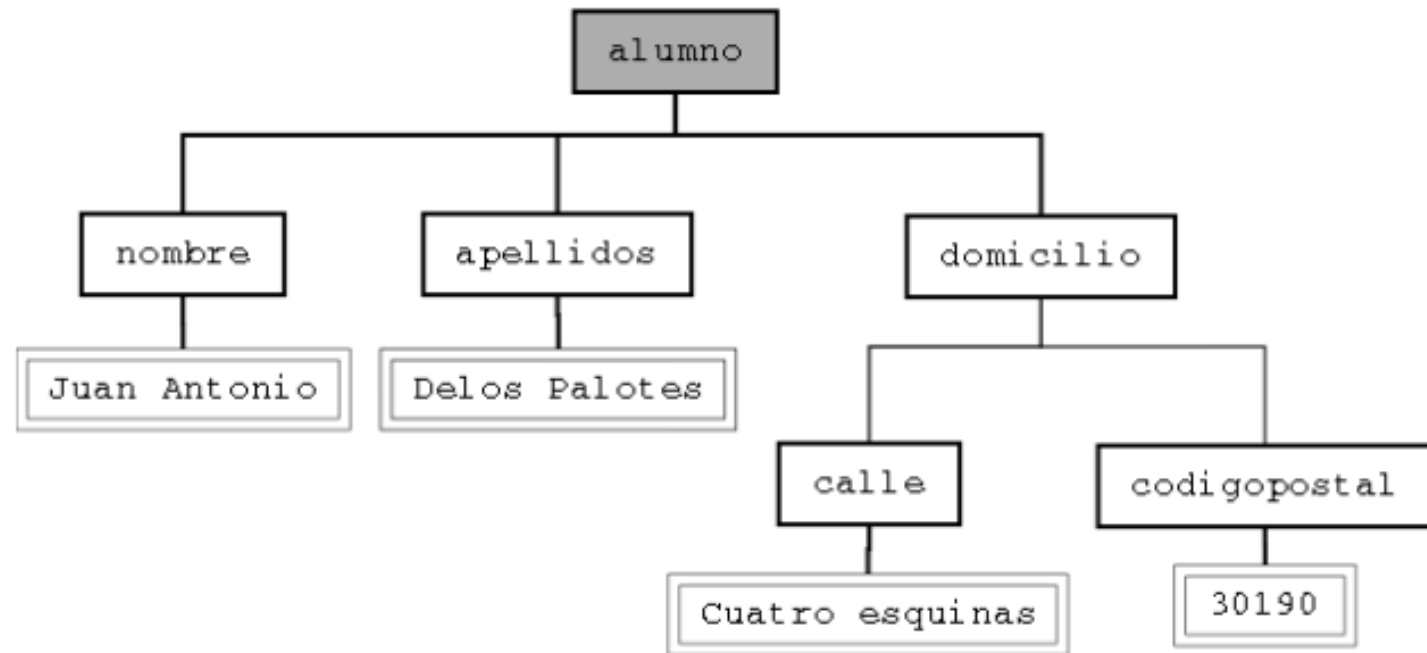
```
<persona>  
  <sexe>femení</sexe>  
  <nom>Carla</nom>  
  <cognom>Gómez</cognom>  
</persona>
```

- ▶ A l'exemple 2, el sexe és un atribut. Al 3 és un element.
- ▶ No és una regla oficial, però a XML s'aconsella evitar els atributs i fer servir elements preferentment.

XML: Imbricament d'elements

- ▶ Els documents XML formen una estructura de dades que s'anomena arbre. Un element no només pot contenir text, sino altres elements, que a la vegada poden tenir més elements, text, etc.
- ▶ Un document XML és un **arbre d'elements**. A l'element que està dintre d'un altre se'l coneix com a **fill**. I en el que està inclòs es diu **pare**.

```
<?xml version="1.0" ?>
<alumno>
  <nombre>Juan Antonio</nombre>
  <apellidos>Delos Palotes</apellidos>
  <domicilio>
    <calle>Cuatro esquinas</calle>
    <codigopostal>30190</codigopostal>
  </domicilio>
</alumno>
```



XML: Eines d'edició

- ▶ Tenim els editors clàssics (jEdit, Notepad ++ , etc...)
- ▶ O bé editors gràfics (XML Copy Editor)

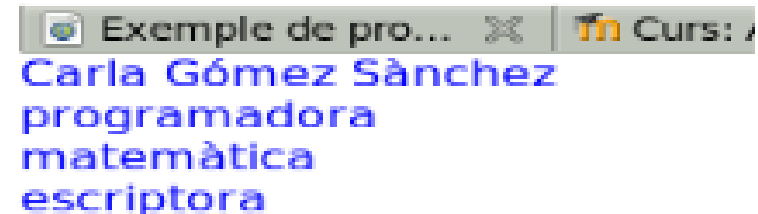
persona.xml

```
<?xml version="1.0" encoding="U TF-8"?>
<!DOCTYPE persona SYSTEM "persona.dtd">
<?xml-stylesheet href="persona.css" type="text/css"?>

<persona>
  <nom>Carla </nom>
  <cognoms>
    <primercognom>Gómez</primercognom>
    <segon cognom>Sánchez</segon cognom>
  </cognoms>
  <professio>programadora</professio>
  <professio>matemàtica</professio>
  <professio>escriptora</professio>
</persona>
```

persona.css

```
persona {font-family: Verdana, sans-serif; color: blue;font-size: 12pt;}
professio{display:block;}
```



Exemple de pro... X Curs: /

Carla Gómez Sánchez
programadora
matemàtica
escriptora

XML: Documents XML ben formats

- ▶ Regles per a què un document XML sigui sintacticament correcte:
 - ▶ Tota etiqueta oberta ha de tenir una de tancament associada
 - ▶ L'organització jeràrquica d'etiquetes ha de ser perfecta.
 - ▶ Un document només pot tenir un element arrel
 - ▶ Els atributs d'una etiqueta ha de tindre el valor entre cometes.
 - ▶ Existeixen regles per anomenar les etiquetes:
 - ▶ El nom ha de començar per una lletra o _ i no pot tenir espais en blanc
 - ▶ Les xifres i signes de puntuació només a partir de la segona lletra
 - ▶ No pot tenir el símbol : ni començar per la cadena "xml"

```
<?xml version="1.0"?>
<Empresa nom="XXX" data_fundacio="2001-11-15">
  <Delegacio nom="Central Barcelona" empleats="100">Text </Delegacio>
  <Delegacio nom="Central Tarragona" empleats="25">Text </Delegacio>
  <Delegacio nom="Lleida" empleats="13">
    <Empleat>
      <Nom>Joan Arbós</Nom>
      <Carrec>Director delegació</Carrec>
    </Empleat>
  </Delegacio>
</Empresa>
```

XML: Namespaces

- Els noms dels elements a XML venen definits pel programador. Això pot crear un conflicte si un programa fa servir varis arxius XML i hem ficat la mateixa etiqueta als dos.

```
<table>  
  <fila>  
    <col>Mobles Fusta</col>  
    <col>Mobles PVC</col>  
  </fila>  
</table>
```

```
<table>  
  <name>Taula de cafè</name>  
  <width>80</width>  
  <length>120</length>  
</table>
```

- Com podem indicar que es tracten de dos elements diferents i que no causi cap conflicte en els programes que puguin necessitar els dos arxius XML?
- Amb els namespaces (espais de noms). Simplement creem un identificador únic (no cal ni que apunti a cap espai real). Es tracta de donar un nom únic.
- Els namespaces s'apliquen tant a atributs com a elements.

XML: Namespaces

Per tal d'**assignar un sinònim** a l'espai de noms es fa mitjançant la etiqueta **xmlns**.

```
<exemple xmlns:ej="http://dominiej.com/rutaej">  
  <ej:element1>Text1 </ej:element1>  
  <ej:element2>Text2 </ej:element2>  
</exemple>
```

XML: Namespaces

```
<root>
  <h:table xmlns:h="http://www.w3.org/TR/html4/">
    <h:fila>
      <h:col>Mobles Fusta</h:col>
      <h:col>Mobles PVC</h:col>
    </h:fila>
  </h:table>

  <f:table xmlns:f="http://www.w3schools.com/mobles">
    <f:name>Taula de cafè</f:name>
    <f:width>80</f:width>
    <f:length>120</f:length>
  </f:table>
</root>
```

En aquest exemple, no hi haurà conflicte, perquè els dos elements **<table>** tenen noms diferents. Pertanyen a diferents espais de noms, per tal són diferents.

En l'exemple anterior, l'atribut **xmlns** de l'etiqueta **<table>** dona als prefixes **h:** i **f:** un espai de noms qualificat.

Quan un espai de noms es defineix per a un element, **xmlns:f="http://www.w3schools.com/mobles"** tots els elements descendents amb el mateix prefix estan associats amb el mateix espai de noms.

XML: Namespaces

También pueden **declararse namespaces en el elemento root**:

```
<root xmlns:h="http://www.w3.org/TR/html5/"
      xmlns:m="http://www.ejemplo.com/muebles">
  <h:table>
    <h:tr>
      <h:td>León</h:td>
      <h:td>Tigre</h:td>
    </h:tr>
  </h:table>
  <m:table >
    <m:name>Mesa de café</m:name>
    <m:colour>Marrón</m:colour>
  </m:table>
</root>
```

La URI no es para buscar información en esa URL, sino que se utiliza para darle un **nombre único al namespace**. Se suele emplear una URL porque las empresas a veces proporcionan información sobre dicho namespace en esa URL.