

# Ontologies et

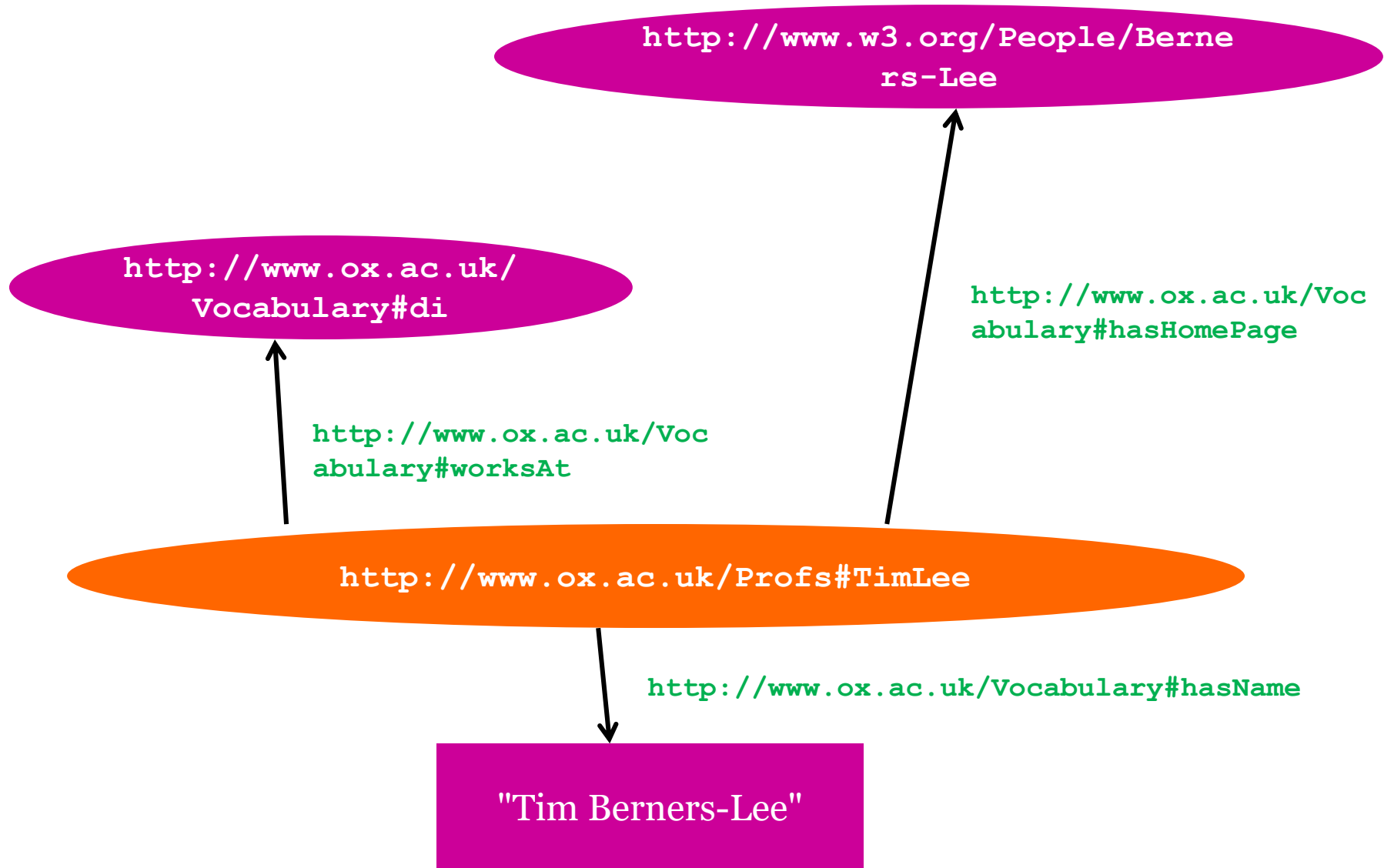
# Web Sémantique

**Décrire avec RDF – Suite + RDFS**

# Plan du cours

1. RDF, c'est quoi ?
2. Syntaxe abstraite : triplets et graphes
3. Syntaxe de sérialisation : N-Triples, Turtle, RDF/XML
4. Valeurs, types, conteneurs, collections, etc.
5. RDF Schema - RDFS

## Syntaxe abstraite - Triplets et Graphes



## Syntaxe de S rialisation - RDF/XML

### 3. RDF/XML

➤ Exemple: Racine

<?xml version="1.0"?>

<rdf:RDF>

</rdf:RDF>

## Syntaxe de Sérialisation - RDF/XML

<?xml version="1.0"?>

<**rdf:RDF**

xmlns:**local**="http://www.ox.ac.uk/Vocabulary#"

xmlns:**prof**="http://www.ox.ac.uk/Profs#"

namespace

xmlns:**rdf**="http://www.w3.org/1999/02/22-rdf-syntax-ns#" >

</**rdf:RDF**>

## Syntaxe de Sérialisation - RDF/XML

```
<?xml version="1.0"?>
```

```
<rdf:RDF
```

```
  xmlns:local="http://www.ox.ac.uk/Vocabulary#"
```

```
  xmlns:prof="http://www.ox.ac.uk/Profs#"
```

```
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
```

```
<rdf:Description>    ...    </rdf:Description>
```

```
<rdf:Description>    ...    </rdf:Description>
```

```
<rdf:Description>    ...    </rdf:Description>
```

Descriptions de  
ressources

```
</rdf:RDF>
```

## Syntaxe de S rialisation - RDF/XML

```
<?xml version="1.0"?>
```

```
<rdf:RDF
```

```
  xmlns:local="http://www.ox.ac.uk/Vocabulary#"
```

```
  xmlns:prof="http://www.ox.ac.uk/Profs#"
```

```
  .... >
```

```
    <rdf:Description    rdf:about="http://www.ox.ac.uk/Profs#TimLee" >
```

```
    </rdf:Description>
```

```
    <rdf:Description    rdf:about="http://www.ox.ac.uk/Profs#TimLee" >
```

```
    </rdf:Description>
```

```
    <rdf:Description    rdf:about="http://www.ox.ac.uk/Profs#TimLee" >
```

```
    </rdf:Description>
```

```
</rdf:RDF>
```

## Syntaxe de Sérialisation - RDF/XML

```
<?xml version="1.0"?>
```

```
<rdf:RDF
```

```
  xmlns:local="http://www.ox.ac.uk/Vocabulary#"
  xmlns:prof="http://www.ox.ac.uk/Profs#"
  .... >
```

```
  <rdf:Description    rdf:about="http://www.ox.ac.uk/Profs#TimLee" >
    <local:hasName>Tim Berners-Lee</local:hasName>
  </rdf:Description>
```

```
  <rdf:Description    rdf:about="http://www.ox.ac.uk/Profs#TimLee" >

  </rdf:Description>
```

```
  <rdf:Description    rdf:about="http://www.ox.ac.uk/Profs#TimLee" >

  </rdf:Description>
```

```
</rdf:RDF>
```



## Syntaxe de Sérialisation - RDF/XML

```
<?xml version="1.0"?>
```

```
<rdf:RDF
```

```
  xmlns:local="http://www.ox.ac.uk/Vocabulary#"
  xmlns:prof="http://www.ox.ac.uk/Profs#"
  .... >
```

```
  <rdf:Description    rdf:about="http://www.ox.ac.uk/Profs#TimLee" >
    <local:hasName>Tim Berners-Lee</local:hasName>
  </rdf:Description>
```

```
  <rdf:Description    rdf:about="http://www.ox.ac.uk/Profs#TimLee" >
    <local:worksAt rdf:resource="http://www.ox.ac.uk/Vocabulary#di"/>
  </rdf:Description>
```

```
  <rdf:Description    rdf:about="http://www.ox.ac.uk/Profs#TimLee" >

  </rdf:Description>
```

```
</rdf:RDF>
```

## Syntaxe de Sérialisation - RDF/XML

```
<?xml version="1.0"?>
```

```
<rdf:RDF
```

```
  xmlns:local="http://www.ox.ac.uk/Vocabulary#"
  xmlns:prof="http://www.ox.ac.uk/Profs#"
  .... >
```

```
  <rdf:Description    rdf:about="http://www.ox.ac.uk/Profs#TimLee" >
    <local:hasName>Tim Berners-Lee</local:hasName>
  </rdf:Description>
```

```
  <rdf:Description    rdf:about="http://www.ox.ac.uk/Profs#TimLee" >
    <local:worksAt rdf:resource="http://www.ox.ac.uk/Vocabulary#di"/>
  </rdf:Description>
```

```
  <rdf:Description    rdf:about="http://www.ox.ac.uk/Profs#TimLee" >
    <local:hasHomePage rdf:resource="http://www.w3.org/People/Berners-Lee"/>
  </rdf:Description>
```

```
</rdf:RDF>
```

## Syntaxe de Sérialisation - RDF/XML

### Abréviation 1 : Combiner

```
<rdf:Description    rdf:about="http://www.ox.ac.uk/Profs#TimLee" >
  <local:hasName>Tim Berners-Lee</local:hasName>
  <local:worksAt rdf:resource=" http://www.ox.ac.uk/Vocabulary#di"/>
  <local:hasHomePage rdf:resource=" http://www.w3.org/People/Berners-Lee"/>
</rdf:Description>
```

---

# Syntaxe de Sérialisation - RDF/XML

## Abréviation 2 : Préfixes des ressources

```
<!DOCTYPE rdf:RDF [<!ENTITY local "http://www.polymtl.ca/Vocabulary#">
    <!ENTITY prof "http://www.polymtl.ca/Profs#">]>

<rdf:RDF ... >
  <rdf:Description rdf:about="&prof;TimLee" >
    <local:hasName>Tim Berners-Lee</local:hasName>
    <local:worksAt rdf:resource="&local;di"/>
    <local:hasHomePage rdf:resource="http://www.w3.org/People/Berners-Lee"/>
  </rdf:Description>
</rdf:RDF>
```

## Syntaxe de S rialisation - RDF/XML

### Abr viation 3 : URI de base et rdf:ID

<rdf:RDF

xmlns:local="http://www.ox.ac.uk/Vocabulary#"

**xml:base**="http://www.ox.ac.uk/Profs"

.... >

<rdf:Description rdf:ID="TimLee" >

<local:hasName>Tim Berners-Lee</local:hasName>

<local:worksAt rdf:resource="http://www.ox.ac.uk/Vocabulary#di"/>

<local:hasHomePage rdf:resource="http://www.w3.org/People/Berners-Lee"/>

</rdf:Description>

</rdf:RDF>

## Syntaxe de S rialisation - RDF/XML

N ud vide ne peut  tre que le sujet  
ou l'objet, jamais une propri t 

### Repr sentation des n uds vides

`<rdf:Description rdf:nodeID="id"> .... </rdf:Description >`

### Exemple :

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [<!ENTITY local "http://www.polymtl.ca/Vocabulary#">
    <!ENTITY prof "http://www.polymtl.ca/Profs#">]>
<rdf:RDF ... >
  <rdf:Description rdf:nodeID="id01">
    <local:hasName>Tim BernersLee</local:hasName>
    <local:worksAt rdf:resource="&local;di"/>
    <local:hasHomePage rdf:resource="http://www.w3.org/People/Berners-Lee"/>
  </rdf:Description>
</rdf:RDF>
```

## Syntaxe de S rialisation - RDF/XML

N ud vide ne peut  tre que le sujet  
ou l'objet, jamais une propri t 

### Repr sentation des n uds vides

`<rdf:Description rdf:nodeID="id"> .... </rdf:Description >`

### Exemple :

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
  <!ENTITY local "http://www.polymtl.ca/Vocabulary#">
  <!ENTITY prof "http://www.polymtl.ca/Profs#">]
<rdf:RDF ... >
  <rdf:Description rdf:nodeID="id01">
    <local:hasName>Tim BernersLee</local:hasName>
  </rdf:Description>
  <rdf:Description rdf:nodeID="id01">
    <local:worksAt rdf:resource=" &local;di" />
  </rdf:Description>
</rdf:RDF>
```

# Syntaxe de Sérialisation - RDF/XML

## Représentation des Littéraux - Types

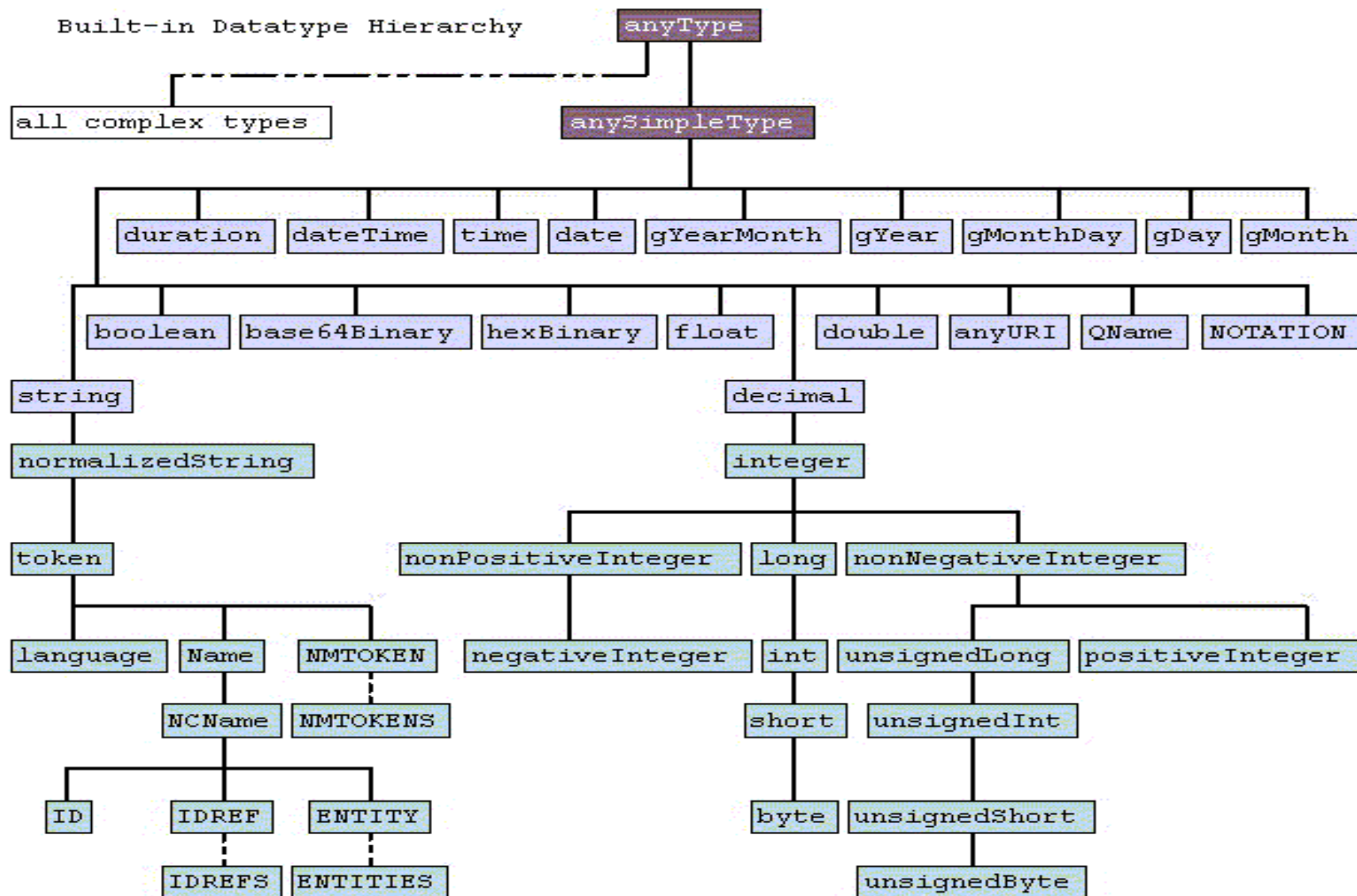
- Les littéraux standards sont des chaînes de caractères.
- Les littéraux typés peuvent être de différents types : date, integer, etc.
- « XML schema datatypes » pour typer les littéraux.

<http://www.w3.org/2001/XMLSchema#>

- `rdf:datatype`



# Syntaxe de Sérialisation - RDF/XML



# Syntaxe de Sérialisation - RDF/XML

## Représentation des Littéraux - Types

### ➤ Exemple : date

```
<rdf:Description  rdf:about="http://www.w3.org/People/Berners-Lee">  
  <local:lastUpdate  
    rdf:datatype="http://www.w3.org/2001/XMLSchema#date">  
    2017-09-04  
  </local:lastUpdate>  
</rdf:Description>
```

<http://www.w3.org/People/Berners-Lee>

<http://www.ox.ac.uk/Vocabulary#lastUpdate>

"2017-09-04"^^xsd:date

# Syntaxe de Sérialisation - RDF/XML

## Représentation des Littéraux - Types

### ➤ Exemple : integer

```
<rdf:Description  rdf:about="http://www.w3.org/People/Berners-Lee">  
  <local:numberWords  
    rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">  
      12345  
    </local:numberWords>  
</rdf:Description>
```

http://www.w3.org/People/Berners-Lee

http://www.ox.ac.uk/Vocabulary#numberWords

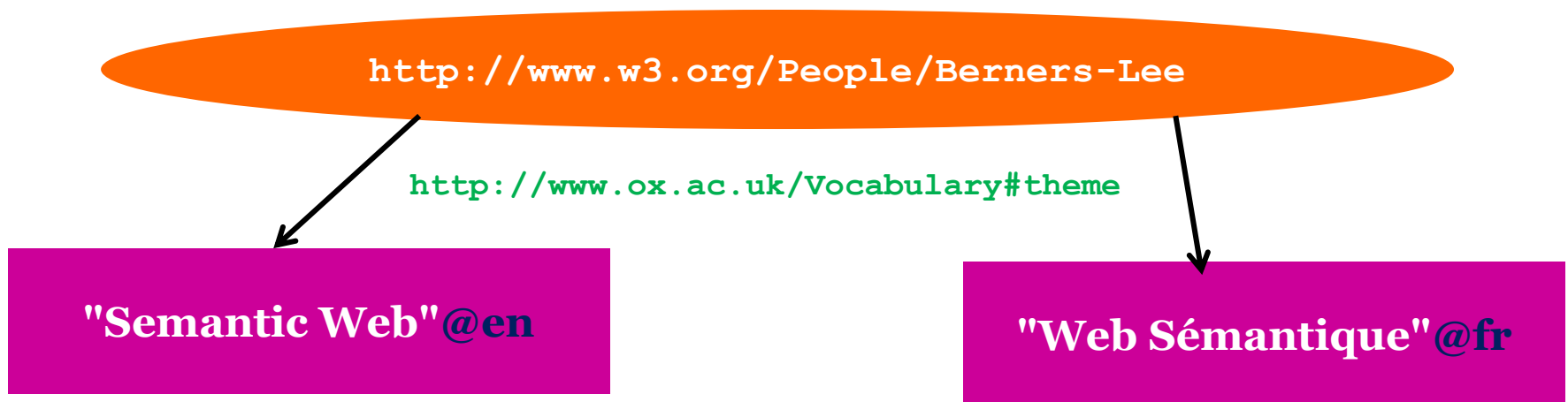
"12345"^^xsd:integer

## Syntaxe de Sérialisation - RDF/XML

### Représentation des Littéraux - Types

- Exemple : les littéraux peuvent être associés à une langue .

```
<rdf:Description  rdf:about="http://www.w3.org/People/Berners-Lee">  
  <local:theme  xml:lang='fr'>Web Sémantique</local:theme>  
  <local:theme  xml:lang='en'>Semantic Web</local:theme>  
</rdf:Description>
```



## Syntaxe de Sérialisation - RDF/XML

### Identification des types de ressources

- Les ressources n'entrent pas toutes dans la même catégorie.
- Exemple: professeur, département universitaire, page personnelle, etc. sont des entités de types différents. Distinguer ces différents types.
- En RDF → ajouter la propriété suivante entre une ressource et son type :

<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>

---

<http://www.ox.ac.uk/Vocabulary#di>

<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>

<http://www.ox.ac.uk/Vocabulary#AcademicDepartment>

# Syntaxe de Sérialisation - RDF/XML

## Identification des types de ressources

- Les ressources n'entrent pas toutes dans la même catégorie.
- Exemple: professeur, département universitaire, page personnelle, etc. sont des entités de types différents.
- Distinguer ces différents types.
- En RDF → ajouter la propriété suivante entre une ressource et son type :

**<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>**

```
<rdf:Description  rdf:about="http://www.ox.ac.uk/Vocabulary#di">  
  <rdf:type rdf:resource= .../Vocabulary#AcademicDepartement/>  
</rdf:Description>
```

## Syntaxe de Sérialisation - RDF/XML

### Identification des types de ressources

- Les ressources n'entrent pas toutes dans la même catégorie.
- Exemple: professeur, département universitaire, page personnelle, etc. sont des entités de types différents.
- Distinguer ces différents types.
- En RDF → ajouter la propriété suivante entre une ressource et son type :

**<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>**

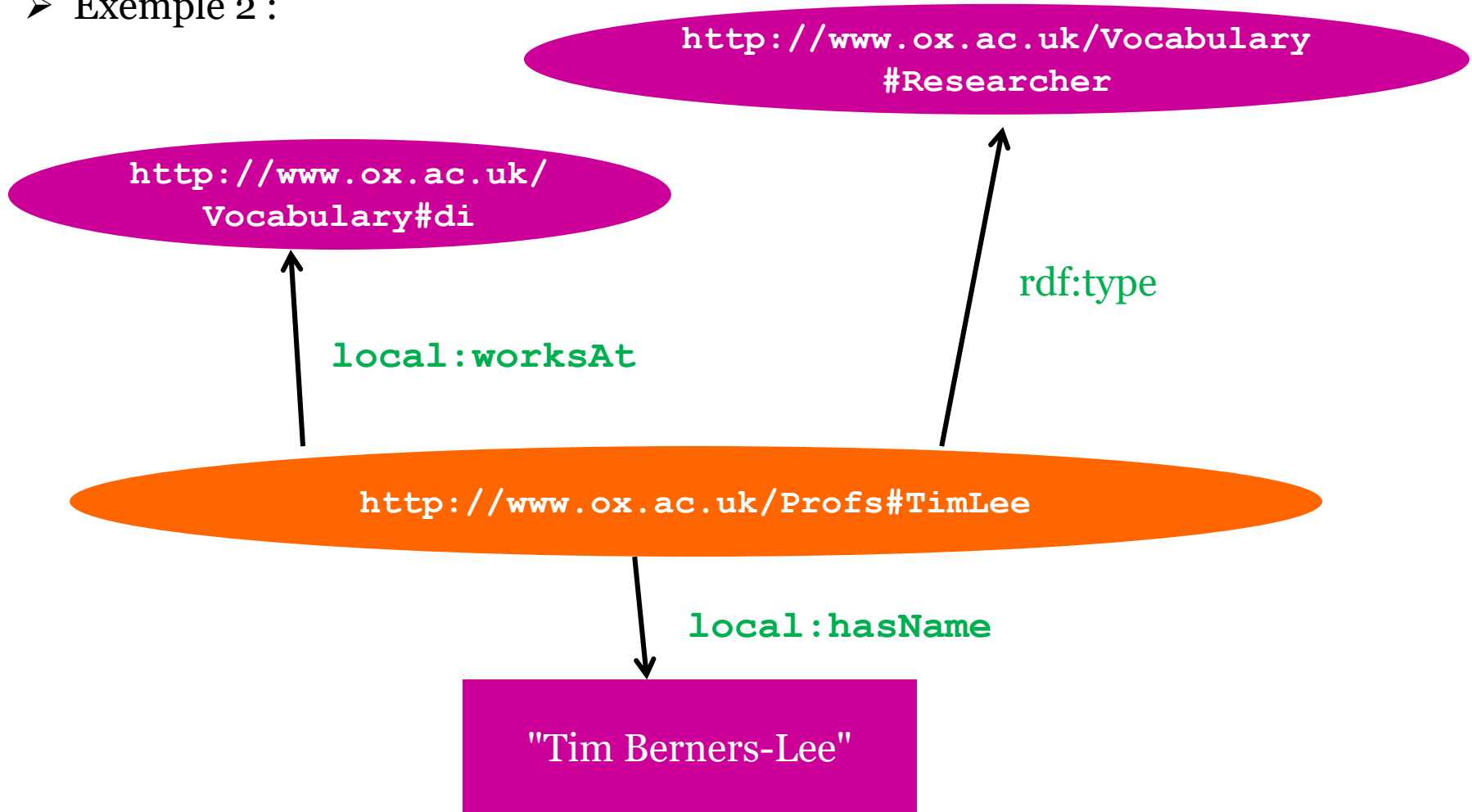
En Turtle :

**local:di** a **local:AcademicDepartment** .

## Syntaxe de Sérialisation - RDF/XML

### Identification des types de ressources

#### ➤ Exemple 2 :





## Syntaxe de Sérialisation - RDF/XML

```
<?xml version="1.0"?>
```

```
<rdf:RDF
```

```
  xmlns:local="http://www.ox.ac.uk/Vocabulary#"
  xmlns:prof="http://www.ox.ac.uk/Profs#"
  .... >
```

```
  <rdf:Description    rdf:about="http://www.ox.ac.uk/Profs#TimLee" >
```

```
    <rdf:type  rdf:resource="http://www.ox.ac.uk/Vocabulary#Researcher"/>
```

```
    <local:hasName>Tim Berners-Lee</local:hasName>
```

```
    <local:worksAt  rdf:resource="http://www.ox.ac.uk/Vocabulary#di"/>
```

```
  </rdf:Description>
```

```
</rdf:RDF>
```

## Syntaxe de Sérialisation - RDF/XML

```
<?xml version="1.0"?>
```

```
<rdf:RDF
```

```
  xmlns:local="http://www.ox.ac.uk/Vocabulary#"
  xmlns:prof="http://www.ox.ac.uk/Profs#"
  .... >
```

```
  <rdf:Description    rdf:about="http://www.ox.ac.uk/Profs#TimLee" >
```

```
    <rdf:type    rdf:resource="http://www.ox.ac.uk/Vocabulary#Researcher"/>
```

```
    <local:hasName>Tim Berners-Lee</local:hasName>
```

```
    { <local:worksAt>
      <rdf:Description    rdf:about="http://www.ox.ac.uk/Vocabulary#di">
        <rdf:type    rdf:resource=".../Vocabulary#AcademicDepartement"/>
      </rdf:Description>
    } </local:worksAt>
  </rdf:Description>
```

```
</rdf:RDF>
```

## Syntaxe de S rialisation - RDF/XML

- **Abbr viation** : Remplacer la balise **Description** par le type de la ressource et retirer **rdf:type** associ     la ressource.

<rdf:RDF

xmlns:local="http://www.ox.ac.uk/Vocabulary#"

xmlns:prof="http://www.ox.ac.uk/Profs#"

.... >

<local:Researcher rdf:about="http://www.ox.ac.uk/Profs#TimLee" >

<local:hasName>Tim Berners-Lee</local:hasName>

<local:worksAt>

<local:AcademicDepartement rdf:about="http://www.ox.ac.uk/Vocabulary#di"/>

</local:worksAt>

</local:Researcher>

</rdf:RDF>

## Syntaxe de Sérialisation - RDF/XML

- **Abréviation** : Remplacer la balise **Description** par le type de la ressource et retirer **rdf:type** associé à la ressource.
- RDF n'interdit pas qu'une ressource ait plus d'un type.
- Mais un seul de ses types pourra faire l'objet de l'abréviation de RDF/XML permettant de remplacer la balise `rdf:Description`.

# Syntaxe de Sérialisation - RDF/XML

## Les Conteneurs

- Une ressource qui contient d'autres ressources. Une liste ouverte.
- Spécifier une ressource et indiquer clairement les relations entre cette ressource et les entités qu'elle contient.
- Exemple : un livre écrit par plusieurs auteurs.
- RDF propose trois classes de conteneur: `rdf:Bag`, `rdf:Seq`, et `rdf:Alt`.
- `rdf:Bag` : conteneur dont les membres n'ont aucun ordre entre eux.
- `rdf:Seq` : conteneur qui suppose l'existence d'un ordre entre les membres.
- `rdf:Alt` : conteneur présentant des alternatives parmi lesquelles on s'attend à ce qu'une seule soit sélectionnée. Ex : même valeur mais dans différentes langues.

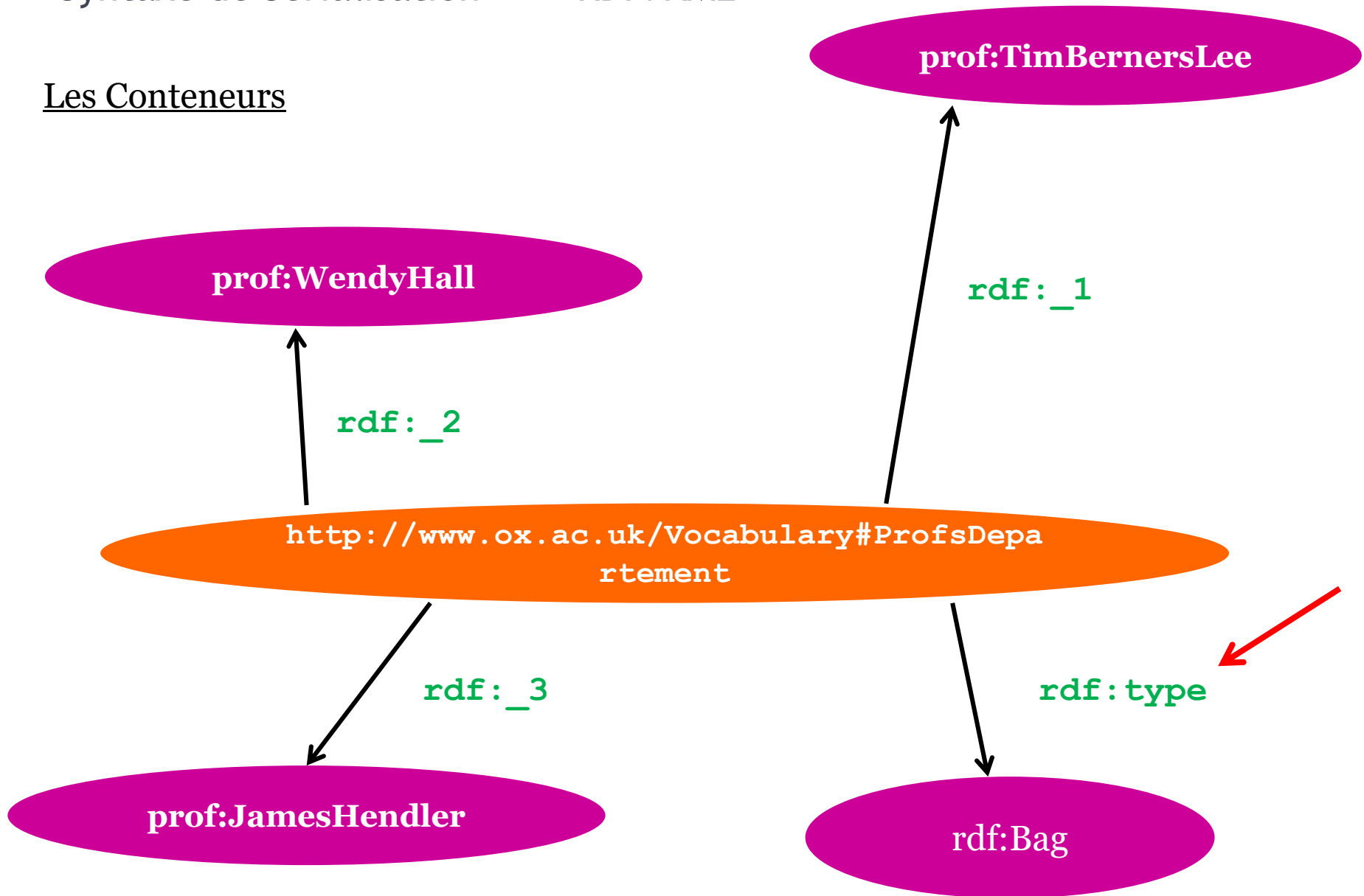
# Syntaxe de S rialisation - RDF/XML

## Les Conteneurs

- Un conteneur est reli      chacun de ses membres par une relation **rdf:\_n**.
- *n* est entier. Il repr  sente l'ordre du membre dans rdf:Seq.
- Il est possible d'avoir deux membres avec la m  me valeur *n*, m  me avec rdf:Seq.
- Il est possible d'avoir des sauts dans la num  rotation. Ex: rdf:\_1, rdf:\_3, rdf:\_4
- Exemple : Un d  partement contenant trois professeurs :

# Syntaxe de S rialisation - RDF/XML

## Les Conteneurs



# Syntaxe de Sérialisation - RDF/XML

## Les Conteneurs

<rdf:RDF

xmlns:local="http://www.ox.ac.uk/Vocabulary#"

xmlns:prof="http://www.ox.ac.uk/Profs#"

xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

.... >

<rdf:Bag rdf:about=".../Vocabulary#ProfsDepartement" >

<rdf:\_1 rdf:resource=".../Profs#TimsBernersLee" />

<rdf:\_2 rdf:resource=".../Profs#WendyHall" />

<rdf:\_3 rdf:resource=".../Profs#JamesHendler" />

</rdf:Bag>

</rdf:RDF>



# Syntaxe de Sérialisation - RDF/XML

## Les Conteneurs

- Abréviation en utilisant *rdf:li* qui génère automatiquement les *rdf:\_n*.

<rdf:RDF

xmlns:local="http://www.ox.ac.uk/Vocabulary#"

xmlns:prof="http://www.ox.ac.uk/Profs#"

xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

.... >

<rdf:Bag rdf:about=".../Vocabulary#ProfsDepartement" >

<rdf:li rdf:resource=".../Profs#TimsBernersLee" />

<rdf:li rdf:resource=".../Profs#WendyHall" />

<rdf:li rdf:resource=".../Profs#JamesHendler" />

</rdf:Bag>

</rdf:RDF>

# Syntaxe de Sérialisation - RDF/XML

## Les Conteneurs - En Turtle

```
@prefix    local: <http://www.ox.ac.uk/Vocabulary#> .
@prefix    prof:  <http://www.ox.ac.uk/Profs#> .
@prefix    rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
```

```
local:ProfsDepartement
    a          rdf:Bag ;
    rdf:_1     prof:TimBernersLee ;
    rdf:_2     prof:WendyHall ;
    rdf:_3     prof:JamesHendler .
```

# Syntaxe de Sérialisation - RDF/XML

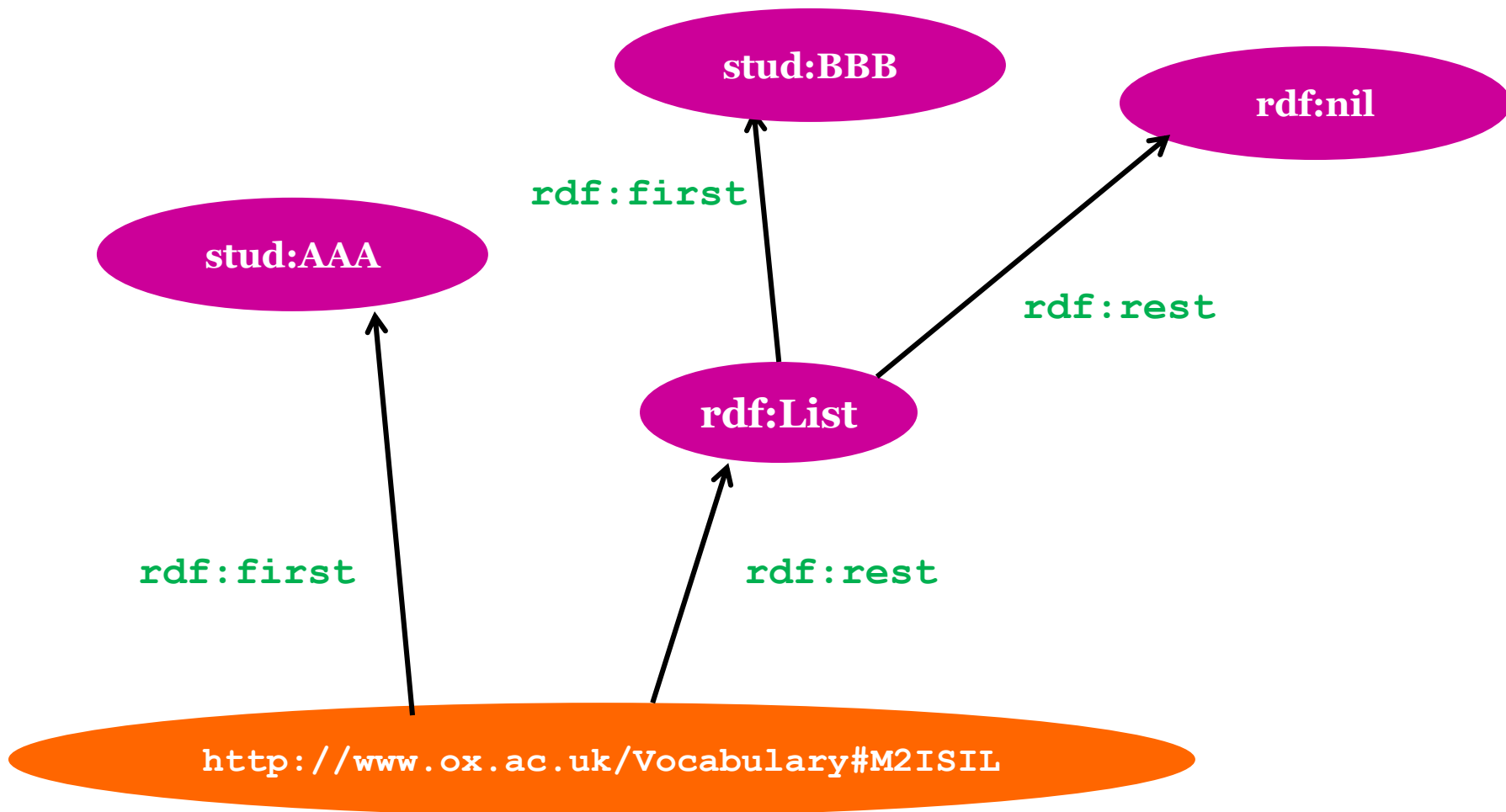
## Les Collections

- Listes exhaustives et ordonnées.
- Ensembles fermés, contrairement aux conteneurs.
- **rdf:List**
- La construction d'une liste se fait de manière récursive en utilisant :
  - ✓ Le prédicat **rdf:first** pour le premier élément de la liste;
  - ✓ Le prédicat **rdf:rest** pour le reste de la liste, qui est lui-même une liste.
  - ✓ La liste vide est représentée par l'URI **rdf:nil**.

# Syntaxe de Sérialisation - RDF/XML

## Les Collections

- Exemple : Les étudiants de Master 2 ISIL sont : AAA et BBB.



# Syntaxe de Sérialisation - RDF/XML

## Les Collections

<rdf:RDF

xmlns:local="http://www.ox.ac.uk/Vocabulary#"

xmlns:stud="http://www.ox.ac.uk/Students#"

xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

.... >

<rdf:Description rdf:about=".../Vocabulary#Master2ISIL" >

<rdf:first rdf:resource=".../Students#AAA" />

<rdf:rest >

<rdf:List >

<rdf:first rdf:resource=".../Students#BBB" />

<rdf:rest rdf:resource=".../22-rdf-syntax-ns#nil" />

</rdf:List >

</rdf:rest >

</rdf:Description>

</rdf:RDF>

# Syntaxe de Sérialisation - RDF/XML

## Les Collections - En Turtle

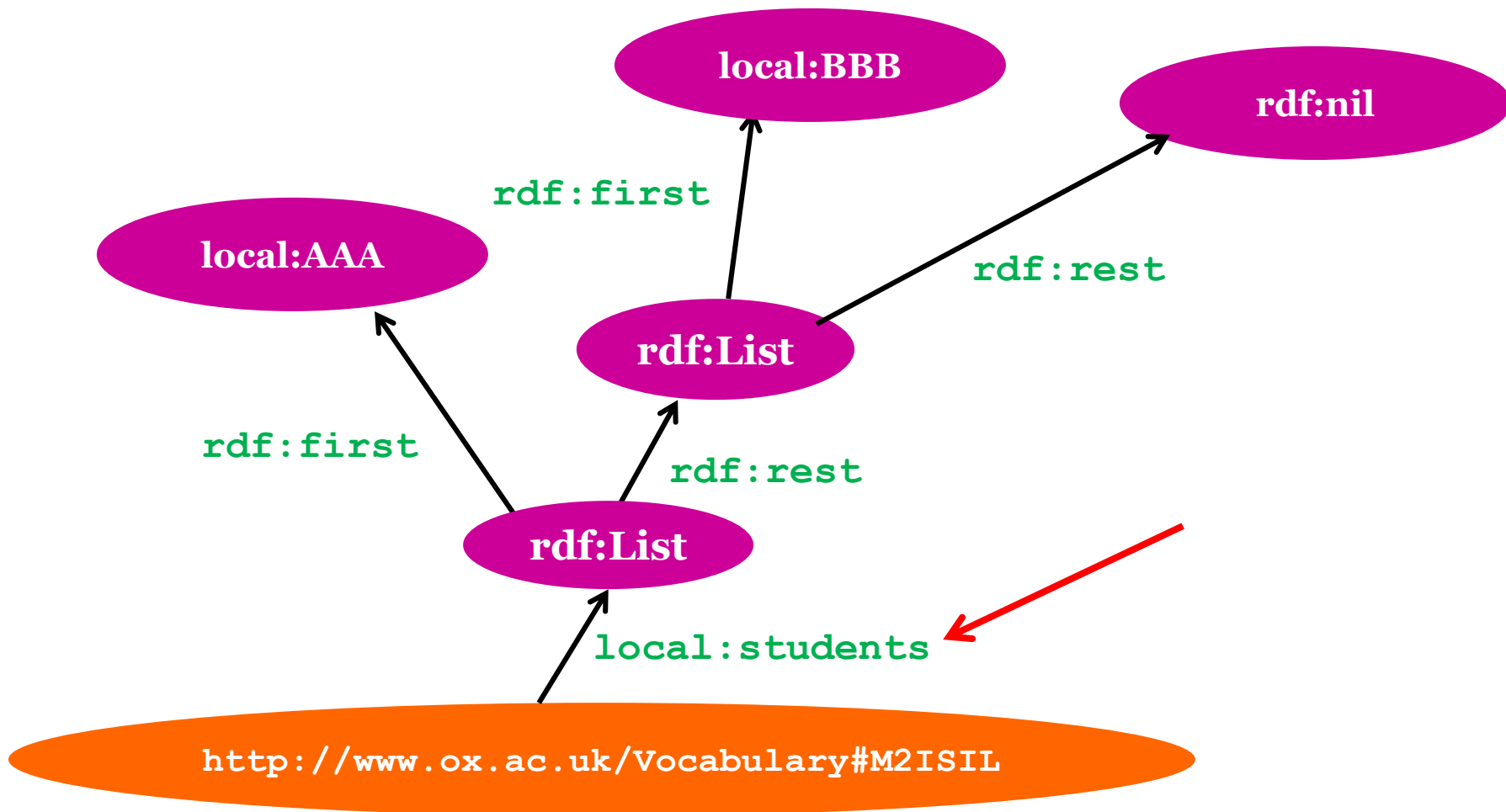
```
@prefix    local: <http://www.ox.ac.uk/Vocabulary#> .
@prefix    stud:  <http://www.ox.ac.uk/Students#> .
@prefix    rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
```

```
local:Master2ISIL
  rdf:List
  (
    stud:AAA
    stud:BBB
  ) .
```

# Syntaxe de Sérialisation - RDF/XML

## Les Collections

- Exemple 2: Les étudiants de Master 2 ISIL sont : AAA et BBB.



## Syntaxe de Sérialisation - RDF/XML

### Les Collections - Abréviation

```
<rdf:Description rdf:about=".../Vocabulary#Master2ISIL">
  <local:students      rdf:parseType="Collection">
    <rdf:Description rdf:about=".../Students#AAA" />
    <rdf:Description rdf:about=".../Students#BBB" />
  </local:students>
</rdf:Description>
```



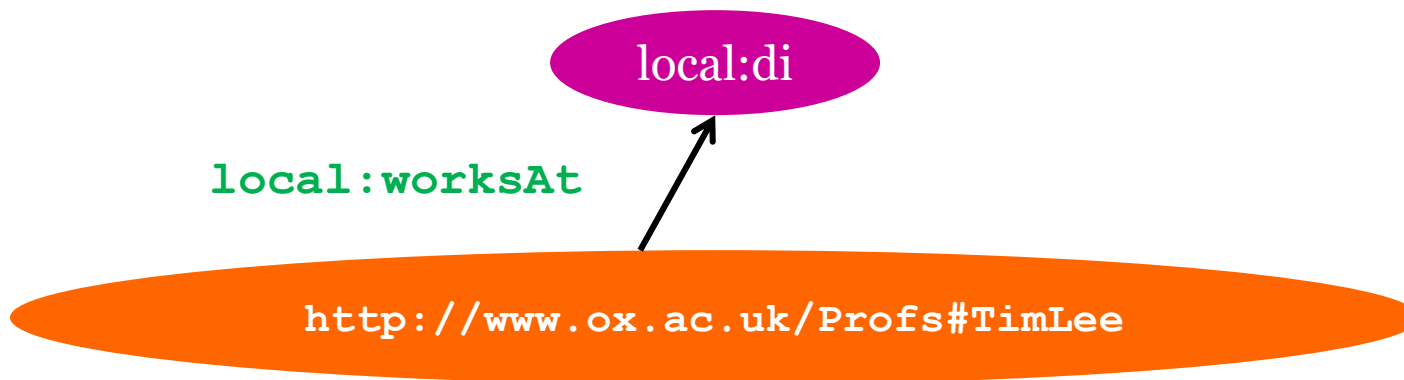
# Syntaxe de S rialisation - RDF/XML

## La R ification

- Les applications RDF ont parfois besoin d'enregistrer des informations sur quand les triplets ont  t  produit, par qui, etc. => Leur provenance.
- RDF fournit un vocabulaire built-in permettant la description de ces triplet.  
***Statements.***
- La description des ces statements utilisant ce vocabulaire est appel e *R ification*.
- Ajouter une ressource de type `rdf:Statement` pour d crire le triplet.
- Pour sp cifier les informations contenues dans ce triplet : utiliser les propri t s `rdf:subject`, `rdf:predicate`, et `rdf:object`.
- Exemple: Pr ciser que le triplet informant sur le lieu de travail a  t  *fourni par Tim Berners-Lee lui-m me*.

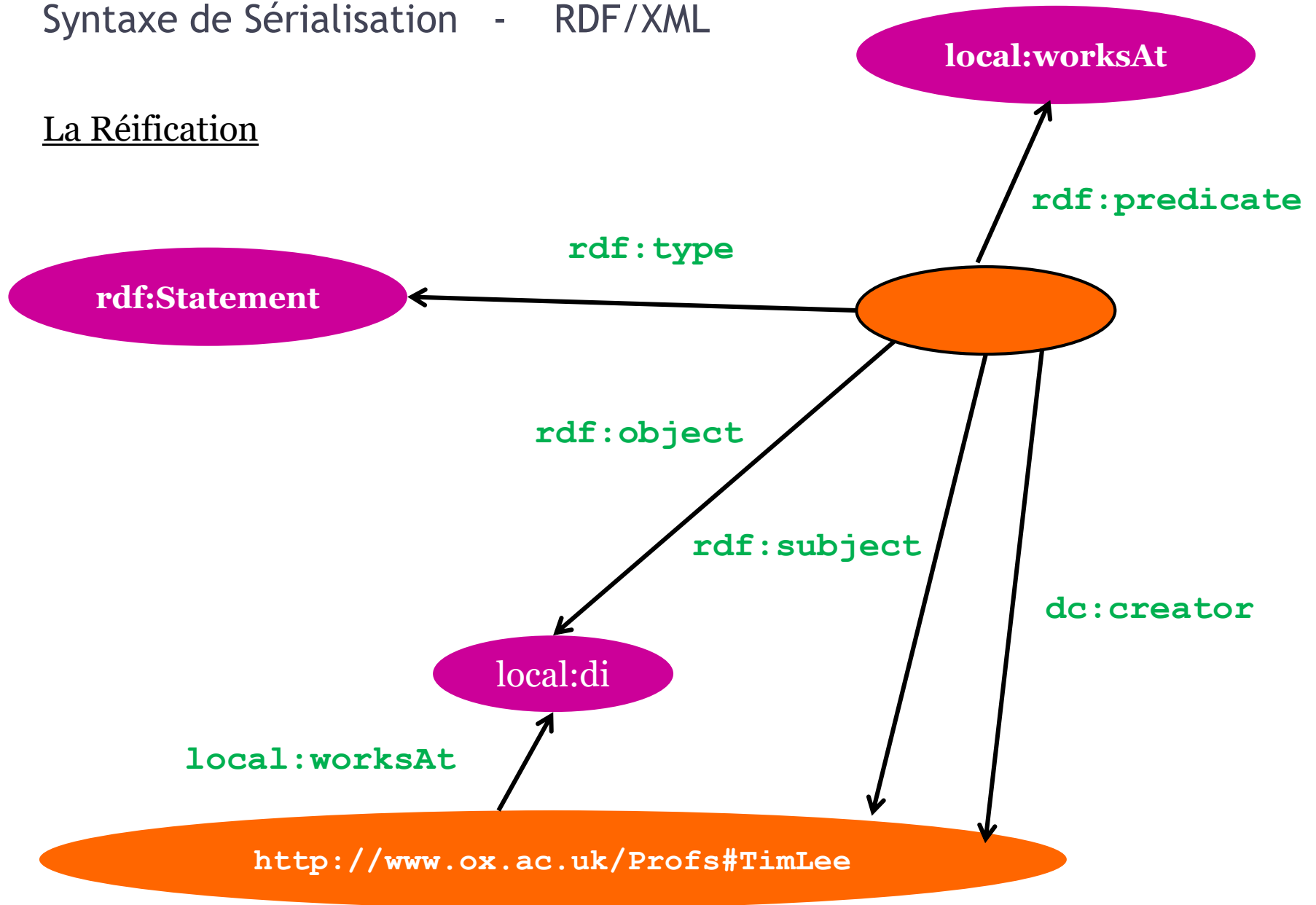
# Syntaxe de Sérialisation - RDF/XML

## La Réification



## Syntaxe de Sérialisation - RDF/XML

### La Réification



# Syntaxe de S rialisation - RDF/XML

## La R ification

```
<rdf:RDF ... >
```

```
<rdf:Description  rdf:about="http://www.ox.ac.uk/Profs#TimLee" >  
  <local:worksAt  rdf:resource="http://www.../Vocabulary#di"/>  
</rdf:Description>
```

```
<rdf:Statement>  
  <rdf:predicate  rdf:resource="http://w.../Vocabulary#worksAt">  
  <rdf:object     rdf:resource="http://w.../Vocabulary#di">  
  <rdf:subject    rdf:resource="http://w.../Profs#TimLee">  
  <dc:creator     rdf:resource="http://w.../Profs#TimLee">  
</rdf:Statement>
```

```
</rdf:RDF>
```

# Syntaxe de Sérialisation - RDF/XML

## La Réification - [Abréviation](#)

```
<rdf:RDF ... >
```

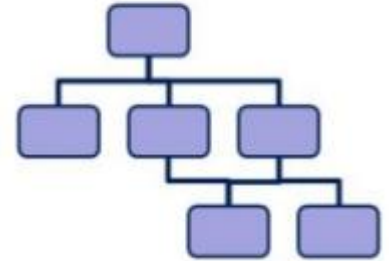
```
<rdf:Description  rdf:about="http://www.ox.ac.uk/Profs#TimLee" >  
  <local:worksAt  rdf:ID="triplet1"  
                  rdf:resource="http://www.../Vocabulary#di"/>  
</rdf:Description>
```

```
<rdf:Statement  rdf:ID="triplet1">  
  <dc:creator  rdf:resource="http://w.../Profs#TimLee">  
</rdf:Statement>
```

```
</rdf:RDF>
```

## RDF Schema - RDFS

- `rdf:type` : pour distinguer le type des ressources décrites.
- Que représente réellement ?
- Comment organiser et hiérarchiser les ressources RDF ?
- Comment donner un sens et une signification aux informations stockées sous forme de triplets RDF ?
- Pour le faire, un vocabulaire a été proposé. Un vocabulaire de métadonnées.
- RDF Schema – RDFS. - RDF Vocabulary Language.
- Le préfix pour tous les éléments du vocabulaire RDFS : **rdfs**

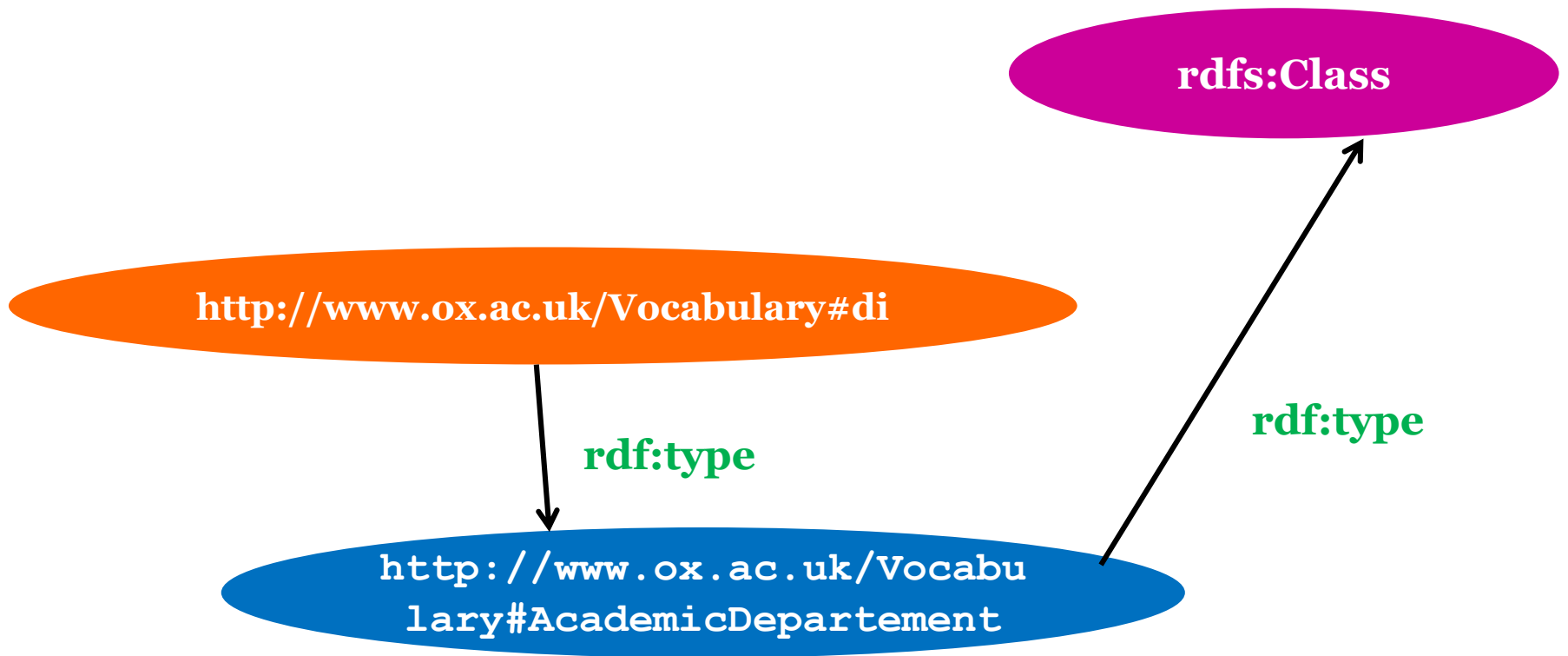


<http://www.w3.org/2000/01/rdf-schema#>

# RDF Schema - RDFS

## ➤ Les Classes

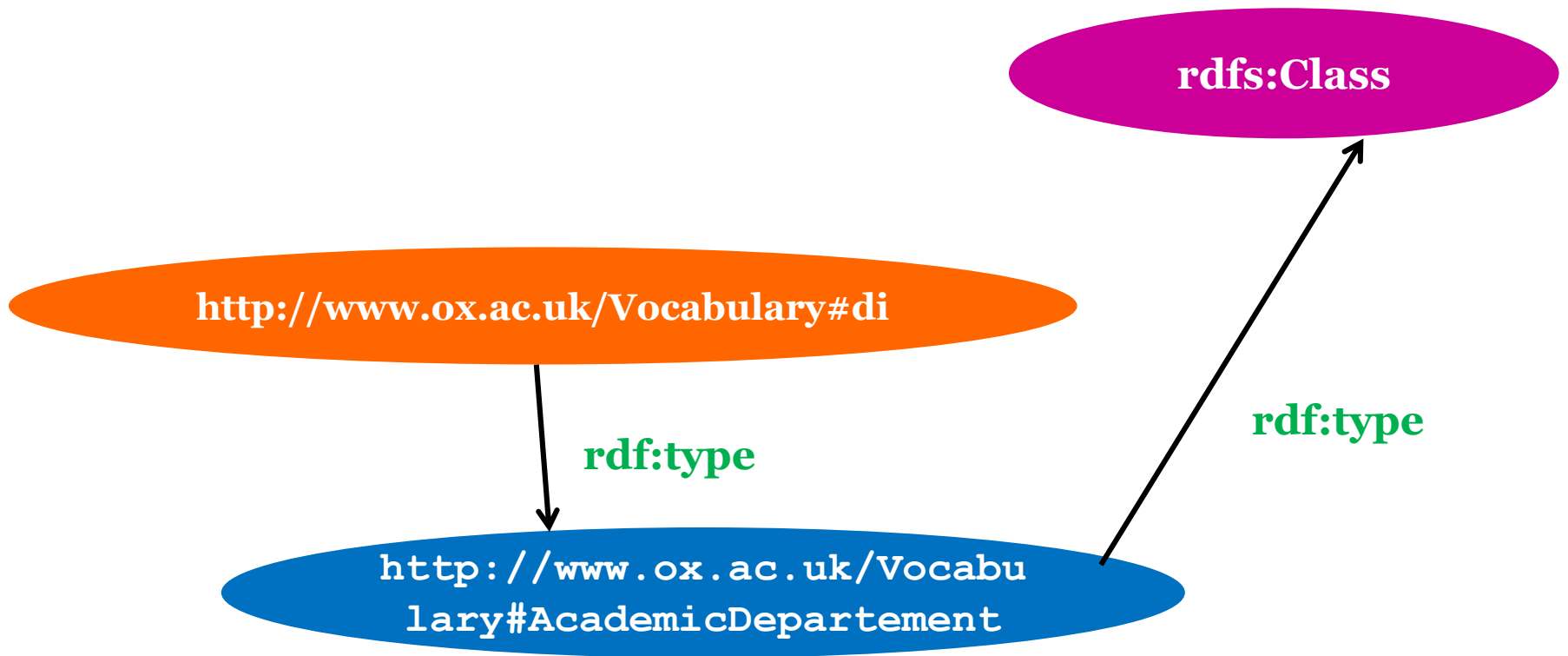
RDFS permet de définir des classes de ressources : **rdfs:Class**



# RDF Schema - RDFS

## ➤ Les Classes

```
<rdfs:Class rdf:about="http://www.ox.ac.uk/Vocabulary#AcademicDepartement" />
```



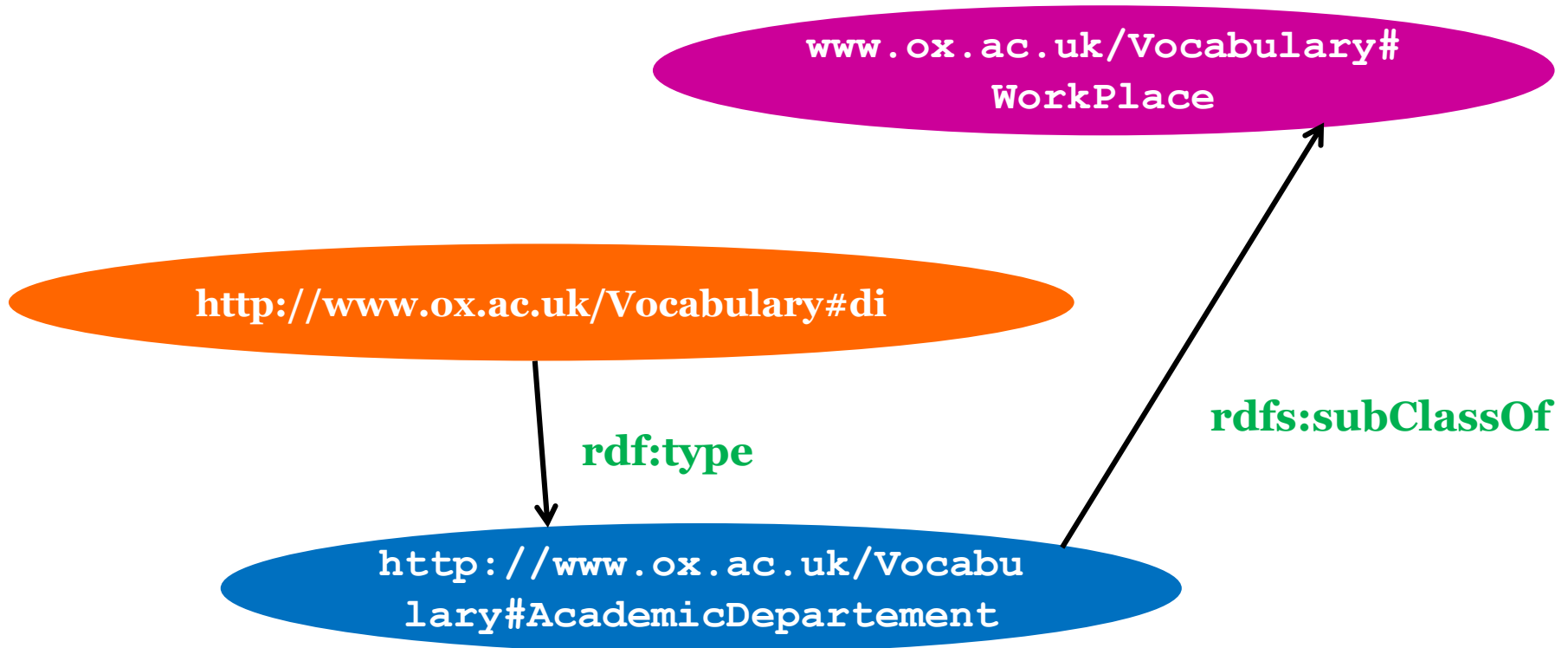


# RDF Schema - RDFS

## ➤ Les Classes

RDFS permet de définir des hiérarchies de classes en utilisant la propriété :

`rdfs:subClassOf`



# RDF Schema - RDFS

## ➤ Les Propriétés

Les ressources ont un(des) type(s). Les propriétés aussi.

**rdf:Property**

<code>local:hasName</code>	<code>rdf:type</code>	<code>rdf:Property</code> .
<code>local:hasHomePage</code>	<code>rdf:type</code>	<code>rdf:Property</code> .
<code>local:worksAt</code>	<code>rdf:type</code>	<code>rdf:Property</code> .

➔      `rdf:type`              `rdf:type`              `rdf:Property` .

Hiérarchie des propriétés :              **rdfs:subPropertyOf**

# RDF Schema - RDFS

## ➤ Les Propriétés

RDFS permet de définir la signature de chaque propriété :

**`rdfs:domain`**

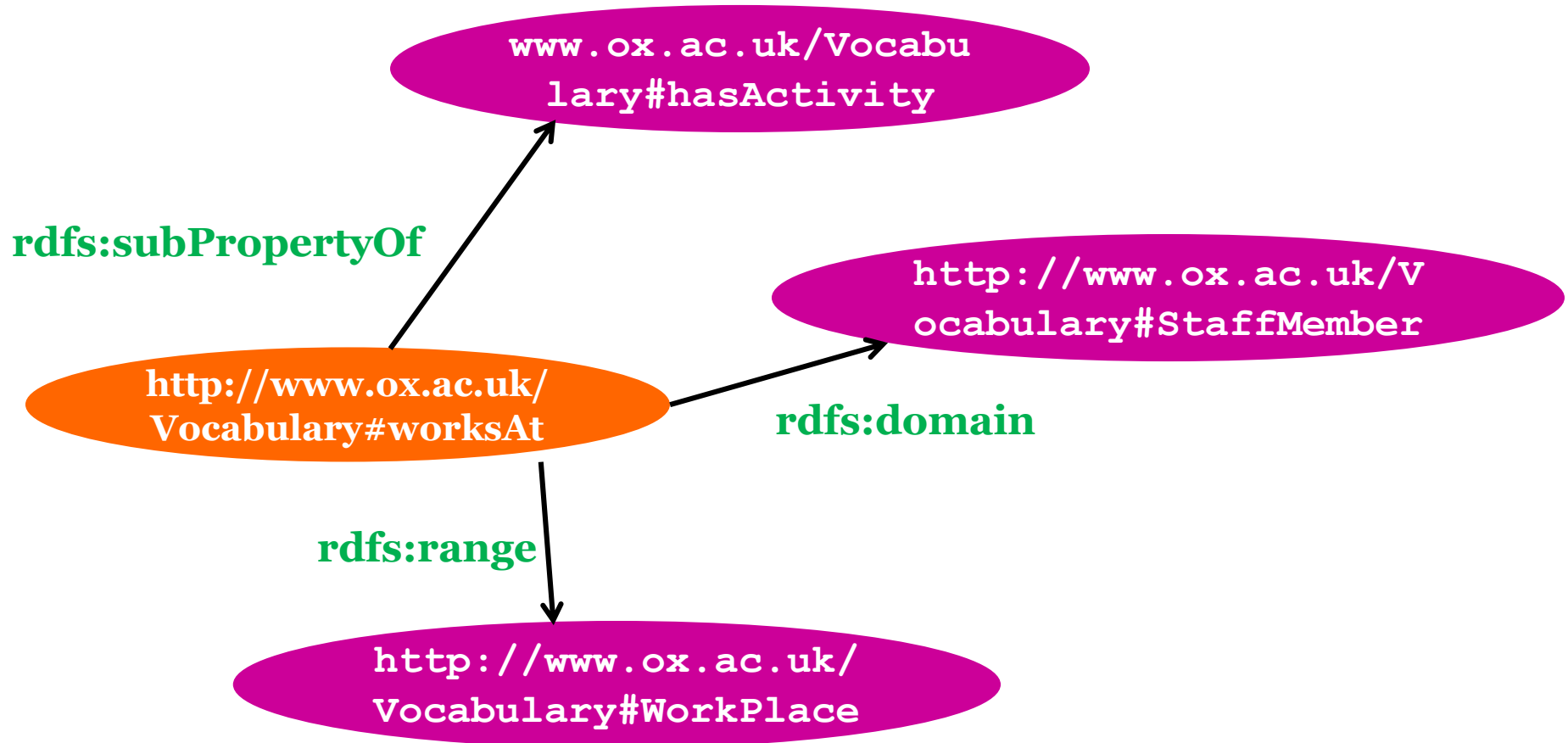
**`rdfs:range`**

Une signature par :

- ✓ le *domain* (domaine) : type de la ressource d'où part la propriété (Sujet);
- ✓ le *range* (co-domaine ou portée) : type de la ressource vers laquelle pointe la propriété (Objet).

# RDF Schema - RDFS

## ➤ Les Propriétés - Exemple



## RDF Schema - RDFS

### ➤ Les Classes / Les Propriétés

- RDFS permet de nommer une ressource avec du texte (en langue naturelle) compréhensible par un humain :

`rdfs:label`

- RDFS permet de commenter une ressource avec du texte (en langue naturelle) compréhensible par un humain :

`rdfs:comment`

## RDF Schema - RDFS

### ➤ Les Classes / Les Propriétés

- RDFS permet de mentionner une ressource contenant d'autres informations sur la ressource sujet :

**`rdfs:seeAlso`**

- RDFS permet d'indiquer l'URI du vocabulaire RDF dans lequel la ressource est décrite :

**`rdfs:isDefinedBy`**

`<http://... bedetheque.com/auteurs/Fmurr > rdfs:seeAlso <https://fr.wikipedia.org/wiki/F'murr > .`

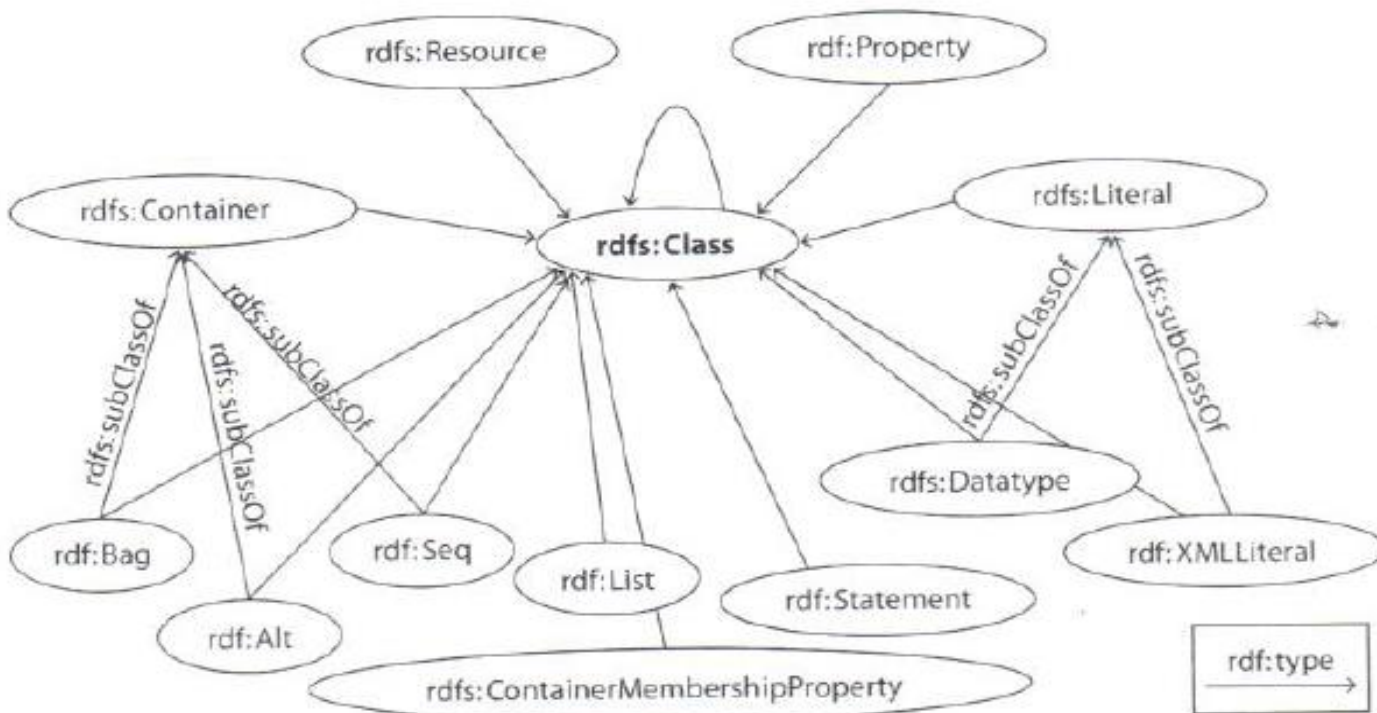
`<http://... bedetheque.com/series/GenieDesAlpages > rdfs:isDefinedBy <http://www.bedetheque.com/> .`

## RDF Schema - RDFS

- La classe des ressources : `rdfs:Resource`
- La classe de tous les littéraux : `rdfs:Literal`
- La classe de tous les types : `rdfs:Datatype`
- `rdf:XMLLiteral`: sous-classe de la classe `rdfs:Literal` et une instance de `rdfs:Datatype` : représente les valeurs codées en XML.

## RDF Schema - RDFS

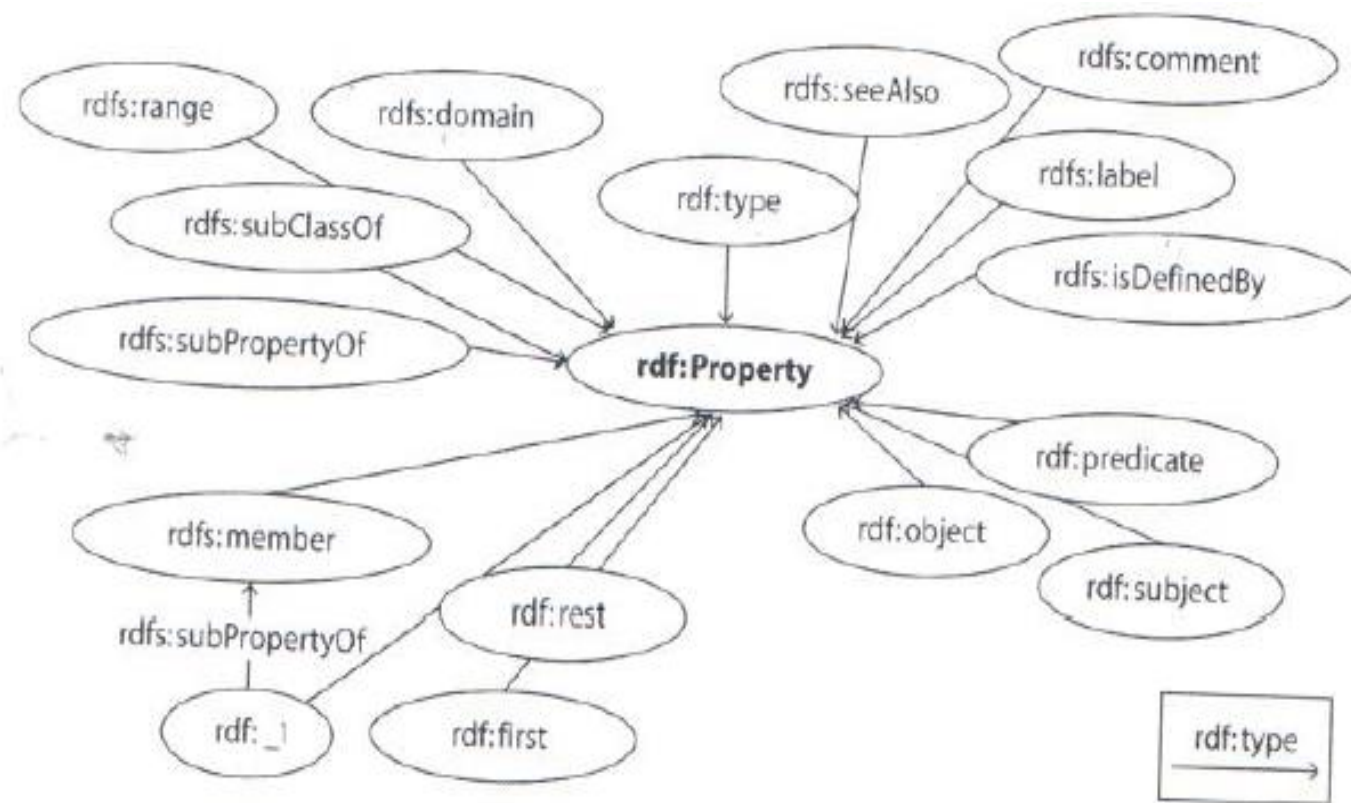
### ➤ Méta-modèle RDFS : déclaration de classes





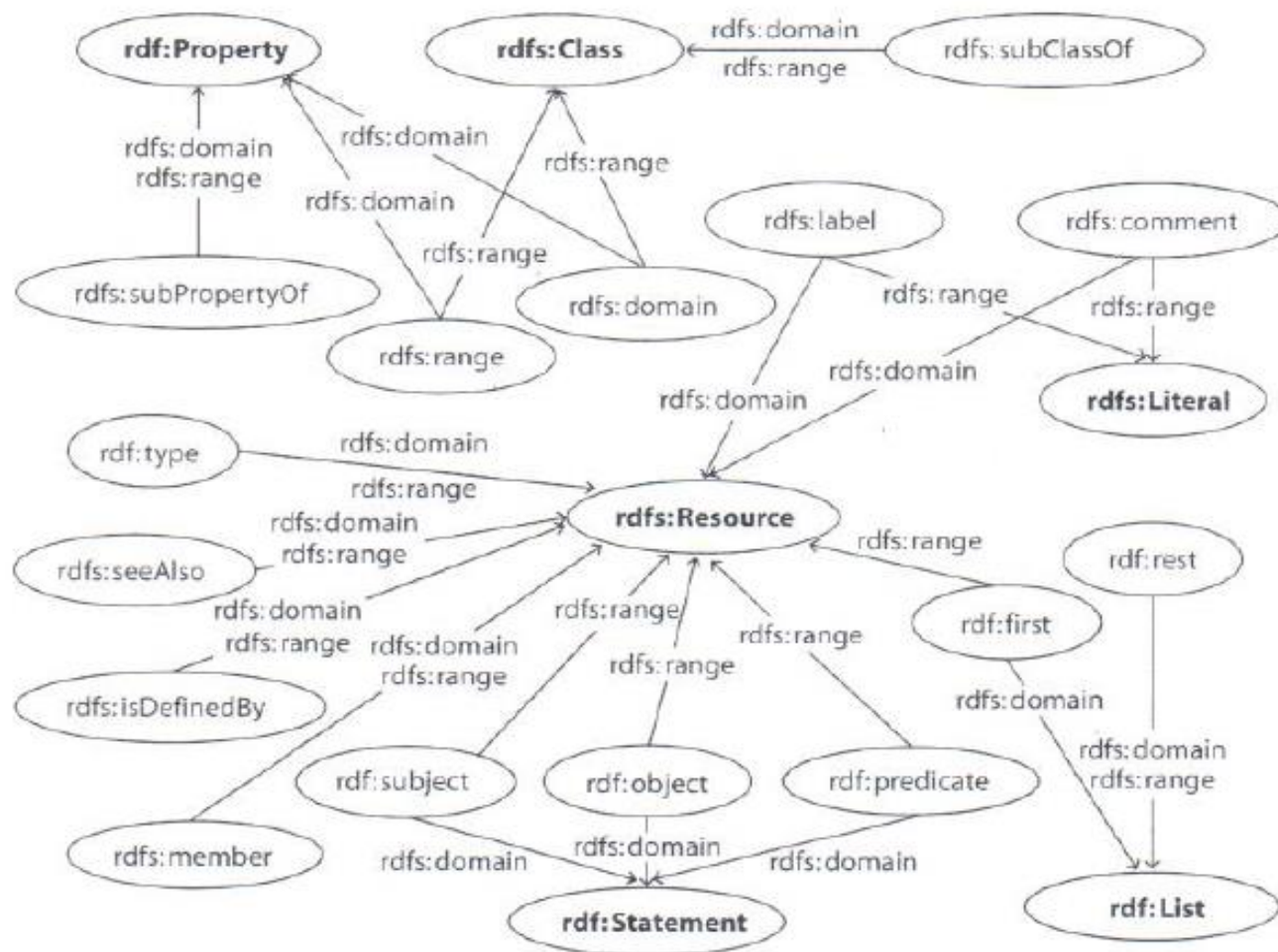
## RDF Schema - RDFS

### ➤ Méta-modèle RDFS : déclaration de propriétés



# RDF Schema - RDFS

## ➤ Méta-modèle RDFS



## RDF Schema - RDFS

### ➤ Exemples de schémas utilisant RDFS :

- Dublin Core (DC), qui décrit des ressources du web (e.g., images, vidéos) ou physiques. <http://dublincore.org/>
- Schema.org, un ensemble de schémas pour structurer les données du web et ses domaines populaires (e.g., personnes, restaurants, e-commerce, événements). <http://schema.org/>
- Friend of a Friend (FOAF), pour décrire des personnes, des activités et des relations. <http://www.foaf-project.org/>

## RDF Schema - RDFS

- RDFS permet de déduire des triplets additionnels à partir des triplets existants:
- Transitivité :

**SI**  $(c_2, \text{subClassOf}, c_1)$

**ET**  $(c_3, \text{subClassOf}, c_2)$

**ALORS**  $(c_3, \text{subClassOf}, c_1)$

**SI**  $(p_2, \text{subPropertyOf}, p_1)$

**ET**  $(p_3, \text{subPropertyOf}, p_2)$

**ALORS**  $(p_3, \text{subPropertyOf}, p_1)$

## RDF Schema - RDFS

- RDFS permet de déduire des triplets additionnels à partir des triplets existants:
- Propagation des types / propriétés:

**SI** (c<sub>2</sub>, subClassOf, c<sub>1</sub>)

**ET** (x, type, c<sub>2</sub>)

**ALORS** (x, type, c<sub>1</sub>)

**SI** (p<sub>2</sub>, subPropertyOf, p<sub>1</sub>)

**ET** (x, p<sub>2</sub>, y)

**ALORS** (x, p<sub>1</sub>, y)

## RDF Schema - RDFS

- RDFS permet de déduire des triplets additionnels à partir des triplets existants:
- Inférence – Range/Domain :

**SI**  $(p_1, \text{domain}, c_1)$   
**ET**  $(x, p_1, y)$   
**ALORS**  $(x, \text{type}, c_1)$

**SI**  $(p_1, \text{range}, c_1)$   
**ET**  $(x, p_1, y)$   
**ALORS**  $(y, \text{type}, c_1)$

# Web Sémantique - Outils

## W3C RDF Validation Service – <https://www.w3.org/RDF/Validator/>

### Check and Visualize your RDF documents

[olde servlet](#)

Enter a URI or paste an RDF/XML document into the text field above. A 3-tuple (triple) representation of the corresponding data model as

Check by Direct Input

```
<rdf:RDF
  xmlns:local="http://www.ox.ac.uk/Vocabulary#"
  xmlns:prof="http://www.ox.ac.uk/Profs#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" >

  <rdf:Description      rdf:about="http://www.ox.ac.uk/Profs#TimLee" >
    <rdf:type
  rdf:resource="http://www.ox.ac.uk/Vocabulary#Researcher"/>
  //
```

Parse RDF

Restore the original example

Clear the textarea

#### Display Result Options:

Triples and/or Graph: Triples and Graph ▾

Graph format: PNG - embedded ▾

Paste an RDF/XML document into the following text field to have it checked. More options are available in the [Extended interface](#).

Check by URI

Parse URI:

Clear the URI

#### Display Result Options:

Triples and/or Graph: Triples Only ▾

Graph format: PNG - embedded ▾

# Web Sémantique - Outils

## W3C RDF Validation Service – <https://www.w3.org/RDF/Validator/>

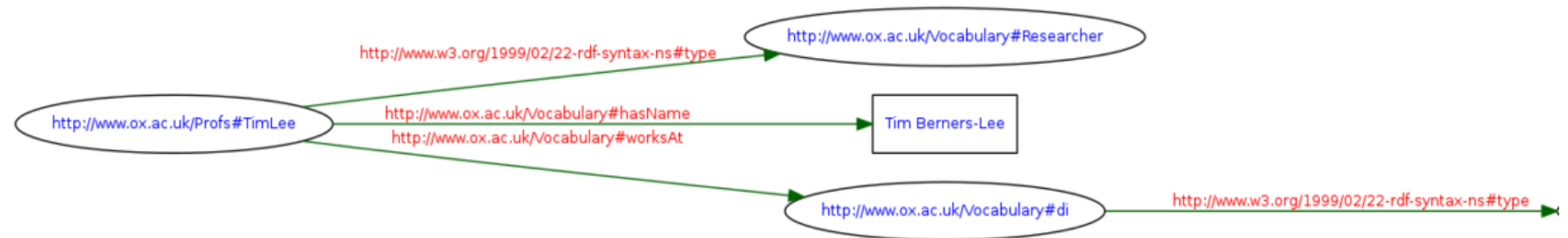
### Validation Results

Your RDF document validated successfully.

### Triples of the Data Model

Number	Subject	Predicate	Object
1	<a href="http://www.ox.ac.uk/Profs#TimLee">http://www.ox.ac.uk/Profs#TimLee</a>	<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#type">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</a>	<a href="http://www.ox.ac.uk/Vocabulary#Researcher">http://www.ox.ac.uk/Vocabulary#Researcher</a>
2	<a href="http://www.ox.ac.uk/Profs#TimLee">http://www.ox.ac.uk/Profs#TimLee</a>	<a href="http://www.ox.ac.uk/Vocabulary#hasName">http://www.ox.ac.uk/Vocabulary#hasName</a>	"Tim Berners-Lee"
3	<a href="http://www.ox.ac.uk/Profs#TimLee">http://www.ox.ac.uk/Profs#TimLee</a>	<a href="http://www.ox.ac.uk/Vocabulary#worksAt">http://www.ox.ac.uk/Vocabulary#worksAt</a>	<a href="http://www.ox.ac.uk/Vocabulary#di">http://www.ox.ac.uk/Vocabulary#di</a>
4	<a href="http://www.ox.ac.uk/Vocabulary#di">http://www.ox.ac.uk/Vocabulary#di</a>	<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#type">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</a>	<a href="http://www.ox.ac.uk/Vocabulary#AcademicDepartement">http://www.ox.ac.uk/Vocabulary#AcademicDepartement</a>

### Graph of the data model



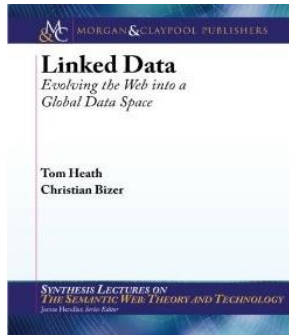
### Feedback

If you suspect the parser is in error, please enter an explanation below and then press the **Submit problem report** button, to mail the report (and listing) to [www-rdf-validator@w3.org](mailto:www-rdf-validator@w3.org)

Submit problem report

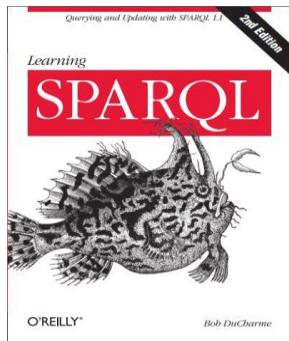


# Références



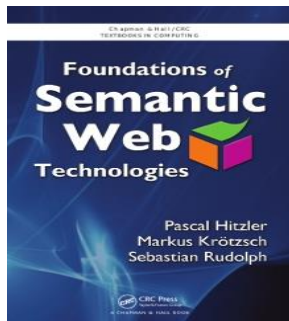
## **Linked Data: Evolving the Web into a Global Data Space**

- ✓ Auteur : Christian Bizer, Tom Heath
- ✓ Éditeur : Morgan & Claypool Publishers
- ✓ Edition : Février 2011 - 136 pages - ISBN 9781608454310



## **Learning SPARQL : Querying and Updating with SPARQL**

- ✓ Auteur : Bob DuCharme
- ✓ Éditeur : O'Reilly Media
- ✓ Edition: Juillet 2013– 386pages -ISBN : 9781449306595



## **Foundations of Semantic Web Technologies**

- ✓ Auteur : Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph
- ✓ Éditeur : CRC Press/Chapman and Hall
- ✓ Edition : 2009 - 455 pages - ISBN : 9781420090505

# Références

- W3C – Semantic Web
  - ✓ [https://www.w3.org/2001/sw/wiki/Main\\_Page](https://www.w3.org/2001/sw/wiki/Main_Page)
- INRIA MOOC - Fabien Gandon – Web Sémantique et Web de Données
  - ✓ [https://www.canal-u.tv/producteurs/inria/cours\\_en\\_ligne/web\\_semantique\\_et\\_web\\_de\\_donnees](https://www.canal-u.tv/producteurs/inria/cours_en_ligne/web_semantique_et_web_de_donnees)
- Cours en Ligne- Michel Gagnon – Ontologies et Web Sémantique
  - ✓ <https://moodle.polymtl.ca/course/view.php?id=30>
- Initiation à RDF - W3C
  - ✓ <http://www.yoyodesign.org/doc/w3c/rdf-primer>
- Fabien Duchateau – BDBIO – RDF et SPARQL
  - ✓ <http://liris.cnrs.fr/~fduchate/ens/BDBIO/cm/rdf.pdf>