Université Abderrahmane Mira de Béjaia
Faculté des Sciences Exactes
Département d'Informatique

Langages du Web sémantique

(Cours 11)

Dr EL BOUHISSI Houda

Octobre 2022

Objectifs

- Découvrir les langages pour exprimer une ontologie.

- Découvrir le langage d'interrogation d'une ontologie.

Extensible Markup Language : XML

Il s'agit plutôt d'un métalangage qui permet de définir nos propres balises pour nos documents.

Langage de description et d'échange de :

- Documents structurés
- Données structurées

- Une syntaxe puissante et souple pour les documents structurés.
- N'impose aucune contrainte sémantique sur la signification de ces documents
- Un document XML est un arbre ordonné étiqueté.

http://www.w3.org/XML

XML : Eléments syntaxiques

XML est sensible à la casse

Chaque élément doit commencer par une balise ouvrante et se termine par une balise fermante. Eventuellement, un élément vide peut être représenté par une balise simple.

L'imbrication des éléments du document se fait sans chevauchement. Concrètement, cela signifie que si la balise ouvrante d'un élément se trouve à l'intérieur d'un élément parent, la balise fermante se trouvera dans le même élément.

La valeur d'un attribut doit être encadrée de guillemets, simples ou doubles.

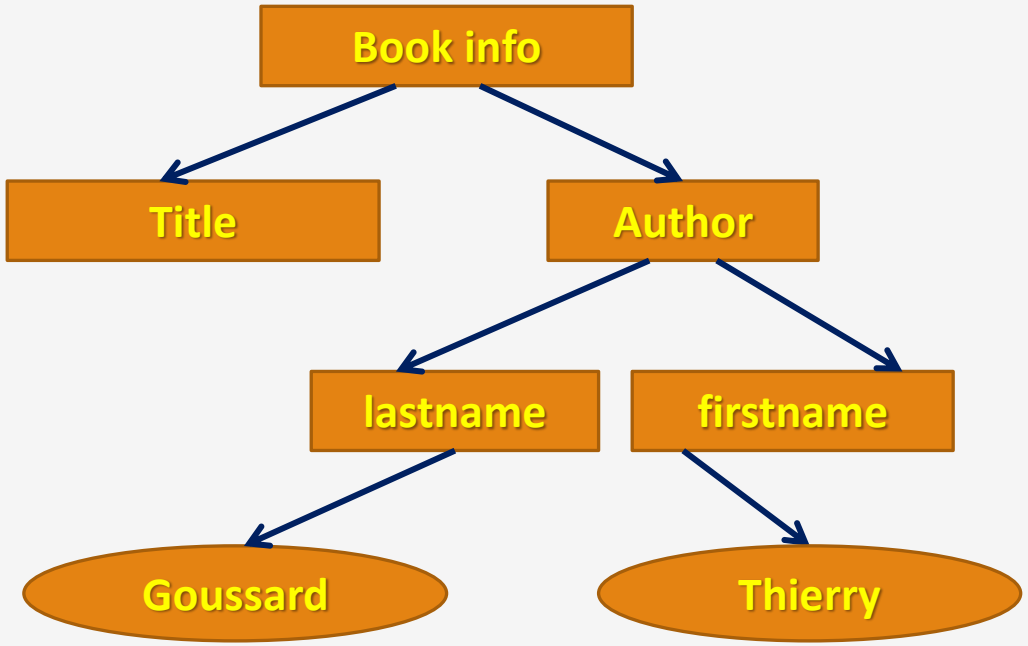
Les espaces de noms introduits en XML afin de pouvoir mélanger plusieurs vocabulaires au sein d'un même document.

XML : Technologie de base

```
<book info>
            <title> Java</title>
            <author>

 <lastname>Goussard</lastname>

 <firstname>Thierry</firstname>
            </author>
</book info>
```



XML : Outils de programmation

XML propose différentes interfaces de programmation (API) qui facilitent la manipulation de documents par les processeurs XML.

Les deux API principales sont :

- Document Object Model (DOM), qui procède par construction et manipulation d'un arbre logique complet.
- Simple API for XML (SAX), qui permet de débuter le traitement d'un document XML avant d'en connaître le contenu complet.

Le XML est insuffisant !!

- Il fournit une syntaxe mais aucune sémantique (importante) lors d'un échange ou de représentation de données à sur le Web.

- Non primitif, des données peuvent être représentées dans différentes manières lors de l'échange ou de représentation des données sur le web → problème dans la structure, à chaque fois qu'on porte un changement sur la structure, on doit utiliser le XSL pour faire les transformations.

→ **Naissance du RDF**

Exercice

```
<?xml version="1.0"?>
<!-- this is a note -->
<note date=3 janvier>
  <to>Bob</To>
  <from>Alice</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
<note date="5 janvier" <!-- this is another note --> >
  <to>Alice</to>
  <from>Bob
  <body>No problem & see you soon</body>
</note>
<note />
```

1. Ce document est-il bien formé (i.e. respecte-t-il la syntaxe XML) ?
2. S'il ne l'est pas, corrigez les erreurs.

Exercice : Solution

Erreurs dans les lignes :
3 : guillemets
4 : casse
9 : une seul racine à l'arborescence, rajouter un élément général document
10 : commentaire à sortir des tags
13 : & interdit
14 : mismatched tag, en fait l'erreur vient de plus haut, il manque la fin du tag <from

Document corrigé

```
<?xml version="1.0"?>
<!-- this is a note -->
<document>
<note date="3 janvier">
<to>Bob</to>
<from>Alice</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
<note date="5 janvier">
<!-- this is another note -->
<to>Alice</to>
<from>Bob</from>
<body>No problem, see you soon</body>
</note>
```

Resource Description Framework : RDF

http://www.w3.org/RDF

Resource Description Framework : RDF

RDF n'est pas à proprement parler un langage. Il s'agit plutôt d'un modèle de données pour décrire et annoter sémantiquement des ressources sur le web. On entend par ressource toute entité que l'on veut décrire sur le web mais qui n'est pas nécessairement accessible sur le web.

- L'information destinée aux applications (pas aux humains) d'extraction d'information, ou aux services web.

- RDF, c'est un moyen d'exprimer des relations. Ces relations sont décrites sous forme de graphe. Chaque nœud du graphe est une ressource ou une valeur. Et chaque nœud est relié à un autre par un arc "nommé« .

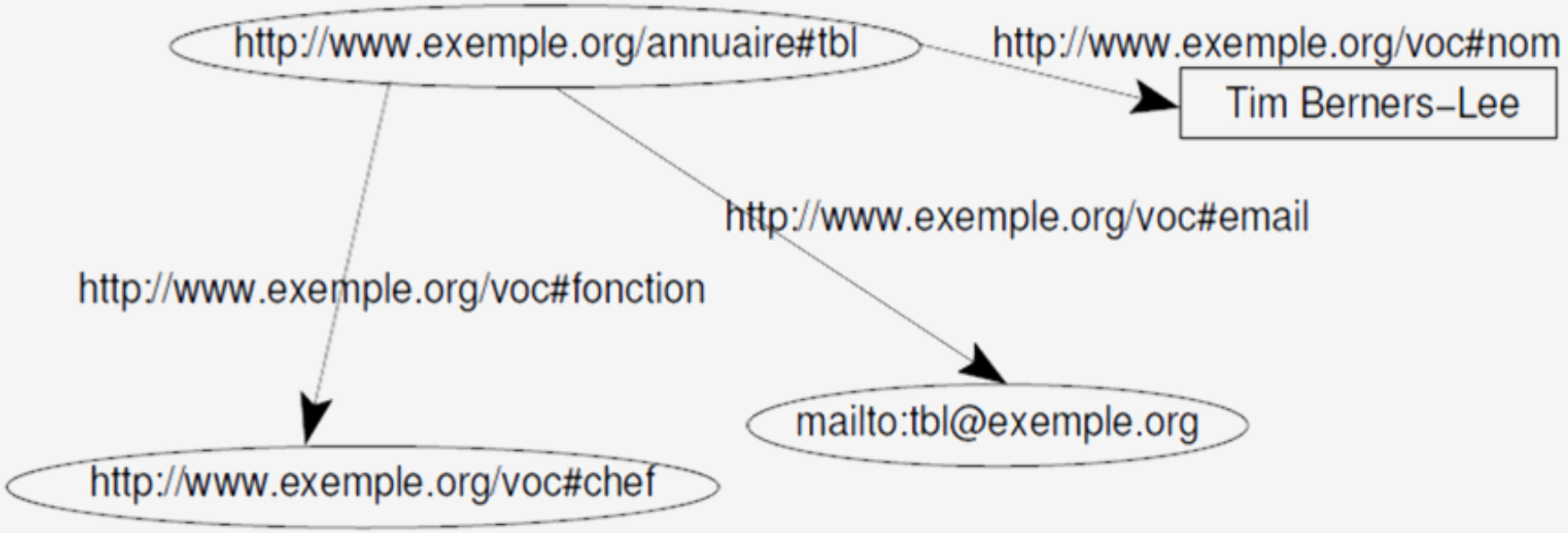
Resource Description Framework : RDF

- Modèle de données pour décrire des ressources du Web.
- Standard W3C depuis 1999.
- Propose donc une interopérabilité entre les applications qui partagent des informations (interprétables par les machines) sur le Web.

Graphe:

- Les nœuds représentent des ressources (concepts, des instances ou les valeurs des propriétés).
- Les arcs représentent des relations(propriétés) entre ces ressources (concepts).
- Les ressources sont représentées par leur URI.

RDF : Représentation graphique



- Ellipse = URI (sujet ou objet)
- Rectangle = Littéral (objet)
- Arc = Prédicat

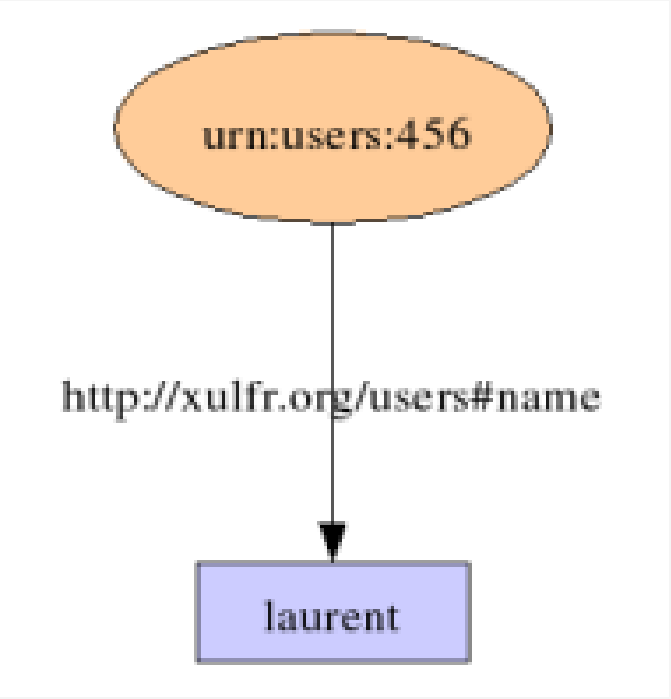
RDF : Exemple

Nous avons ici deux noeuds, "urn:users:456" et "laurent", reliés par un arc nommé "http://xulfr.org/users#name".

En d'autres termes :
urn:users:456 est ce qu'on appelle une ressource, ou encore sujet, source

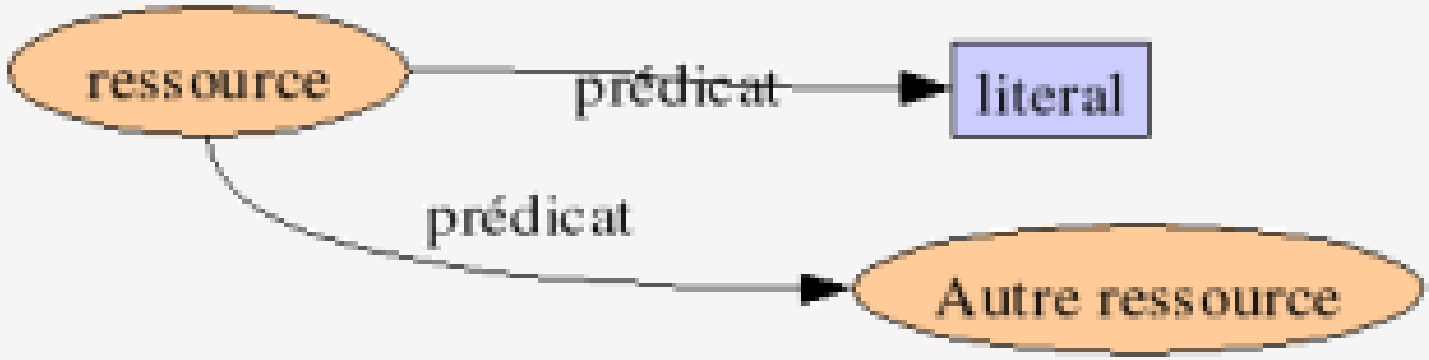
http://xulfr.org/users#name est ce qu'on appelle un prédicat ou encore une propriété

laurent est ce qu'on appelle une valeur ou encore un objet, une cible (target)



RDF : Exemple

Sachant que la cible peut être un littéral comme ici, ou une autre ressource. Toute ressource possède un identifiant sous forme d'URI (URL ou URN). Si une ressource ne possède pas explicitement un identifiant, le logiciel qui lira le RDF en attribuera un implicitement. Les prédicats sont également identifiés par une url.



Attention : les URLs que l'on utilise dans un fichier RDF ne sont pas forcément des URLs web existantes. C'est juste un moyen d'avoir un identifiant unique pour chaque ressource et prédicat.

RDF : Syntaxe

élément **rdf:RDF** contenant les déclarations d'espaces de noms.

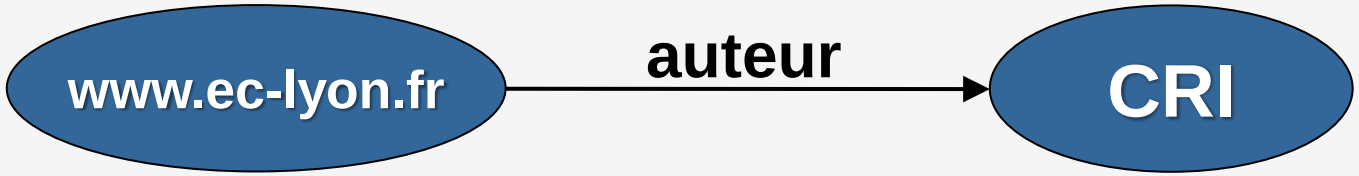
élément **rdf:Description** contient l'URI du sujet dans l'attribut **rdf:about**. Un élément RDF peut contenir plusieurs Description.

élément prédicat ayant pour nom le nom du prédicat et contenant :

- Un attribut **rdf:resource** objet ou
- Un texte emboîté objet

Plusieurs prédicats possibles dans une Description

Exemple RDF



(http://www.ec-lyon.fr, auteur, Cri)

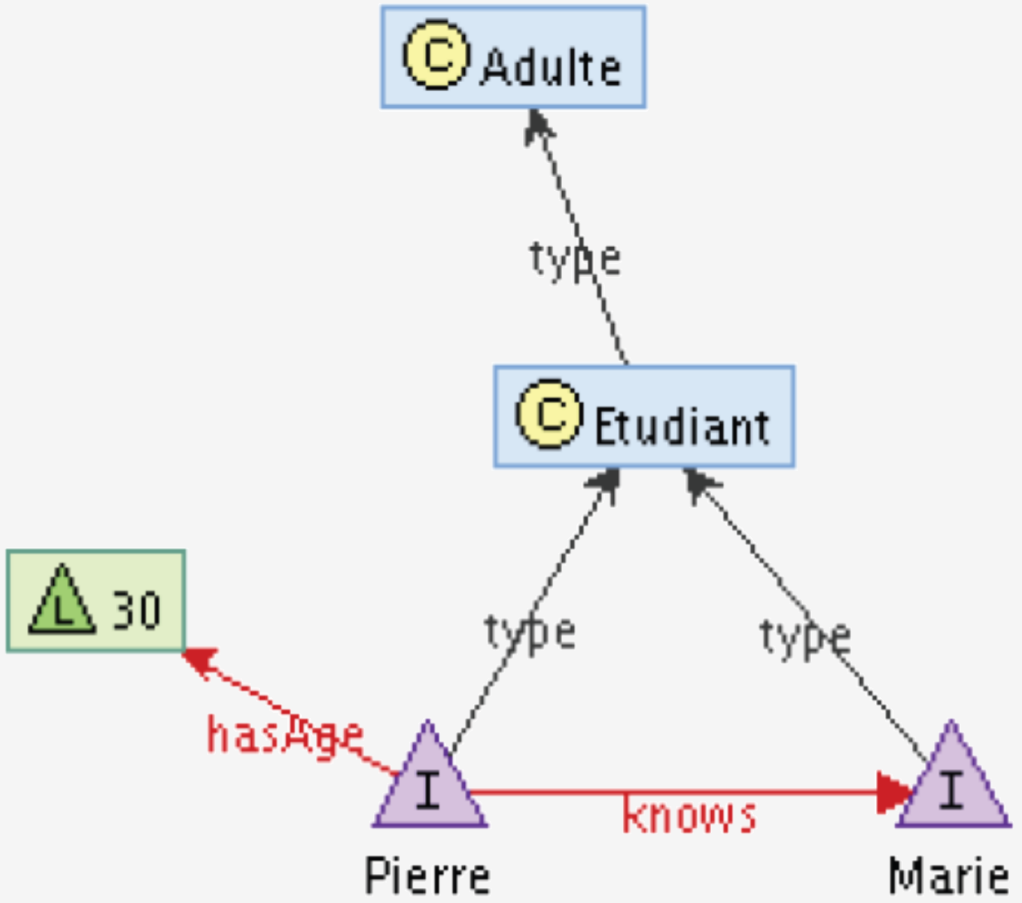
```
<rdf:Description about="http://www.ec-lyon.fr">
    <auteur>"Cri"</auteur>
</rdf:Description>
```

Solution : Graphe RDF

On souhaite modéliser les connaissances suivantes en RDF:

Pierre a 30 ans. Pierre connaît Marie. Pierre et Marie sont des étudiants. Les étudiant sont des Adultes.

Pierre	hasAge	30 (type xsd :nonNegativeInteger)
Pierre	knows	Marie
Pierre	rdf :type	Etudiant
Marie	rdf :type	Etudiant
Etudiant	rdf :type	Adulte



Donnez la représentation graphique de ce graphe RDF ?

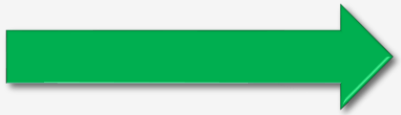
Solution : Représentation en XML

```
<!DOCTYPE rdf :RDF [< !ENTITY xsd "http
://www.w3.org/2001/XMLSchema#">]>
<rdf :RDF xmlns :rdf="http ://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns :ex="http ://td-morec/#">
<rdf :Description rdf :about="http ://td-morec/#Pierre">
<rdf :type>
<rdf :Description rdf :about="http ://td-morec/#Etudiant">
<rdf :type rdf :resource="http ://td-morec/#Adulte"/>
</rdf :Description>
</rdf :type>
<ex :hasAge rdf :datatype="&xsd ;nonNegativeInteger">30</ex :hasAge>
<ex :knows>
<ex :Etudiant rdf :about="http ://td-morec/#Marie"/>
</ex :knows>
</rdf :Description>
</rdf :RDF>
```

Insuffisances du RDF

RDF permet de représenter des déclarations de propriétés sur des ressources mais ne permet pas d'exprimer des connaissances sur les propriétés ou sur les types de ressources :

1. Quelles sont les propriétés autorisées sur un type de ressources ?
2. Quelles sont les valeurs autorisées pour une propriété ?
3. Quels sont les liens entre les types de ressources (généralisation / spécialisation) ?



Naissance du RDF(S)

SPARQL Protocol And RDF Query Language

http://www.w3.org/TR/rdf-sparql-query/

SPARQL

SPARQL est un langage de requêtes du W3C pour RDF/RDFS,

Implémenté dans plusieurs outils utilisables en ligne, en local ou depuis un langage de programmation.

Recommandation du W3C :

- **SPARQL 1.0** depuis 15 janvier 2008.
- **SPARQL 1.1** depuis 26 Mars 2013.

La structure d'une requête SPARQL est très similaire à celle employée dans le langage SQL.

Requêtes SPARQL

Requête Select : Renvoie des t-uples où chaque composante d'un t-uple correspond à la valuation d'une variable de la requête par un terme RDF.

Requête Construct, Describe : Renvoie un graphe RDF utilisant les valuations des variables présentes dans la requête.

Requête Ask : Renvoie un booléen en fonction de la satisfiabilité de la requête.

Forme principale

```
PREFIX foaf:   <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE
  { ?x foaf:name ?name .
    ?x foaf:mbox ?mbox }
```

Résultat

name	mbox
"Johnny Lee Outlaw"	<mailto:jlow@example.com>
"Peter Goodguy"	<mailto:peter@example.org>

Exemple : Donner les ?t et ?p tels que ?p est le dessinateur de ?t.

```
PREFIX bd: <http://www.collection.com/bd#>

SELECT * WHERE {
{ ?t bd:dessinateur ?p }
}
```

```
<?xml version="1.0"?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <variable name="t"/>
    <variable name="p"/>
  </head>
  <results>
    <result>
      <binding name="t"><uri>http://www.collection.com/bd/serie/Laufeust
de Troy/Thanos l incongru</uri></binding>
      <binding name="p"><uri>http://www.collection.com/personne/Tarquin</
uri></binding>
    </result>
    <result>
      <binding name="t"><uri>http://www.collection.com/bd/Laufeust de
Troy/L ivoire du Magohamoth</uri></binding>
      <binding name="p"><uri>http://www.collection.com/personne/Tarquin</
uri></binding>
    </result>
  </results>
</sparql>
```

Forme principale :en 3 clauses (SQL)

La clause **SELECT** permet de choisir les variables du résultat, parmi les variables de la clause WHERE.

La clause **WHERE** contient des triplets qui définissent les conditions de sélection.

La clause **FROM** depuis quelle base. Optionnel, peut être précisé au moment de l'exécution de la requête.

Formes de requête

Il existe également d'autres éléments dans le langage SPARQL qui permettent de spécifier des préfixes (**PREFIX**), **des conditions (FILTER), des disjonctions (UNION), des filtres sur la** production des résultats (**LIMIT et OFFSET**).

Le terme **UNION** dans une clause WHERE permet de spécifier des disjonctions entre ensembles de contraintes, décrits dans des blocs {...}.

Le terme **FILTER** dans une clause WHERE permet de contraindre certaines variables d'une requête SPARQL.

Il est possible de limiter le nombre de réponses à afficher à l'utilisateur grâce aux termes **LIMIT p** et **OFFSET q** (avec **p>0** et **q>0**) situés après la clause WHERE. Le terme LIMIT permet de spécifier le nombre maximum p de résultats à afficher, tandis que le terme OFFSET débute l'affichage des réponses au q-ième résultats.

Requête SPARQL

De plus, une requête SPARQL peut avoir d'autres finalités que de fournir un ensemble de correspondances aux variables spécifiées dans le SELECT.

En effet, dans le langage SPARQL, il est possible de demander si une requête dispose d'au moins une solution. Pour ce faire, le **SELECT** est remplacé par un **ASK**. **Il est aussi possible de construire un nouveau graphe RDF via le terme CONSTRUCT.**

Variables

Une variable dans une requête commence par le caractère "?" ou « $" et ce caractère ne fait pas partie du nom de la variable. Le nom d'une variable :

Ne doit pas commencer par un chiffre

Est sensible à la casse

Ne doit pas contenir d’espace

Doit être signifiant, car il sert de nom de colonne dans le résultat SPARQL.

SELECT ?name ?mbox

Littéraux

En général, chaine de caractères entre simples quottes ou guillemets.

Pour simplifier,

- 1 est synonyme de "1"^^xsd:integer
- 1.3 est synonyme de "1.3"^^xsd:decimal
- true est synonyme de "true"^^xsd:boolean

Outil d'exécution des requêtes

Télécharger l'outil **Twinkle** (*http://www.ldodds.com/projects/twinkle/*).

Exécuter le programme.

Exécuter une requête en appuyant sur « ***Write simple Query*** ».

Dans le champ data URL mettez le chemin de votre fichier XML/RDF que vous voulez interroger.

Dans le champ requête mettez votre requête et faites **Run**: