Introduction to Markup Languages

LMSGI – UNIT 1

Table of contents


XML
Markup
document
Language
Element
Processor
characters
Content
Character
Extensible
Application
Specification
example
edit
data
Unicode
Tag
empty-element
start-tag
section
attributes
may
design
several
elements
number
line-break
world
rules
begin
construct
step
alt
often
two
name
Tags
ends
use
languages

01

Concept

Concept

- Spoken language gets enriched by using intonation, gestures, pauses...
- Written language makes use of font size, bold, colors, numbered lists, bulleted lists..

FORMAT

Markup languages allow, among other things:

- Apply format to digital documents
- Structure the information
- Establish rules to validate the structure of the information

Concept

- Computer applications interpret documents encoded with markup languages to present them appropriately and pleasing

USER-AGENT

- Interprets marks ands applies the appropriate format to text:
  - Text processors: Microsoft Word, LibreOffice Writer...
  - Web browsers: Chrome, Edge, Firefox...
  - Printers
  - Voice synthesizers
  - ...

Concept

- Marks: signs located within a text that delimit and, sometimes, transform it (for example, by applying format).
  - Usually paired (initial and final)
  - Typical appearance: <xxx> (initial), </xxx> (final)

EXAMPLE:

- Open a **plain text editor** (such as Notepad or Kwrite) and type these lines:

```
<h1>I am a big text</h1>
<h3>I am a small text</h3>
```

- Save the file, not with txt extension but **html**
- Open the file with a web browser

Concept

- Sometimes, marks have other looks.

EXAMPLE:

- Open a text processor (such as Word or Write) and type just one word
- Apply bold to the word
- Save the file as RTF document (rtf extension).
- Now open that file in a plain text editor.

```
fe3082\kerning2\loch\af31506\hich\af31506\dbch\af31505\cgrid\la
ch\fcs1 \ab\af37\afs22 \ltrch\fcs0 \b\f37\fs22\lang10\langfe308
ch\af37\dbch\af31505\loch\f37 potato}{
fs22 \ltrch\fcs0 \f37\fs22\lang10\langfe3082\kerning0\langnp10\
```

Concept

MARKUP LANGUAGE
≠
PROGRAMMING LANGUAGE

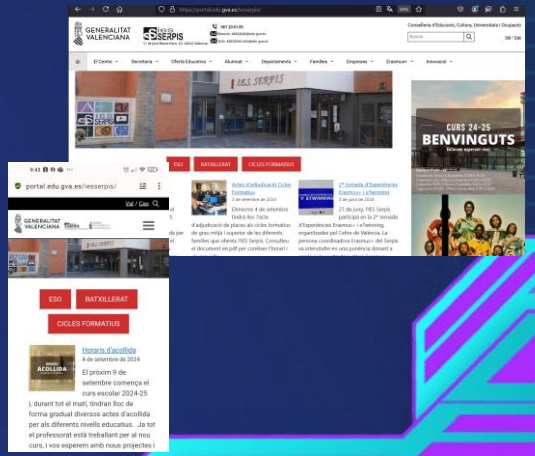
- Encoding with a markup language is not programming.
- Markup languages can be combined in the same document with other programming languages, such as JavaScript or PHP.

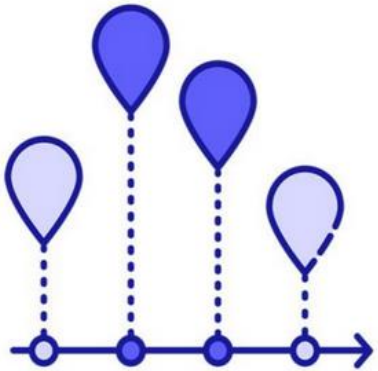
Concept

- Markup languages are independent from the final recipient (user-agent).
- Sometimes, user-agents with different interpretations of the same code can be found (for example, web browsers considering that bold letters should have different thickness).
- There are markup languages specialized in styling text, such as CSS. Linking a CSS stylesheet to a HTML document allows:
  - Separate between content and format.
  - Set differences in representation based on user-agent

EXAMPLE:

- Visit our school's website using a computer web browser
- Now try it with a smartphone

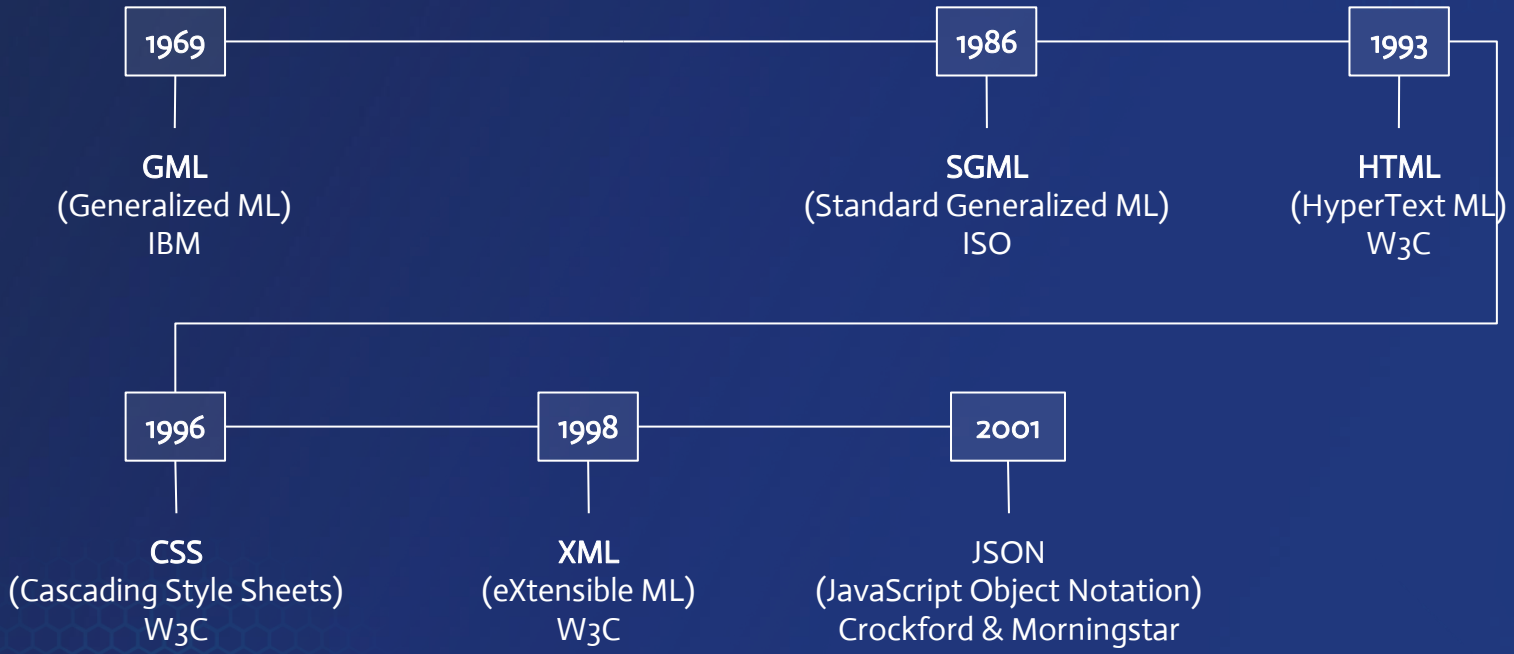




02

Timeline

Timeline


1969
GML
(Generalized ML)
IBM
1986
SGML
(Standard Generalized ML)
ISO
1993
HTML
(HyperText ML)
W3C
1996
CSS
(Cascading Style Sheets)
W3C
1998
XML
(eXtensible ML)
W3C
2001
JSON
(JavaScript Object Notation)
Crockford & Morningstar

03

Characteristics

Characteristics

Plain text

Formatless text-only code to promote compatibility and interoperability

Embedded

Markup and content, blended in the same document

Independent

...from the final device; Code interpretation depends on the user-agent

Flexible

Combinable with other markup and/or programming languages

Specialization

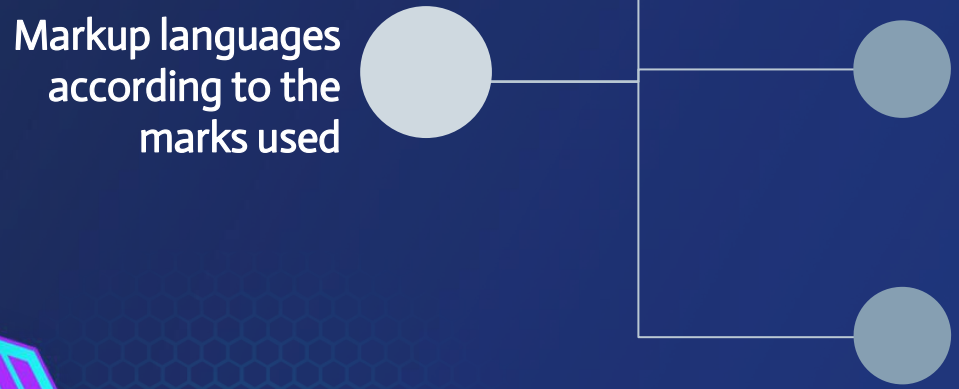
Original purpose: displaying text documents; but nowadays they have diverse and specific uses, such as:

- Vector graphic creation (SVG)
- Content sindication (RSS)
- Voice synthesis (SSML)

04

Classification

Classification



Presentation: they indicate the **format** of the text, but not its structure.

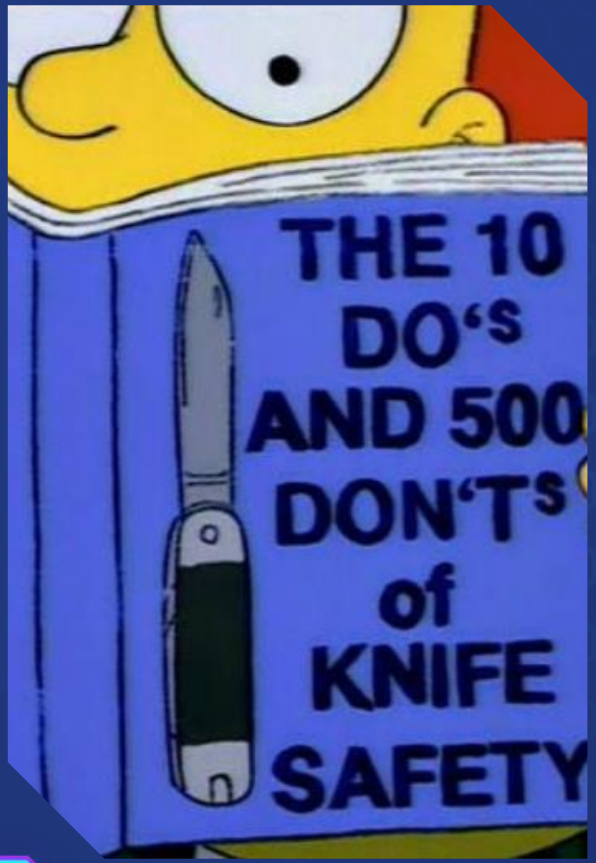
- Examples: RTF, TeX

Structural or **descriptive**: they indicate **the parts** into which the document is **structured**, but without detailing how it should be presented, or in what order

- Examples: XML, YAML

Hybrid: the indicate **both the format and the structure** of the document

- Examples: HTML, OOXML (MS Office)



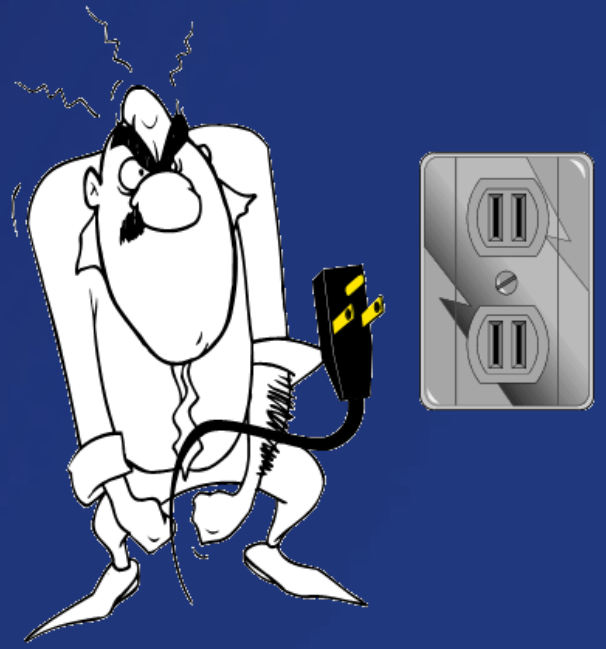
05

Organizations and Standards

Standardization

- Specification of standards to ensure the right proper functioning of independently-built elements
- Also called normalization

- In our field, the most relevant organizations that deal with standardization are ISO and W3C



Organizations

International Organization for Standardization

- Deals with standards in a wide variety of topics
- Experts worldwide reach an agreement to develop these standards
- For example:
  - ISO 9001: Quality measure systems
  - ISO 3166: Country codes
  - ISO 13216: ISOFIX child seats for cars
  - ISO 8879: SGML

Organizations

World Wide Web Consortium

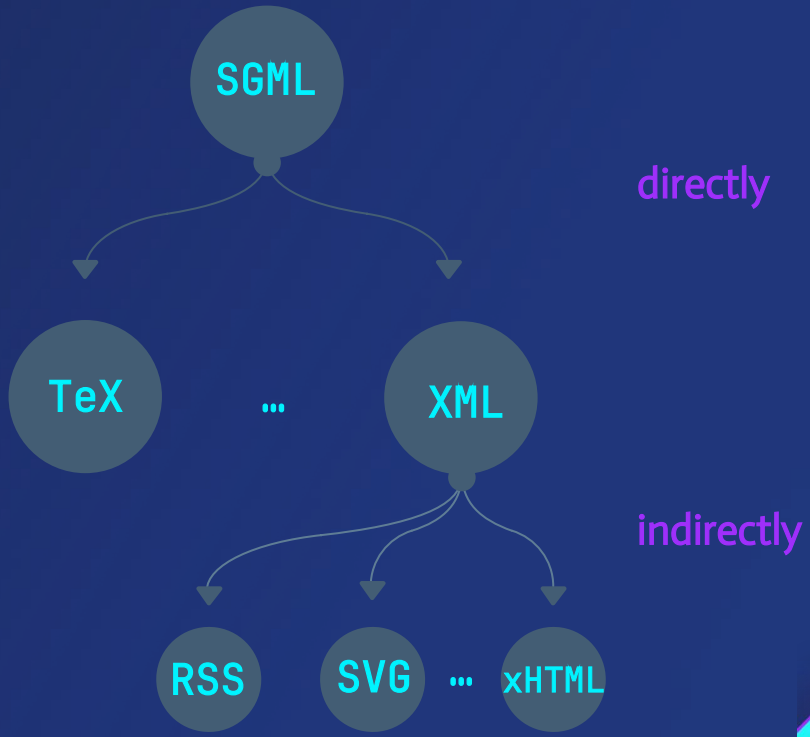
- Non-profit organization
- Formed by both private and public organizations and companies in the Internet sector, such as:
  - Software developing companies
  - Companies offering services via the Internet
  - Mobile device manufacturers
  - Universities
- They created the Web

06

Main Markup Languages

Main Markup Languages

Many markup languages are derived from SGML



HTML

- For creating web pages
- Capable of displaying text and multimedia (images, sound and video clips)
- Makes often use of hyperlinks (navigation to other pages, file downloading...)

- Created by Tim Berners-Lee and his team at W3C
  - Unable to foresee the expansion of the Web → major initial shortcomings, improved with new versions over the years and combined use with other technologies: CSS, JavaScript, browser plugins

```
<!DOCTYPE html>
<html>
  <head>
    <title>Web page example</title>
  </head>
  <body>
    <h1>Main title</h1>
    <h2>Second level title</h2>
    <p>Paragraph text with
<strong>bold</strong> and
<em>italics</em></p>
  </body>
</html>
```

XML

- Born as a simplification of SGML
- Metalanguage: language that allows creation of more specific languages

```
<?xml version="1.0" ?>
<person>
  <name>Guillermo</name>
  <surname1>Domingo</surname1>
  <address>
    <street>C/ Joncito, 23</street>
    <town>Requena</town>
    <zip>46340</zip>
    <country>España</country>
  </address>
  <phones>
    <phone type="fixed">999999999
    </phone>
    <phone type="mobile">777777777
    </phone>
  </phones>
</person>
```

XML

Characteristics:

- Extensible: allows you to create new labels
- Versatile: separates content, structure and presentation
- Structured: allows you to model data at any level of complexity
- Validatable: one can check the validity of the generated document
- Open: not linked to any company, OS, programming language...
- Simple

Its use extends both inside and outside the Internet, for structured information exchange among platforms:

- Light databases
- Spreadsheets
- Commercial transactions
- Storing information without using a relational DBMS

JSON

- Born to work together with JavaScript
- Currently very popular
- (Almost) the same characteristics as XML

```
{
"name": "Guillermo",
"surname1": "Domingo",
"address": {
  "street": "C/ Joncito, 23",
  "town": "Requena",
  "zip": "46340",
  "country": "Spain"
},
"phones": [
  {
    "type": "fixed",
    "number": "999999999"
  },
  {
    "type": "mobile",
    "number": "777777777"
  }
]}
```

Thanks!

Do you have any questions?

g.domingomartinez@edu.gva.es