



# XML

## XML - I Module 1

# Agenda

1

## **Introduction to XML**

# Objectives

At the end of this module, you will be able to:

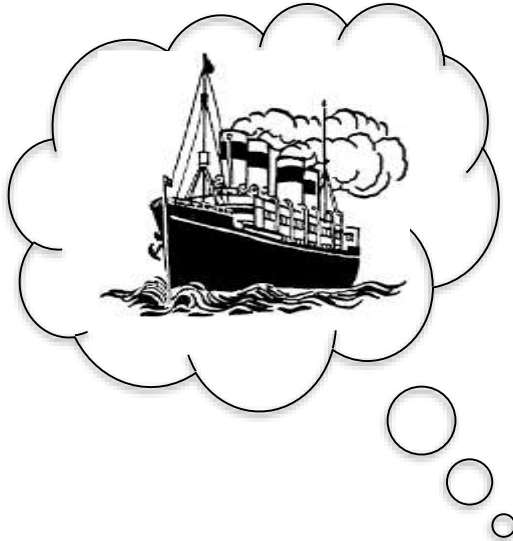
- Explain evolution of XML
- Describe XML Structure

# Introduction to XML



# How to understand data?

- What comes to your mind you hear the word 'TITANIC'?



**Will you think  
about the Ship  
or the  
Movie???**



# How to understand data? (Contd.).

- Here we are presented with a problem:

When you specify something, like ‘Titanic’, there should be a way to find out whether you meant the ship or the movie.

**How to describe data?**

This is what we are going to learn here...

# What is a Document ?

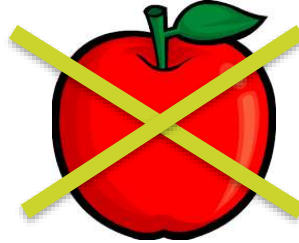
- Before we start learning XML, we will have a look at the need to have such a language. We begin with understanding what is a document.

*A document is a combination of data and markup.*

- What is data and what is markup?
- Imagine that you are reading a e-book. Find all the occurrences of the word 'Apple' in the book.
- Using 'find' option, we can easily find the word wherever it is present in the document.

# What is a Document ?(Contd.).

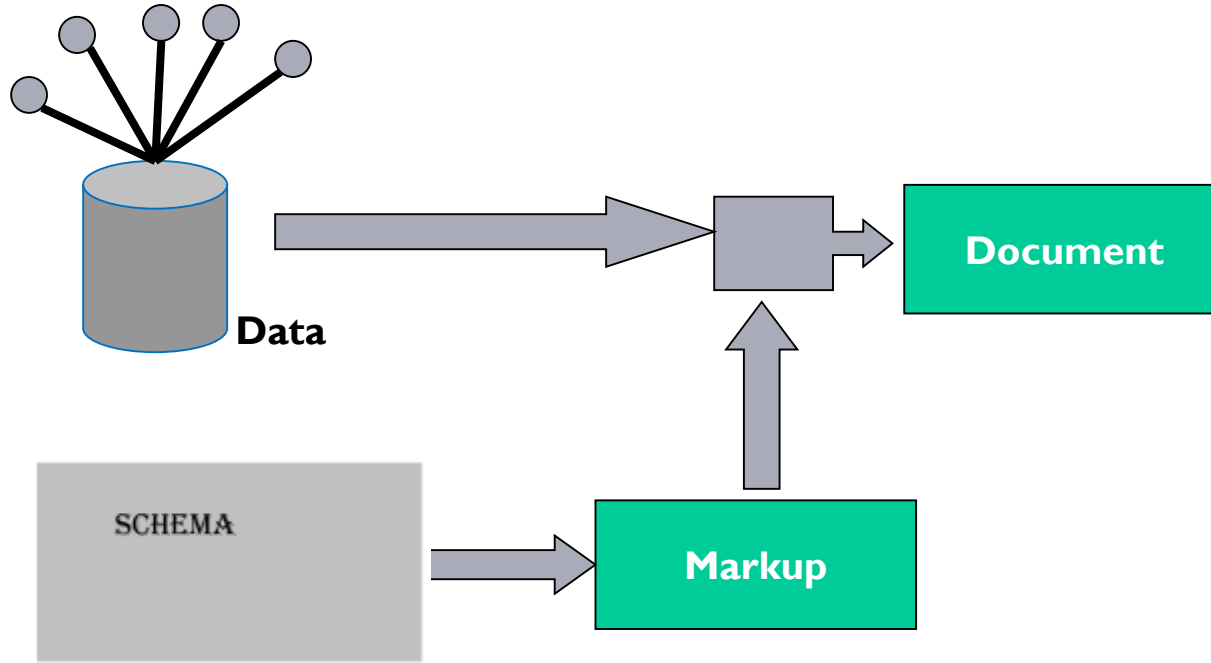
- But now what if we need to find all the occurrences of the word 'Apple' where it is meant as the 'Apple' company and not the fruit.



**In a plain text document,  
finding a word with a specific  
meaning is difficult!!!  
Especially if that word has  
more than one meaning!  
The same problem occurs  
during data processing also...**



# What is a Document?(Contd.).



A document is a combination of data and markup. It is very similar to a person reading a book. While reading one markup's information he finds it relevant. Similarly, its an effort to mark relevant data. The data could be pulled out of any relational schema or any application.

# Role of Markup

- Markup describes the data in the document and how it should be interpreted.
- Markup is anything inside a pair of angle brackets (< >)
- **Need for markups**

The basic purpose of going for a markup is to guarantee “What You See Is What You Get” Format. Meaning, there should not be any discrepancies in display even across dissimilar systems at any given time.

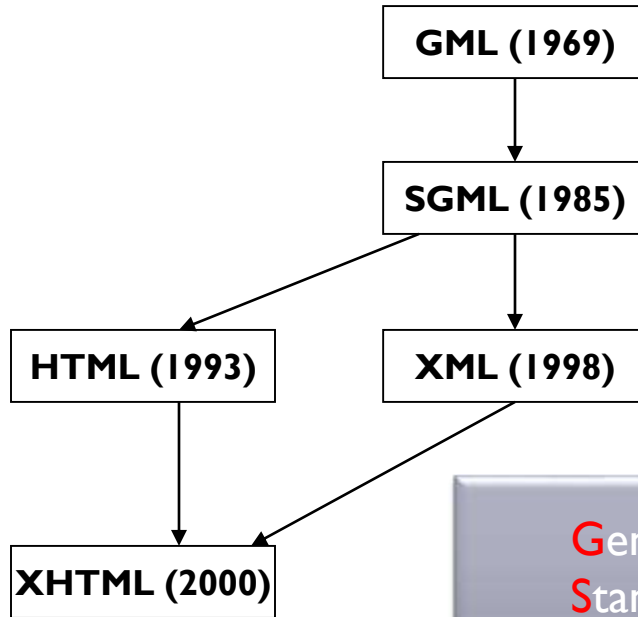
- Data Transfer across various dissimilar applications
- A markup language must specify

What markup is allowed

What markup is required

How markup is to be distinguished from text

# Markup Languages



The concept of markup languages was initially implemented by IBM in 1969 with the development of the Generalized Markup Language by Goldfrab, Mosher, Lorie.

Generalized Markup Language  
Standard Generalized Markup Language  
HyperText Markup Language  
eXtensible Markup Language  
eXtensible HyperText Markup Language

# HTML – Hyper Text Markup Language

Primarily designed for desired look and feel

Intended to remove difficulties of GML and SGML

A non structured markup language

Did not support what is called as 'Content Markup'.

Both presentation markups and structural markups were placed in the same Document.

# What is XML?



Extensible Markup Language

A standard for data exchange

A specification recommended by W3C (World Wide Web Consortium)

Structured markup language

Simplified SGML

Designed to describe data

# HTML and XML – A comparison

## HTML

Used to mark up text so that it can be displayed to the user

Uses fixed set of predefined tags

Describes appearance as well as structure of the data

Is for humans. So, most browsers correct or ignore as many errors as they can

Designed to display data and to focus on how data looks

HTML describes both structure (e.g. <p>, <h2>) and appearance (e.g. <br>, <font>, <i>)

## XML

Used to be processed by applications/computers so that it can be marked up

You can create your own tags in XML

Describes the content

Is for computers. So, the rules are strict and errors are not allowed

Designed to describe data and to focus on what data is

# Uses of XML

Can separate data from HTML

With XML, your data is stored outside your HTML

Used to exchange data

With XML, data can be exchanged between incompatible systems

Can be used to share data

With XML, plain text files can be used to share data

Can be used to store data

With XML, plain text files can be used to store data.

# Example of an XML Document

<?xml version="1.0"?>



Prologue

<Person>

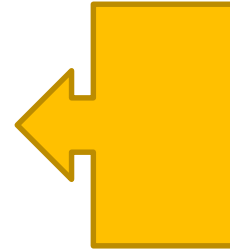


Root Element

<Name>Jessy</Name>

<Age>12</Age>

<Country>India</Country>



Document

</Person>



# XML Building Blocks - Prolog

- A part of an XML document that precedes XML data
- It includes
  - A declaration: version [, encoding, standalone]
  - An optional DTD (Document Type Definition )\*
- Example of an XML declaration

```
<?xml version="1.0" encoding="ISO-8859-1"  
standalone="yes"?>
```

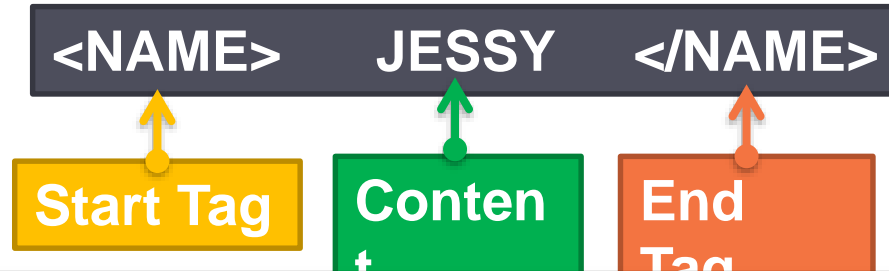
\* Note: DTD will be discussed later

# XML Building Blocks

The XML document is also built from:

**Elements:** Basic unit of an XML document

- An element is a logical structure in XML that is delimited by a start tag and an end tag
- Element consists of 3 parts: Start Tag, Content and End Tag



Rules:

- Start Tag and End Tag should match and it is case sensitive
- Content cannot contain `<` or `&`

# XML Building Blocks

**Attributes:** provides additional information

- Specified in the start tag of the element
- Have a key- value pair
- An element can have any number of attributes
- Attribute can have only one value

```
<FLOWER COLOR="RED" > ROSE </FLOWER>
```

Attribute  
Name

Attribute  
Value

Rules:

- Attribute value should be enclosed in double quotes
- Attribute names within an element are unique
- attribute names must begin with a letter or underscore and can contain letters, digits, underscore(\_), dot(.), hyphen(-)

# Well-formed XML document

An XML document is said to be well-formed if it follows basic syntax rules specified for XML by W3C

**XML tags are case sensitive**

**Must have only one root element**

**Every element must have a closing tag**

**Elements must be properly nested**

**Attribute values must always be quoted (single or double)**

# Well-formed XML document(Contd.).

- Others:

**Comments in XML: `<!-- This is a comment -->`**

**Every element must have a start tag and an end tag. However empty elements can end with `/>` instead of `>` (e.g. : `<book id =“001” />`)**

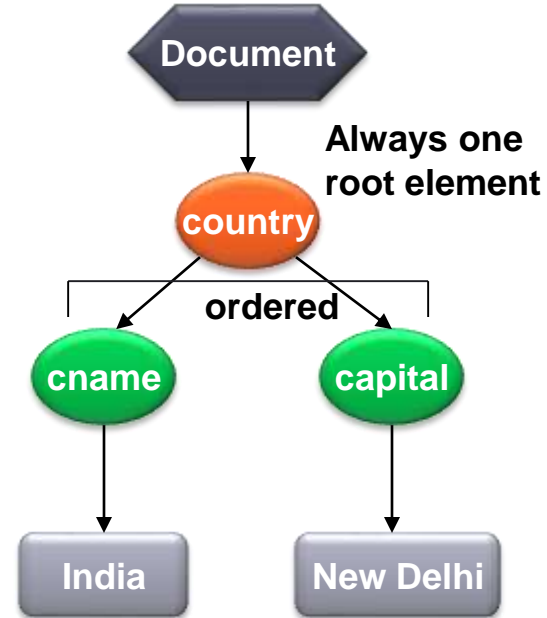
**Colon(:) is used for namespaces.**

# XML Tree Structure

## XML Document

```
<country>  
  <cname>India</cname>  
  <capital>New Delhi</capital>  
</country>
```

## Document tree



# Assignment



# Summary

**In this module, you were able to**

- Explain evolution of XML
- Describe XML Structure
- Understand well-formedness of documents





# Thank You