



# Langages du Web

MI – Semestre 7  
Département informatique  
UFR des Sciences et technologies  
Université de Thiès



# Plan du cours

1. Présentation du cours
2. Introduction
3. XML et ses applications
4. RDF, OWL & RIF



# Plan du cours

1. Présentation du cours
2. Introduction
3. XML et ses applications
4. RDF, OWL & RIF



# Présentation du cours (1)

## **Unité d'Enseignement**

Titre : INFORMATIQUE

Sigle : INF 113

## **Élément constitutif**

Titre : Langages du Web

Sigle : INF 1132

## **Autres éléments constitutifs de l'UE**

Titre : Bases de données réparties

Sigle : INF 1131

Volume horaire



# Présentation du cours (2)

## Horaires

- CM : 20H
- TD/TP : 20H
- TPE : 40H
- Coefficient de l'UE : 3
- Crédits de l'UE : 9

## Evaluation

- Contrôle des connaissances : 40%
- Examen écrit : 60%



## Présentation du cours (3)

### Responsable du cours

Pr Mouhamadou THIAM

Maître de conférences en Informatique

Intelligence Artificielle : Sémantique Web

Email : [mthiam@univ-thies.sn](mailto:mthiam@univ-thies.sn)

### Travaux dirigés et pratiques

M. ???????



# Plan du cours

1. Présentation du cours
- 2. Introduction**
3. XML et ses applications
4. RDF, OWL & RIF







# Langage ...

- Langage humain
- Langage animal
- Langage formel
- Langage naturel
- Langage de programmation
- Langage informatique
- Langage de script
- Langage SMS
- Langage des fleurs
- Langage interprété informatique



# Web...





## Web [wikipedia]...

- **World Wide Web (WWW)**, littéralement la « **toile (d'araignée) mondiale** », communément appelé le **Web**, et parfois la **Toile**, est un système **hypertexte** public fonctionnant sur **Internet**.
- Le Web permet de consulter, avec un navigateur, des pages accessibles sur des sites. L'image de la toile d'araignée vient des hyperliens qui lient les pages web entre elles



## Web...

- Applications d'Internet
- Distinct d'autres applications comme
  - courrier électronique,
  - messagerie instantanée,
  - partage de fichiers en pair à pair.
- Inventé par ***Tim Berners-Lee*** et ***Robert Cailliau*** au début des années 90



# La révolution du Web...

- Depuis vingt cinq ans, Internet révolutionne l'informatique « grand public »
- **HTML** est le langage du Web...
  - Même si on trouve aussi du .doc, .ps, .pdf... et des images (jpg, gif), du son, de la vidéo...



# La révolution du Web...

- Des milliards de pages existent actuellement
  - public/privé,
  - statique/dynamique,
  - visible/caché
- Support naturel pour l'information distribuée
  - À destination d'êtres humains et
  - de plus en plus, pour des applications



# La révolution du Web...

- Exemples:
  - B2C (commerce électronique)
  - B2B (achats groupés)
  - Bibliothèques et fonds documentaires en ligne
- HTML n'est pas adapté pour ces applications
  - Ces applications (programmes) ont besoin de typage pour représenter **la structure** des données



## Prérequis ...

- Les bases de données (relationnelle, ...)
  - Conception de schémas : relationnel, UML, ...
  - Les langages de requêtes : SQL, OQL, ...
- Le langage HTML : tables, listes, formulaires, ...
- Le langage Java
  - Héritage
  - Utilisation d'API





## Prérequis ...

- Les bases de données (relationnelle, ...)
  - Conception de schémas : relationnel, UML, ...
  - Les langages de requêtes : SQL, OQL, ...
- Le langage HTML : tables, listes, formulaires, ...
- Le langage Java
  - Héritage
  - Utilisation d'API



## Problématique ...

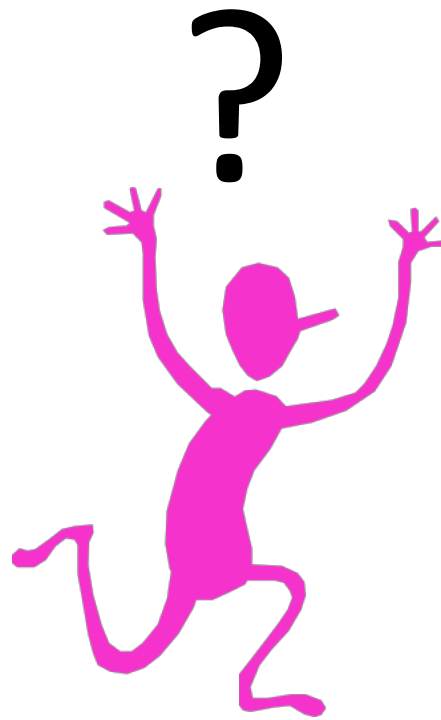
- Besoins d'accès contenu du web de manière précise
- Recherche mot-clef utilisée : pas satisfaisant;
- Méthodes ne garantissent pas
  - Pertinence de la réponse,
  - Réponse : document lui-même
  - Nombre exorbitant de réponses
  - Majeure partie répondent pas à la question posée
- Le web sémantique **human → machine readable**



## Résumé historique du web

- **1989** : **Tim Berners-Lee** (CERN, Genève) commence le développement d'un **système hypertexte**.
- **1990** : Premières définitions pour **HTTP, HTML, URL**.
- **1992** : Premier **annuaire de sites Web**. **26 sites**.
- **1994** : **Netscape Navigator 1.0** ; Fondation du **W3C**
- **1998** : Plus de **2 millions de sites** ; Création de **Google**.
- **2000** : **XHTML 1.0**.
- **2004** : **Firefox 1.0**.
- **2005** : Plus de 60 millions de sites !!!!

END





# Plan du cours

1. Présentation du cours
2. Introduction
- 3. XML et ses applications**
4. RDF, OWL & RIF

# Lectures

- Obligatoires
  - Recommendations du W3C - [http://www.w3.org/TR/tr-technology-stds#w3c\\_all](http://www.w3.org/TR/tr-technology-stds#w3c_all)
    - XML, XML Schema
    - XPath, XQuery
    - XSLT
  - <http://www.w3.org/History/1989/proposal.html>
  - *The Semantic Web : A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities* by TIM BERNERS-LEE, JAMES HENDLER and ORA LASSILA
- Facultatifs
  - Recommendations HTML du W3C



# XML et ses applications : plan

1. Introduction à XML
2. Schémas de documents XML
  - a. DTD
  - b. XML Schema
3. Le modèle DOM
4. Interrogation : XPath
5. Transformation et présentation : XSLT
6. Programmation : XML et Java (API DOM, SAX, ...)

# **Introduction à l'eXtensible Markup Language –XML**



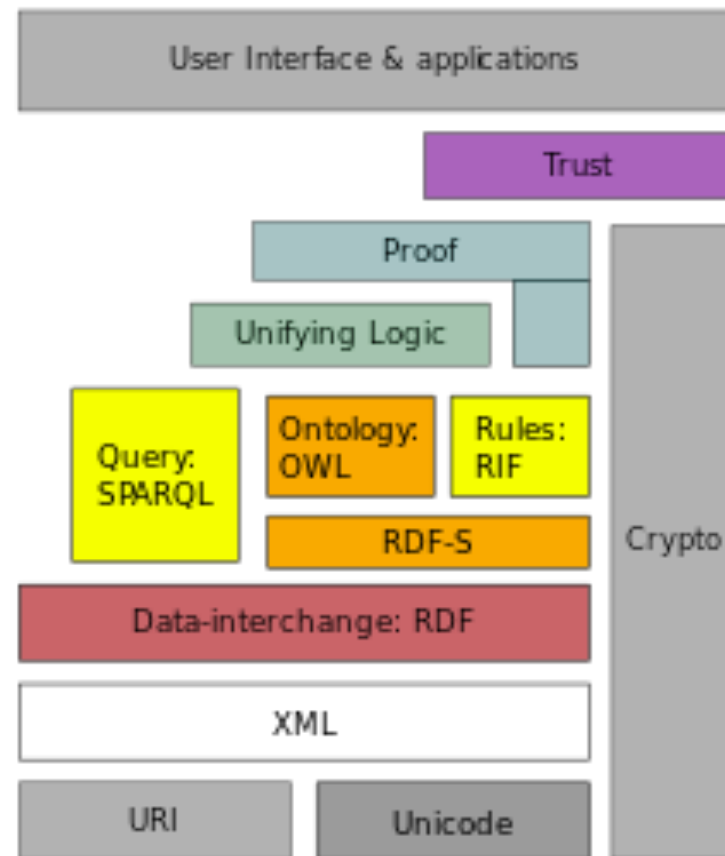


## De SGML à XML en passant par HTML

- SGML[1969, ISO 8879 1986]
  - Inventé par *Charles Goldfarb*, juriste chez IBM.
  - Du balisage typographique au balisage logique.
- HTML[1989-1992, HTML5 depuis 2007]
  - Inventé par *Tim Berners-Lee* comme langage de description des pages du Web.
  - Concept d'ancre : hypertexte.
- XML[1996, Rec. W3C 1998, Dan Cannolly]
  - Une synthèse de SGML et de HTML.



# Modèle en couches de Tim Berners-Lee





# XML : Document semi-structuré

- Un Document Semi-Structuré peut se définir par la définition de ses principaux composants (Poulet et al, 1997): **contenu** et **structure**
  - **Contenu** : le contenu d'un document semi-structuré désigne le contenu textuel ou multimédia par exemple des paragraphes, des figures ou des images
  - **Structure**
    - **Logique** : elle définit l'organisation hiérarchique des données du document
    - **Physique** : elle définit la présentation du document sur le support



# XML : le modèle de données du Web

- HTML versus XML
  - Attention : HTML et XML ne sont pas concurrents !!!
    - **Présentation des données** pour HTML
    - **Représentation des données** pour XML : les données et leur visuel est indépendant
  - Pour s'en convaincre, la présentation d'un document XML nécessite sa transformation en HTML via XSLT
    - XSLT= eXtensible Stylesheet Language Transform



# XML : le modèle de données du Web

- Exemple
  - Le fichier [HTML](#) contenant présentation et données
  - Le fichier [XML](#) ne contenant que les données
  - Le fichier [XSL](#) transformant le fichier XML en HTML



# Exemple de document (text / HTML)

Construire une application XML

Domaines d'application d'XML  
Vue d'ensemble

XML : battage ou réalité ?

Un document

En HTML

```
<!--extrait d'un document HTML-->
<HTML>
<HEAD>
  <TITLE> Construire une application XML </TITLE>
</HEAD>
<BODY>
  <center>
    <P>Domaines d'application d'XML <br>
      Vue d'ensemble</P>

    <P>XML : battage ou réalité ? </P></center>
  </BODY>
</HTML>
```



# Exemple de document (HTML / XML)

```
<!--extrait d'un document HTML-->
<HTML>
<HEAD>
  <TITLE> Construire une application XML </TITLE>
</HEAD>
<BODY>
  <center>
    <P>Domaines d'application d'XML <br>
    Vue d'ensemble</P>
    <P>XML : battage ou réalité ? </P></center>
  </BODY>
</HTML>
```

En HTML

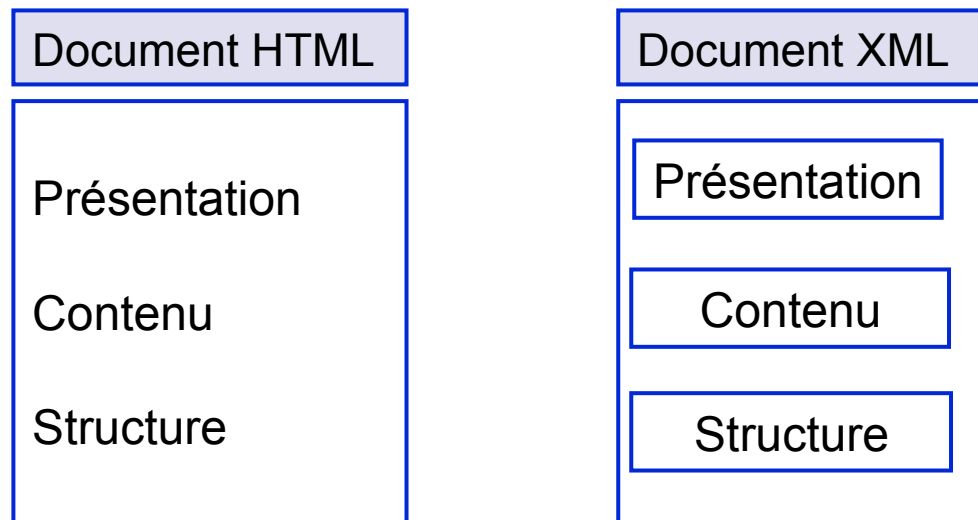
En XML

Presentation  
Contenu  
Structure

```
<!--extrait d'un document XML-->
<?xml version="1.0" ?>
<livre>
  <couverture>
    <titre> Construire une application XML </titre>
    <auteur> Jean-Christophe Bernadac </auteur>
    <auteur> François Knab </auteur>
  </couverture>
  <chapitre n="1">
    <p> XML : battage ou réalité ? </p>
  </chapitre>
</livre>
```



# HTML versus XML







# Exemple de document

## *Itinéraires skieurs dans la Vallée de la Clarée*

par Jean-Gabriel Ravary

Le Polygraphe, *éditeur*  
1991

### **Vallon des Muandes**

Vallon situé à l'est du refuge des Drayères.

Le vallon le plus utilisé pour la traversée sur la Vallée Etroite. Ce vallon est également accessible du refuge Laval.

#### **Col de Névache (2 794 m) \*\* n° 1**

S'élever au-dessus du refuge des Drayères en direction est. Suivre la rive droite du torrent de Brune puis s'engager sur le flanc droit du ravin des Muandes que l'on quitte vers 2500 m pour rejoindre le col situé au nord.

Descente possible sur Valmeinier. Départ assez raide.

#### **Pointe de Névache (2 892 m) \*\*\* n° 2**

Du col de Névache (*itinéraire n° 1*), suivre la ligne de crête qui mène à la pointe de Névache. Attention : corniches possibles. Crampons utiles au printemps.



# Le guide “itinéraires skieurs” en XML

```
<?xml version="1.0"?>
<guide>
  <titre>Itinéraires skieurs dans la vallée de la Clarée</titre>
  <auteur>Jean-Gabriel Ravary</auteur>
  <editeur>Le Polygraphe</editeur>
  <année>1991</année>
  ...
  <vallon id="V15">
    <nom>Vallon des Muandes</nom>
    <intro>
      <para>Vallon situé à l'est du refuge des Drayères.</para>
      <para>Le vallon le plus utilisé pour la traversée sur la Vallée
        Étroite. Ce vallon est également accessible du refuge Laval.</para>
    </intro>
    ... Itinéraires ...
  </vallon>
</guide>
```



# Le guide “itinéraires skieurs” en XML

```
<itinéraire id="I15.1">
  <nom>Col de Névache</nom><alt>2794</alt><cotation>**</cotation>
  <num>1</num>
  <para>
    S'élever au-dessus du refuge des Drayères en direction est. Suivre la rive droite du
    torrent de Brune puis s'engager sur le flanc droit du ravin des Muandes que l'on quitte
    vers 2500 m pour rejoindre le col situé au nord. Descente possible sur Valmeinier.
    <note type="prudence">Départ assez raide.</note>
  </para>
</itinéraire>
```



# Le guide “itinéraires skieurs” en XML

```
<itinéraire id="I15.2">
  <nom>Pointe de Névache</nom><alt>2892</alt><cotation>***</cotation>
  <num>2</num>
  <para>Du col de Névache <renvoi cible="I15.1"/>, suivre la ligne de crête qui mène à la
  pointe de Névache.
    <note type="prudence">Attention : corniches possibles.</note>
    <note type="materiel">Crampons utiles au printemps.</note>
  </para>
</itinéraire>
```



# En résumé

- Un **document** XML est découpé en **éléments** structurés hiérarchiquement.
- Un document a un élément racine appelé **élément du document**.
- Un élément est composé :
  - d'un **nom** qui spécifie son type,
  - **d'attributs**,
  - d'un **contenu** formé d'**éléments** ou de **textes**.



# En résumé

- Un texte est une chaîne de caractères.
- Un attribut a un nom et une valeur qui est une chaîne de caractères.
- Syntaxiquement, les éléments d'un document XML sont marqués dans le document lui-même par des paires de balises ouvrantes et fermantes.



# Structure d'un document XML

- Un document XML est composé :
  - d'un **prologue facultatif** (voir ci-après « Organisation d'un document XML »),
  - de **l'élément du document** qui est lui-même composé **d'éléments et de textes**



# Structure d'un document XML

- Dans le prologue et dans le contenu d'un élément, on peut insérer :
  - des **commentaires**,
  - des **instructions de traitement** destinés aux applications traitant le document.
- Un document peut être découpé en **entités** enregistrées dans un ou plusieurs fichiers.





# Nom et tokens de nom

- Un **caractère de nom** est soit une lettre, soit un chiffre, soit un point, soit un tiret, soit un espace souligné, soit un deux-points.
  - $\{[A..Z], [0..9], ., -, \_, : \}$
- Un **nom** est une suite de un ou plusieurs caractères dont : le premier est soit une lettre, soit un espace souligné, soit un deux-points  $\{ \_, : \}$ , chacun des suivants est un caractère de nom.

Par exemple : **xml:lan**, **extrait\_de titre**, **poeme-79**

- Un **token de nom** est une suite de un ou plus caractères de nom.



# Un élément XML

- Un élément est composé :
  - d'une **balise de début** qui contient le nom de l'élément et éventuellement ses attributs `<elt attr="valAttr">`,
  - d'un **contenu**,
  - d'une **balise de fin** `</elt>`.



# Un élément XML

- Par exemple :  

```
<note type="prudence">  
    Départ assez raide.  
</note>
```
- balise de début : `<note type="prudence">`
- nom : `note`
- attribut : `type="prudence"`
- contenu : `Départ assez raide.`
- balise de fin : `</note>`



# Contenu d'un élément XML

- **Vide** : `<elt></elt>` ou `<elt/>`

Exemple :

`<renvoi cible="I15.1"></renvoi>`

ou

`<renvoi cible="I15.1"/>`

- **Composé d'éléments** :

`<intro>`

`<para>Vallon situé à l'est du refuge ...</para>`

`<para>Le vallon le plus utilisé pour la traversée ...</para>`

`</intro>`



# Contenu d'un élément XML

- **Mixte** : mélange de textes et d'éléments

<nom>Col de Névache</nom>

<para>Du col de Névache <renvoi cible="I15.1"/> ,

suivre la ligne de crête qui mène à la pointe de Névache.

<note type ="prudence">Attention : corniches possibles. </note>

<note type="matériel">Crampons utiles au printemps.</note>

</para>



# Élément mixte

- Le contenu d'un élément mixte est constitué d'une chaîne de caractères dans laquelle peuvent être insérés des éléments. Cette insertion découpe ce contenu en deux types de constituants :
  - les plus longues suites d'au moins un caractère dans lesquelles ne sont pas insérés des éléments : nous les appellerons **textes**,
  - les éléments.



# Élément mixte

- Par exemple, le contenu :  
Du col de Névache <renvoi cible="I15.1"/>, suivre la ligne de crête qui mène à la pointe de Névache.
- comprend dans l'ordre :  
le texte : Du col de Névache  
l'élément : <renvoi cible="I15.1"/>  
le texte : , suivre la ligne de crête qui mène à la pointe de Névache.



## Section CDATA

- Lorsqu'un texte contient des caractères qui jouent un rôle de **délimiteur** dans la syntaxe XML, il est nécessaire de pouvoir **inhiber** ce rôle. Ceci peut être fait en insérant le texte contenant ces délimiteurs dans une section **CDATA** sous la forme suivante :

**<![CDATA[*texte contenant des délimiteurs*]]>**





## Section CDATA

- Le texte inséré peut contenir n'importe quels caractères excepté la chaîne `]]`. Une section CDATA ne peut donc pas en contenir une autre.
- Par exemple, la phrase :  
« L'expression `<ALT>2794</ALT>` est un élément XML. »
- peut être représentée par l'élément suivant :  
`<phrase>L'expression  
    <![CDATA[<ALT>2794</ALT>]]> est un élément XML.  
</phrase>`



# Attributs

- Un **attribut** est un couple **nom-valeur** où :
  - le nom est un nom XML,
  - la valeur est une suite de caractères.
- Par exemple :  
`type="prudence"`
- Si une valeur d'attribut est placée entre guillemets, elle peut contenir des apostrophes et si elle est placée entre apostrophes, elle peut contenir des guillemets.
- Par exemple :  
`select="itinéraire[cotation='*****']"`  
`select='itinéraire[cotation="*****"]'`



# Liens internes

- Tout élément peut avoir un attribut ayant pour valeur un token de nom qui l'identifie dans le document :
  - les tokens de nom **I15.1** et **I15.2** identifient les itinéraires n° 1 et n° 2 du Vallon des Muandes dans le document « Itinéraires skieurs ».
- L'identificateur d'un élément permet d'y faire référence depuis d'autres éléments :
  - `<itinéraire id="I15.1"> ....</itinéraire>`
  - l'élément `<renvoi cible="I15.1">` renvoie à l'itinéraire n° 1 de ce même vallon).



# Commentaires

- Un commentaire est une phrase ayant la forme suivante :  
*<!--texte du commentaire-->*
- Un commentaire peut contenir n'importe quel caractère **excepté --**.  
Un commentaire ne peut donc pas inclure un autre commentaire.
- Un commentaire peut être inclus dans le contenu d'un élément mais pas à l'intérieur d'une balise.
- Exemple :  
*<!-- Les itinéraires sont classes par vallon -->*



# Résumé

- **langage de balises** (simple à implanter, mais extensible) : chacun définit ses propres balises
- **séparation** de la **présentation** et du **contenu**, conçu pour
  - **décrire données** en se concentrant sur leur **structure**
  - **assurer l'interopérabilité**
- Utilisation d'un **DTD** ou un **XML Schéma** : modèle de données
- XML et HTML complémentaires
  - **XML** ne remplace pas **HTML**
  - **XSL / XSLT** transforme **XML** en **HTML**

END

