



Introduction au Web sémantique et au Linked Data

Jean-Baptiste Pressac

► To cite this version:

Jean-Baptiste Pressac. Introduction au Web sémantique et au Linked Data. Licence. Action nationale de formation (ANF) “ Initiation au SPARQL et à la diffusion de données en RDF ”, En distanciel, France. 2020, pp.30. hal-04267466

HAL Id: hal-04267466

<https://hal.science/hal-04267466>

Submitted on 1 Nov 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.


L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.





Distributed under a Creative Commons Attribution 4.0 International License

Introduction au Web sémantique et au Linked Data

ANF « Initiation au SPARQL et à la diffusion de données en RDF » 7-8 décembre 2020
Jean-Baptiste Pressac, Centre de recherche bretonne et celtique (UMS 3554)

Introduction	2
L'enjeu du Web sémantique : Séparer contenu et contenant	2
RDF et SPARQL	5
Triplets RDF, l'exemple de data.idref.fr	5
Exercices 	9
Exercice 1	9
Exercice 2	10
Réponses 	10
Réponse 1	10
Réponse 2	11
URI et littéraux	12
Ressources RDF identifiées par des URI	12
Les littéraux	14
Type des littéraux	15
Langue des littéraux	16
Exercices 	17
Exercice 1	17
Exercice 2	18
Exercice 3	18
Exercice 4	18
Réponses 	18
Réponse 1	18
Réponse 2	19
Réponse 3	20
Réponse 4	20
Et le sémantique dans tout ça ?	21
Linked data, Linked Open Data	21
Déduire des triplets : les inférences	22
Réutiliser des ontologies existantes ou créer sa propre ontologie	24
Au départ, il y avait RDF, RDFS et OWL	25

Exercices 	26
Exercice 1	26
Exercice 2	26
Réponses 	26
Réponse 1	26
Réponse 2	27
Où sont les triplets ? SPARQL endpoint et bases de données de graphes	27
Quelques exemples de SPARQL endpoint	28
Les alternatives au SPARQL et au Web sémantique	29
Pour en savoir plus	30

Introduction

Recommandation du W3C¹ depuis le 15 janvier 2008, le langage de requête **SPARQL** est une des briques du **Web sémantique**. Mais qu'est ce que le Web sémantique ? Qui en est à l'origine ? A quelles problématiques tente-t-il de répondre ? Comment s'intègre-t-il dans le Web des données (Linked Data), le Web ouvert des données (Open Linked Data) ?

L'enjeu du Web sémantique : Séparer contenu et contenant

Actuellement, le web repose sur des technologies qui emprisonnent les données, les encapsulent dans du code HTML. Si vous souhaitez récupérer le texte et les images d'une page web, vous pouvez utiliser des flux RSS, des API², à la condition que les données s'y prêtent et que l'éditeur du site ait prévu de mettre ces services à votre disposition. Si vos compétences en programmation sont suffisantes, vous pouvez aspirer le contenu d'un site Web avec des logiciels de web scraping. Mais le web actuel n'est pas nativement conçu pour dissocier contenu (le texte et les images) et contenant (le code HTML).

Le contenu textuel des sites internet est la plupart du temps stocké dans des bases de données, notamment des bases de données relationnelles reposant sur le langage SQL. Mais le SQL n'est pas un protocole et il n'est pas possible d'interroger ces bases à travers HTTP (avec les méthodes GET, PUT, POST). Qui plus est, les bases de données des CMS (Wordpress, Drupal, SPIP, etc.) et des framework (Django, Symfony, Laravel, etc.) servent aussi bien à stocker les données publiques (le texte d'un article de blog, par exemple) que les données servant à leur gestion (les comptes Wordpress des rédacteurs de l'article). Donner libre accès à ces bases supposerait donc une gestion contraignante des droits

¹ Le W3C ou world wide web consortium est un organisme de standardisation à but non lucratif, fondé en octobre 1994 chargé de promouvoir la compatibilité des technologies du World Wide Web telles que HTML5, HTML, XHTML, XML, RDF, SPARQL, CSS, XSL, PNG, SVG et SOAP (source : Wikipédia)

² Application Programmable Interface

d'accès aux seules tables de données publiques.

Pourtant, Tim Berners-Lee avait envisagé dès 1994³ que le Web devait être plus qu'un espace documentaire avec des liens entre documents⁴. Chaque notion du monde réel (ou plutôt de l'*univers du discours*) devait avoir son alter-ego sur le Web, identifié par son **URI**⁵ (Uniform Resource Identifier). Ces **ressources** (le R de URI) liées entre elles par des propriétés, constituant un gigantesque graphe. Dans ce graphe, l'URI de la ressource correspondant à Victor Hugo, par exemple, aurait été reliée à l'URI de la ville de Besançon par la propriété "ville de naissance" (une autre ressource identifiée par une URI).

Dans le modèle du Web sémantique, la déclaration (*Victor Hugo a pour ville de naissance Besançon*) constitue un **triplet** où la ressource *Victor Hugo* est le **sujet**, lié par le **prédicat** *a pour ville de naissance* (aussi appelé une propriété) à l'**objet** *Besançon*.

Victor Hugo	<i>a pour ville de naissance</i>	Besançon
Sujet	Prédicat	Objet

Victor Hugo, *a pour ville de naissance* et *Besançon* sont tous les trois des ressources qui doivent être identifiées par des URI. Si ce triplet était déclaré sur le site de votre laboratoire, il pourrait s'écrire :

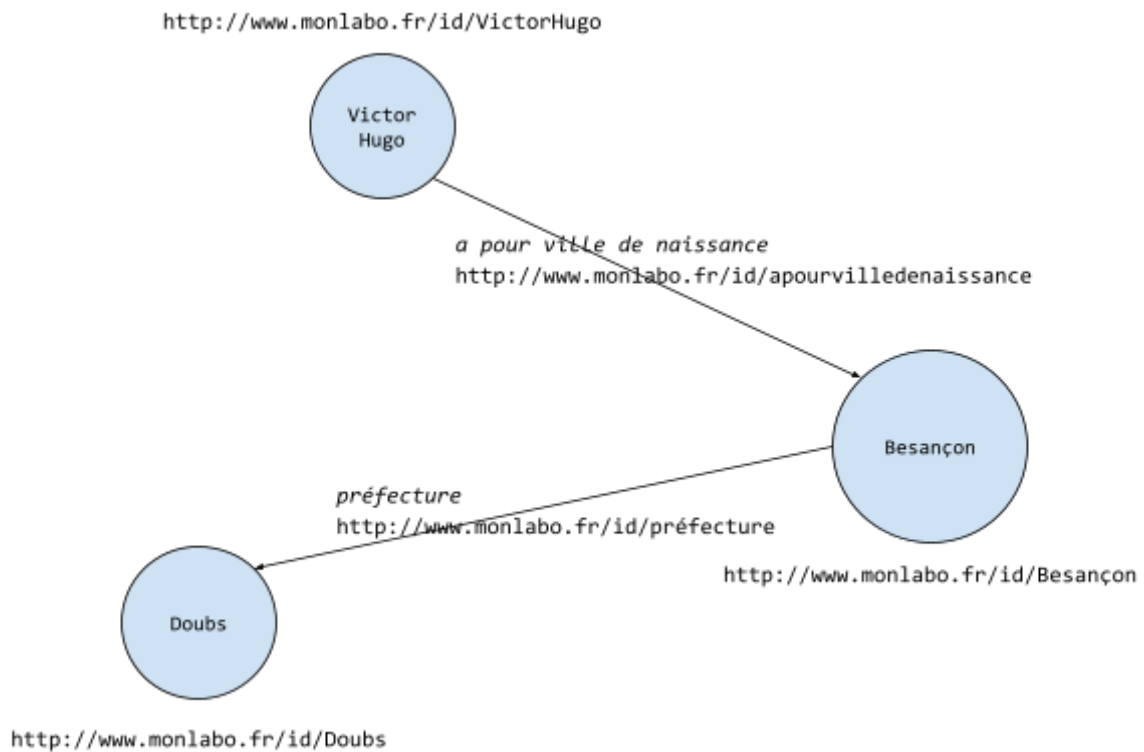
```
http://www.monlabo.fr/id/VictorHugo
http://www.monlabo.fr/id/apourvilledenaissance
http://www.monlabo.fr/id/Besançon
```

La ressource Besançon peut elle-même faire l'objet de triplets pour exprimer le fait que Besançon est la préfecture du Doubs. Le Doubs étant lui-même une ressource, décrite par d'autres triplets, on obtient rapidement un graphe où les sujets des triplets sont liés aux objets par les prédicats.

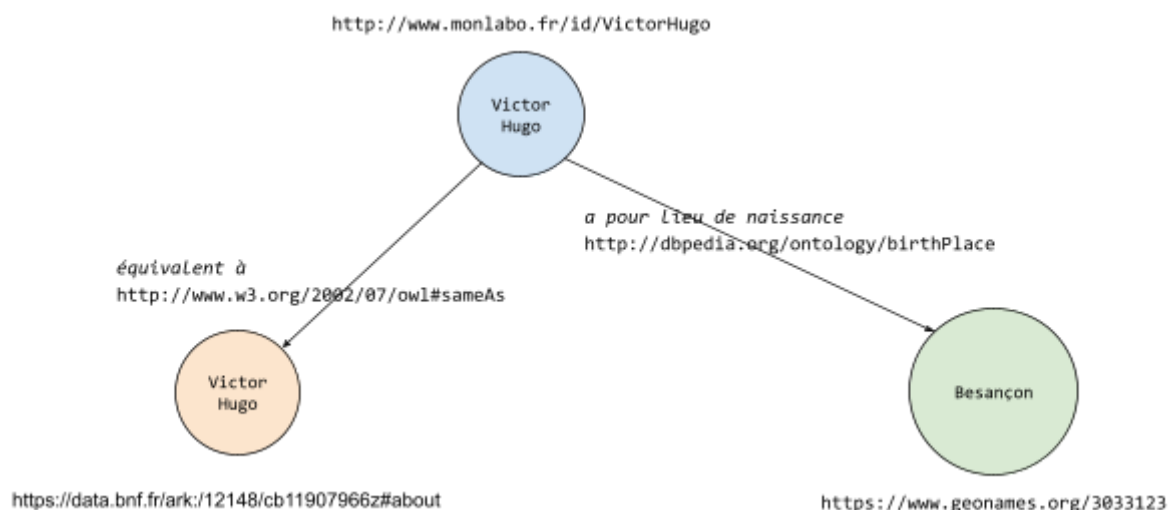
³ <https://www.w3.org/Talks/WWW94Tim/>

⁴ Le Web sémantique, Gandon, Faron-Zucker, Corby, 2012 p.8

⁵ Les URI (Uniform Resource Identifier) identifient des ressources. Pour en savoir plus, consulter [Cool URIs for the Semantic Web](#) et [URIs, URLs, and URNs: Clarifications and Recommendations 1.0](#)



Ce graphe peut être lié à d'autres graphes, par exemple en déclarant que Victor Hugo est équivalent à la ressource <https://data.bnf.fr/ark:/12148/cb11907966z#about> sur data.bnf.fr⁶, ou que Besançon est équivalent à la ressource <https://www.wikidata.org/entity/Q37776> sur la base de connaissances collaborative Wikidata.



⁶ data.bnf.fr est une base de données contenant une version sémantique des données sur les œuvres, les auteurs et les thèmes du catalogue de la Bibliothèque nationale de France (source : Wikidata).

C'est dans le but de créer ce gigantesque graphe de données, qu'a été créé en 2004 par le W3C, le **RDF**⁷ (Resource Description Framework) et en 2008 le langage de requête et protocole **SPARQL** (SPARQL Protocol and RDF Query Language).

💡 A noter : *a pour ville de naissance* et *préfecture* étant également des ressources, elles peuvent aussi être sujets de triplets exprimant par exemple le fait que *a pour ville de naissance* s'applique à des êtres humains ou que *préfecture* est une division administrative territoriale.

RDF et SPARQL

Selon la note du groupe de travail du W3C [RDF 1.1 Primer](#), le Resource Description Framework (RDF) est un framework, un ensemble de recommandations pour exprimer des informations à propos de ressources. Les **ressources**⁸ pouvant être n'importe quoi, y compris des documents, des individus, des objets physiques, des concepts abstraits, des chaînes de caractères, des chiffres.

Le RDF permet d'exprimer des déclarations à propos des ressources. Ces déclarations sont exprimées sous la forme de **triplets** qui peuvent être interrogés, créés et modifiés grâce au **SPARQL**.

Selon Wikipédia, SPARQL (prononcé sparkle, en anglais : « étincelle ») est un **langage de requête** et un [protocole](#) qui permet de rechercher, d'ajouter, de modifier ou de supprimer des données RDF **disponibles à travers Internet**. Son nom est un acronyme récursif qui signifie *SPARQL Protocol and RDF Query Language*.

💡 A noter : Le F de RDF ne signifie pas Format mais Framework. Le RDF n'est pas un *format* de données mais un *modèle* de données.

Triplets RDF, l'exemple de data.idref.fr

A quoi ressemblent des triplets RDF ? Pour y voir plus clair, partons de l'exemple de IdRef, le référentiel d'autorité de l'Abes, Agence bibliographique de l'enseignement supérieur.

En [science de l'information](#), une **autorité** (ou **notice d'autorité** ou **forme d'autorité**) sert à identifier sans ambiguïté des personnes ou d'autres concepts et à faciliter la recherche dans les catalogues de bibliothèques (source : [Wikidata](#)). Autrement dit, quand les documentalistes de votre bibliothèque universitaire favorite saisissent un nouvel ouvrage dans leur catalogue, se pose la question de la dénomination de l'auteur et de la distinction des homonymes. Afin que toutes les bibliothèques universitaires de France adoptent la même dénomination et puissent identifier de manière univoque des individus, l'Agence

⁷ Pour une des meilleures introduction au RDF, consulter [RDF 1.1 Primer](#), la note du groupe de travail sur le RDF au W3C.

⁸ Pour une définition du concept de ressource, voir la recommandation du W3C [RDF 1.1 Concepts and Abstract Syntax](#).

bibliographique pour l'enseignement supérieur (Abes) à développé IdRef⁹, accessible sur internet à l'adresse <https://www.idref.fr/>.

IdRef a la particularité de diffuser des données selon les principes du Web sémantique et du **Linked Data**. Par exemple, La notice du folkloriste breton Théodore Hersart de la Villemarqué est une page Web classique <https://www.idref.fr/026956845> qui peut être consultée avec un navigateur web. Mais les données qu'elle contient sont aussi accessibles sous forme de triplets RDF en rajoutant le suffixe .rdf à l'URL : <https://www.idref.fr/026956845.rdf>.

Les triplets sont exprimés en [RDF/XML](#), une des syntaxes du RDF¹⁰. Le RDF/XML fera le bonheur d'un logiciel mais pour nous, humains, la syntaxe Turtle sera plus lisible. Pour exprimer les triplets en Turtle, utilisons le convertisseur [EasyRDF Converter](#) en précisant l'URL <https://www.idref.fr/026956845.rdf> dans le champ "or Uri" du formulaire :

The screenshot shows the 'EasyRDF Converter' web application. The header includes the logo and navigation links: Documentation, Examples, Converter, Support, Downloads. The main section is titled 'Converter' and contains a form with the following fields:

- Input Data:** A large text area for pasting data.
- or Uri:** A text input field containing the URL `http://www.idref.fr/026956845`.
- Input Format:** A dropdown menu set to 'Guess'.
- Output Format:** A dropdown menu set to 'Turtle Terse RDF Triple Language'.
- Raw output:** An unchecked checkbox.
- Buttons:** 'Clear' and 'Submit' buttons.

Below the form, the 'Output' section shows a blue bar indicating 'Number of triples parsed: 294'. Below this, a code block displays the generated Turtle output:

```
@prefix bibo: <http://purl.org/ontology/bibo/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix dc: <http://purl.org/dc/terms/> .
```

⁹ Identifiants et Référentiels pour l'enseignement supérieur et la recherche

¹⁰ Le RDF est un ensemble de recommandations, de règles générales pour exprimer des informations à propos de ressources mais pour écrire (on dit aussi sérialiser) des triplets RDF, on utilise des [syntaxes RDF](#) : [RDFa](#), [JSON-LD](#), [RDF/XML](#), [Turtle](#)...

La liste des triplets est précédée par la déclaration de préfixes.

```
@prefix bibo: <http://purl.org/ontology/bibo/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix dc: <http://purl.org/dc/terms/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix ns0: <http://data.bnf.fr/ontology/bnf-onto/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix ns1: <http://isni.org/ontology#> .
@prefix ns2: <http://purl.org/vocab/bio/0.1/> .
@prefix ns3: <http://dbpedia.org/ontology/> .
@prefix ns4: <http://rdaregistry.info/Elements/u/> .
@prefix ns5: <http://id.loc.gov/vocabulary/relators/> .
```

Les préfixes permettent d'alléger l'écriture des URL et des URI dans les triplets.

Par exemple, le premier triplet de <https://www.idref.fr/026956845.rdf> peut s'écrire comme suit :

```
<http://www.idref.fr/026956845> a bibo:Document .
```

Sans la déclaration du préfixe bibo, le triplet aurait dû s'écrire :

```
<http://www.idref.fr/026956845> a <http://purl.org/ontology/bibo/Document> .
```

💡 A noter : Si vous publiez des triplets RDF, vous êtes libres d'utiliser les préfixes que vous voulez. Vous pouvez très bien par exemple utiliser bibonto: au lieu de bibo: comme préfixe de BIBO mais pour une bonne compréhension de vos triplets, vous pouvez consulter le site prefix.cc pour connaître les préfixes les plus couramment utilisés. Par exemple, une recherche sur <http://purl.org/ontology/bibo/> indique bien que le préfixe généralement associé à cette URI est bibo.

💡 A noter : L'Abes aurait aussi pu déclarer un préfixe pour ses propres URI :

```
@prefix idref: <http://www.idref.fr/> .
@prefix bibo: <http://purl.org/ontology/bibo/> .
```

```
idref:026956845 a bibo:Document .
```

La première série de triplets a pour **sujet** (au sens de sujet / prédicat / objet) la ressource <http://www.idref.fr/026956845>, qui correspond à l'adresse Web de la notice de Théodore Hersart de la Villemarqué.

```
# La notice IdRef de Théodore Hersart de la Villemarqué est le sujet de ces triplets
<http://www.idref.fr/026956845>
  a bibo:Document ;
  foaf:primaryTopic <http://www.idref.fr/026956845/id> ;
```



```

dc:identifier "026956845" ;
dc:creator <http://www.idref.fr/033702462/id> ;
dc:created "1978-02-08"^^xsd:date ;
dc:modified "2020-04-15T15:35:24"^^xsd:dateTime ;
skos:editorialNote "Les Romans de la table ronde / par H. de la Villemarqué, 1989-.
Barzhaz breizh / [éd. par] Kervarker, 1988", "Larousse 19 s.. GDEL. 500 bretons à
connaître / Jean-Loup Avril, 1989", "http://catalogue.bnf.fr/servlet/autorite, 2007" .

```

💡 A noter : En Turtle, les URI et les noms de fichiers sont écrits entre crochets.

Cette URI (Uniform Resource Identifier) est **déréférencable**¹¹. C'est à dire qu'elle ne sert pas uniquement à identifier la ressource mais qu'il est possible d'obtenir des triplets RDF ou des informations à son sujet en accédant à son URI avec un navigateur Web ou une interface en ligne de commande telle que [cURL](#). Mais notez bien qu'une URI est en premier lieu un identifiant. Le W3C recommande de fournir des triplets RDF ou du HTML en tentant d'accéder à une URI travers le Web, mais ce n'est pas obligatoire.

En Turtle, les triplets se terminent par un point (surligné en jaune). Dans le cas où plusieurs prédicats (en bleu) et objets (en noir) partagent le même sujet (en vert), il est possible d'alléger l'écriture en utilisant le point-virgule.

Sans l'utilisation des points-virgules, les sept triplets précédents auraient été écrit de la façon suivante, en répétant à chaque fois le même sujet :

```

# La notice IdRef de Théodore Hersart de la Villemarqué est le sujet de ces triplets
<http://www.idref.fr/026956845> a bibo:Document .
<http://www.idref.fr/026956845> foaf:primaryTopic <http://www.idref.fr/026956845/id> .
<http://www.idref.fr/026956845> dc:identifier "026956845" .
<http://www.idref.fr/026956845> dc:creator <http://www.idref.fr/033702462/id> .
<http://www.idref.fr/026956845> dc:created "1978-02-08"^^xsd:date .
<http://www.idref.fr/026956845> dc:modified "2020-04-15T15:35:24"^^xsd:dateTime .
<http://www.idref.fr/026956845> skos:editorialNote "Les Romans de la table ronde / par H.
de la Villemarqué, 1989-. Barzhaz breizh / [éd. par] Kervarker, 1988", "Larousse 19 s..
GDEL. 500 bretons à connaître / Jean-Loup Avril, 1989",
"http://catalogue.bnf.fr/servlet/autorite, 2007" .

```

De même, si des triplets partagent le même sujet et le même prédicat, les objets peuvent être écrites à la suite du prédicat en utilisant des virgules. Par exemple, les triplets liant la ressource représentant Théodore Hersart de la Villemarqué sur IdRef avec les ressources équivalentes sur Wikidata, ISNI, data.bnf.fr et VIAF :

```

# Le sujet de ces triplets est Théodore Hersart, l'individu
<http://www.idref.fr/026956845/id> owl:sameAs <http://www.wikidata.org/entity/Q1391034> .
<http://www.idref.fr/026956845/id> owl:sameAs <http://isni.org/isni/0000000110263523> .
<http://www.idref.fr/026956845/id> owl:sameAs

```

¹¹ [Qu'est-ce qu'une "URI déréférencable" ? \(Christian Fauré\)](#)

```
<http://data.bnf.fr/ark:/12148/cb119103377#foaf:Person> .
<http://www.idref.fr/026956845/id> owl:sameAs <http://viaf.org/viaf/49227556> .
```

... peuvent être écrits en abrégé avec des virgules (notez le point final) :

```
<http://www.idref.fr/026956845/id> owl:sameAs <http://www.wikidata.org/entity/Q1391034>,
<http://isni.org/isni/0000000110263523>,
<http://data.bnf.fr/ark:/12148/cb119103377#foaf:Person>, <http://viaf.org/viaf/49227556> .
```

Le tableau suivant tiré de [Wikidata:SPARQL tutorial](#) résume les **abréviations** en usage pour la syntaxe RDF Turtle (sujet en rouge, prédicat en bleu, objet en vert) :

natural language	example	SPARQL	example
sentence	Juliet loves Romeo.	period	juliet loves romeo.
conjunction (clause)	Romeo loves Juliet and kills himself.	semicolon	romeo loves juliet; kills romeo.
conjunction (noun)	Romeo kills Tybalt and himself.	comma	romeo kills tybalt, romeo.
relative clause	Juliet loves someone who kills Tybalt.	brackets	juliet loves [kills tybalt].

Exercices

Exercice 1

Dans quelle syntaxe les triplets suivants ont-ils été écrits ? Citez une syntaxe RDF alternative. Comment auriez-vous écrit, sans les préfixes, les triplets dont le prédicat est foaf:Person et owl:sameAs ? Réécrivez les triplets de manière abrégée.

```
@prefix bibo: <http://purl.org/ontology/bibo/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix dc: <http://purl.org/dc/terms/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix ns0: <http://data.bnf.fr/ontology/bnf-onto/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix ns1: <http://isni.org/ontology#> .
@prefix ns2: <http://purl.org/vocab/bio/0.1/> .
@prefix ns3: <http://dbpedia.org/ontology/> .
@prefix ns4: <http://rdaregistry.info/Elements/u/> .
@prefix ns5: <http://id.loc.gov/vocabulary/relators/> .

<http://www.idref.fr/026956845/id> a foaf:Person .
<http://www.idref.fr/026956845/id> owl:sameAs <http://www.wikidata.org/entity/Q1391034> .
<http://www.idref.fr/026956845/id> owl:sameAs <http://isni.org/isni/0000000110263523> .
<http://www.idref.fr/026956845/id> owl:sameAs
<http://data.bnf.fr/ark:/12148/cb119103377#foaf:Person> .
<http://www.idref.fr/026956845/id> owl:sameAs <http://viaf.org/viaf/49227556> .
<http://www.idref.fr/026956845/id> skos:prefLabel "La Villemarqué, Théodore Hersart
```

```
(1815-1895 ; vicomte de)" .  
<http://www.idref.fr/026956845/id> ns2:event <http://www.idref.fr/026956845/birth> .  
<http://www.idref.fr/026956845/id> ns2:event <http://www.idref.fr/026956845/death> .
```

Exercice 2

Dans les triplets suivants, désignez le prédicat, le sujet et l'objet :

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .  
@prefix ns2: <http://purl.org/vocab/bio/0.1/> .  
  
<http://www.idref.fr/026956845/id> a foaf:Person ;  
    skos:prefLabel "La Villemarqué, Théodore Hersart (1815-1895 ; vicomte de)" ;  
    ns2:event <http://www.idref.fr/026956845/birth>,  
    <http://www.idref.fr/026956845/death> .
```

Réponses

Réponse 1

Les triplets RDF ont été écrits avec la syntaxe Turtle. Le RDF/XML est une autre syntaxe possible pour écrire des triplets RDF.

Sans le préfixe **foaf**: les triplets suivants :

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
@prefix owl: <http://www.w3.org/2002/07/owl#> .  
  
<http://www.idref.fr/026956845/id> a foaf:Person .  
<http://www.idref.fr/026956845/id> owl:sameAs <http://viaf.org/viaf/49227556> .
```

... auraient été écrits :

```
<http://www.idref.fr/026956845/id> a <http://xmlns.com/foaf/0.1/Person> .  
<http://www.idref.fr/026956845/id> <http://www.w3.org/2002/07/owl#  
sameAs> <http://viaf.org/viaf/49227556> .
```

Notez l'utilisation des crochets entourant les URI, la suppression du deux-points avant **:Person** et **:sameAs** et la présence du **#** dans l'URI de la propriété **sameAs**.

Voici la version abrégée des triplets :

```
<http://www.idref.fr/026956845/id>  
a foaf:Person ;  
owl:sameAs <http://www.wikidata.org/entity/Q1391034> ,
```

```

<http://isni.org/isni/0000000110263523>,
<http://data.bnf.fr/ark:/12148/cb119103377#foaf:Person>,
<http://viaf.org/viaf/49227556> ;
skos:prefLabel "La Villemarqué, Théodore Hersart (1815-1895 ; vicomte de)" ;
ns2:event <http://www.idref.fr/026956845/birth>, <http://www.idref.fr/026956845/death> .

```

Réponse 2

```

@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix ns2: <http://purl.org/vocab/bio/0.1/> .

<http://www.idref.fr/026956845/id> a foaf:Person ;
    skos:prefLabel "La Villemarqué, Théodore Hersart (1815-1895 ; vicomte de)" ;
    ns2:event <http://www.idref.fr/026956845/birth>,
    <http://www.idref.fr/026956845/death> .

```

Tous les triplets RDF ont la forme suivante :

sujet prédicat objet .

Les triplets RDF écrits avec la syntaxe Turtle peuvent être abrégés de la manière suivante :

```

# Version non abrégée de triplets écrits dans la syntaxe Turtle
sujet prédicat-A objet-A1 .
sujet prédicat-A objet-A2 .
sujet prédicat-B objet-B1 .
sujet prédicat-B objet-B2 .

# Les mêmes triplets écrits de manière abrégée en Turtle
sujet prédicat-A objet-A1 , objet-A2 ;
    prédicat-B objet-B1 , objet-B2 .

```

D'où les réponses à l'exercice :

sujet	prédicat	objet
<http://www.idref.fr/026956845/id>	a	foaf:Person
<http://www.idref.fr/026956845/id>	skos:prefLabel	"La Villemarqué, Théodore Hersart (1815-1895 ; vicomte de)"
<http://www.idref.fr/026956845/id>	ns2:event	<http://www.idref.fr/026956845/birth>

<http://www.idref.fr/026956845/id>	ns2:event	<http://www.idref.fr/026956845/death>
------------------------------------	-----------	---------------------------------------

URI et littéraux

Les triplets RDF expriment la relation entre deux ressources, le **sujet** et l'**objet**, à l'aide d'une troisième ressource, le **prédicat**. Les ressources RDF peuvent être identifiées par des URI ou des littéraux.

Ressources RDF identifiées par des URI

Les ressources identifiées par des URI appartiennent en général à une **classe**, définie par une **ontologie**. Les ontologies définissent un ensemble de classes, sous-classes et **propriétés** permettant d'exprimer des connaissances dans un domaine (connaissances sur des individus et les relations qu'ils entretiennent entre eux, connaissances bibliographiques sur des ouvrages, connaissances géographiques sur des lieux, etc.).

Les ontologies sont exprimées en RDF, ce qui permet de les interroger en SPARQL.

La notice de Théodore Hersart de la Villemarqué, c'est à dire la ressource identifiée par l'URI <http://www.idref.fr/026956845>, est un *document* selon l'ontologie BIBO, acronyme de [BIBliographic Ontology](#). Cette déclaration s'exprime de la manière suivante en Turtle :

```
# Notice IdRef de Théodore Hersart de la Villemarqué
<http://www.idref.fr/026956845> a bibo:Document .
```

Le symbole¹² **a** de la syntaxe RDF Turtle est équivalent à la propriété **rdf:type** qui permet de déclarer l'appartenance d'une ressource à une **classe** :

```
<http://www.idref.fr/026956845> rdf:type bibo:Document .
```

La classe `bibo:Document` est définie dans la version RDF de l'ontologie BIBO, dont l'URI est indiquée par le préfixe `bibo:` précédant les triplets :

```
@prefix bibo: <http://purl.org/ontology/bibo/> .
```

Pour en savoir plus sur la classe `bibo:Document`, on soumet à [EasyRDF Converter](#) l'URL de BIBO (en précisant Input Format: RDF/XML dans l'interface de EasyRDF). Un triplet déclare `bibo:Document` comme appartenant à la classe `owl:Class` :

```
bibo:Document a owl:Class ;
  rdfs:label "Document"@en ;
```

¹² **a** est un token en anglais, d'après la spécification du W3C [RDF 1.1 Turtle](#)

```
    rdfs:comment "A document (noun) is a bounded physical representation of body of
information designed with the capacity (and usually intent) to communicate. A document may
manifest symbolic, diagrammatic or sensory-representational information."@en .
```

Les propriétés `rdfs:label` et `rdfs:comment` permettent d'en savoir plus ce qui est considéré comme un document par BIBO (si vous comprenez l'anglais).

BIBO définit également des sous-classes de la classe `bibo:Document`, notamment les classes `bibo:Image`, `bibo:Manuscript`, `bibo:SlideShow`. Et les propriétés, qui peuvent être utilisées comme prédicats, par exemple, `bibo:authorList` qui permet de déclarer les auteurs d'un document.

```
bibo:authorList
  a owl:ObjectProperty ;
  rdfs:label "list of authors"@en ;
  rdfs:isDefinedBy "http://purl.org/ontology/bibo/"^^xsd:anyURI ;
  ns0:term_status "stable" ;
  rdfs:comment "An ordered list of authors. Normally, this list is seen as a priority list
that order authors by importance."@en ;
  rdfs:domain bibo:Document ;
  rdfs:subPropertyOf bibo:contributorList ;
  rdfs:range [
    a owl:Class ;
    owl:unionOf (
      rdf:List
      rdf:Seq
    )
  ] .
```

Aussi, nous pourrions déclarer que l'Agence bibliographique de l'enseignement supérieur, dont l'URI est `<http://www.idref.fr/033702462/id>` (notez bien que l'URI de cette ressource se termine par un `/id`, il s'agit donc bien de l'organisme et non pas de sa notice IdRef) est l'auteur de la notice IdRef de Théodore Hersart de la Villemarqué :

```
# La notice de Théodore (026956845) a pour auteur l'Agence bibliographique de
l'enseignement supérieur (033702462/id)
<http://www.idref.fr/026956845> bibo:authorList <http://www.idref.fr/033702462/id> .
```

L'Abes a néanmoins choisi d'utiliser l'élément `dc:creator` du Dublin Core :

```
# La notice de Théodore (026956845) a pour créateur l'Agence bibliographique de
l'enseignement supérieur (033702462/id)
<http://www.idref.fr/026956845> dc:creator <http://www.idref.fr/033702462/id> .
```

Bien entendu, il n'existe pas une seule ressource sur le Web qui représente l'Abes ou Théodore Hersart de la Villemarqué. Tout le monde peut créer une ressource RDF censée

les représenter. Malheureusement, il n'existe pas de moteur de recherche de ressources RDF. Le mieux que l'on puisse espérer, c'est que ces ressources aient été liées avec d'autres référentiels avec la propriété `owl:sameAs`. Par exemple, la ressource `IdRef` qui représente Théodore a été liée avec des ressources équivalentes sur `data.bnf.fr` (la base de données sémantique de la Bibliothèque nationale de France), Wikidata (la base de connaissance libre de la Wikimedia foundation, éditeur de Wikipédia), ISNI (International Standard Name Identifier ou Code international normalisé des noms) et VIAF (Virtual International Authority File, un référentiel d'autorité international) :

```
<http://www.idref.fr/026956845/id> owl:sameAs <http://www.wikidata.org/entity/Q1391034>,  
<http://isni.org/isni/0000000110263523>,  
<http://data.bnf.fr/ark:/12148/cb119103377#foaf:Person>, <http://viaf.org/viaf/49227556> .
```

Une **classe RDF** est donc une ressource RDF de type `owl:Class` déclarée dans une ontologie utilisant l'ontologie OWL¹³ (oui, le serpent se mord la queue). Si on voulait pousser encore plus loin nos investigations, on pourrait consulter la version RDF de l'ontologie OWL¹⁴ (<http://www.w3.org/2002/07/owl#>) où l'on trouve le triplet définissant `owl:Class` :

```
owl:Class a rdfs:Class .
```

Et pour finir, dans l'ontologie RDFS (<http://www.w3.org/2000/01/rdf-schema#>), le triplet définissant `rdfs:Class` de manière récursive :

```
rdfs:Class a rdfs:Class .
```

Les littéraux

L'Abes a suivi les recommandations de Tim Berners-Lee, co-créateur du Web et président du W3C dans son [design issue sur le Linked Data](#) (Web des données, en français) :

1. Use URIs as names for things (*Utiliser des URI pour nommer les choses*)
2. Use HTTP URIs so that people can look up those names. (*Utiliser des URI HTTP de manière à ce que les gens puissent y accéder*)
3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL) (Quand quelqu'un cherche à accéder à la ressource, lui fournir des informations utiles en utilisant les standards RDF et SPARQL.)
4. Include links to other URIs. so that they can discover more things. (créer des liens vers d'autres URI de manière à ce que d'autres informations puissent être trouvées)

¹³ OWL est un langage conçu pour représenter des connaissances riches et complexes à propos des objets, des groupes d'objets et des relations entre des objets (source : [OWL - Semantic Web Standards](#)) dont l'implémentation en RDF est l'ontologie du même nom.

¹⁴ Pour éditer le fichier téléchargé, ajoutez l'extension `.ttl` et ouvrez-le avec Notepad++ (sous Windows)

Aussi, on ne trouve pas dans la version RDF de la notice de Théodore, le triplet :

```
# La notice de Théodore (026956845) a pour créateur "Agence bibliographique de l'enseignement supérieur (France)"

<http://www.idref.fr/026956845> dc:creator "Agence bibliographique de l'enseignement supérieur (France)" .
```

Mais celui-ci :

```
# La notice de Théodore (026956845) a pour créateur l'Agence bibliographique de l'enseignement supérieur représentée par l'URI <http://www.idref.fr/033702462/id>

<http://www.idref.fr/026956845> dc:creator <http://www.idref.fr/033702462/id> .
```

Dans le vocabulaire du framework RDF, "Agence bibliographique de l'enseignement supérieur (France)" est un **littéral**, une ressource RDF sans URI, constituée d'une chaîne alphanumérique.

Utiliser une URI plutôt qu'un littéral permet de fournir des triplets complémentaires à propos de la ressource (sa classe, son label, sa date de naissance s'il s'agit d'un humain, etc.). Car un triplet RDF ne peut avoir un littéral comme sujet ou prédicat. On ne peut par exemple pas écrire le triplet :

```
"Agence bibliographique de l'enseignement supérieur (France)" a foaf:Organization .
```

Mais il faut bien écrire certaines informations en toutes lettres comme l'intitulé de l'Agence bibliographique de l'enseignement supérieur ou la date de publication d'une notice. Pour cela, il est recommandé de créer une nouvelle ressource représentant l'Abes puis d'utiliser un littéral :

```
# La notice de Théodore (026956845) a pour créateur l'Agence bibliographique de l'enseignement supérieur représentée par l'URI <http://www.idref.fr/033702462/id>

<http://www.idref.fr/026956845> dc:creator <http://www.idref.fr/033702462/id> .
<http://www.idref.fr/033702462/id> a foaf:Organization ;
    skos:prefLabel "Agence bibliographique de l'enseignement supérieur (France)" .
```

Type des littéraux

Le type et la langue d'un littéral peuvent être déclarés explicitement. Si on reprend l'exemple de la notice de Théodore, la date de création est précisée par le littéral de type date "1978-02-08"^^xsd:date :

```
<http://www.idref.fr/026956845> dc:created "1978-02-08"^^xsd:date .
```


L'Abes aurait tout aussi bien fournir le triplet sans préciser le type du littéral, dans ce cas, les guillemets auraient pu être omis. Mais préciser le type évite les quiproquos. Après tout, 1978-02-08 pourrait tout aussi bien désigner un numéro de commande, un numéro de client ou le titre d'un livre de science-fiction.

```
<http://www.idref.fr/026956845> dc:created 1978-02-08 . # Ce triplet est valide mais plus ambigu pour une machine
```

Le préfixe `xsd:` correspond à l'URI du XML Schema <http://www.w3.org/2001/XMLSchema#>, le type `xsd:date` est donc identifié par la ressource <http://www.w3.org/2001/XMLSchema#date>. Si vous tentez d'y accéder, vous arriverez sur une page web de présentation du Schéma XML. Pour la liste exhaustive des types de littéraux reconnus par défaut par RDF, consultez le chapitre [Datatypes](#) de la recommandation RDF 1.1 Concepts and Abstract Syntax, donc vous trouverez un extrait ci-dessous, illustré par des exemples :

- "Agence bibliographique de l'enseignement supérieur (France)"^{^^xsd:string}
- "false"^{^^xsd:boolean}
- "3.50"^{^^xsd:decimal}
- "4"^{^^xsd:integer}
- "2011-02-14"^{^^xsd:date}
- "2011"^{^^xsd:gYear}
- "2011"^{^^xsd:gYearMonth}

Notez l'usage systématique des guillemets dès que le type est précisé, même pour les dates, les valeurs numériques ou les booléens (vrai, faux).

Si aucun type ne correspond à vos données vous pouvez utiliser d'autres types que ceux proposés par le schéma XML.

Langue des littéraux

La langue d'un littéral peut être précisée en Turtle par une étiquette de langue (*langage tag*) : un `@` suivi du code ISO 639 de la langue¹⁵.

La langue des littéraux n'est pas précisée dans la notice de Théodore Hersart de la Villemarqué mais elle l'est par exemple dans [la notice](#) de la *Cellule nationale de coordination de l'indexation-matière* où l'étiquette de langue `efr` précise que le label de la ressource est en français :

```
<http://www.idref.fr/030081327/id>  
  a foaf:Organization ;
```

¹⁵ Les language tags sont définis dans le [RFC 3066](#) (request for comment) de l'IETF (Internet Engineering Task Force).

```
skos:label "France. Cellule nationale de coordination de l'indexation-matière"@fr .
```

Les langues sont à choisir dans le standard ISO 639-1, qui définit [des identifiants sur deux lettres](#) pour toutes les langues humaines connues, mortes ou vivantes. Elles peuvent être complétées par un code du standard ISO 3166-1 précisant le pays ou une de ses subdivision. Par exemple¹⁶ :

```
"camion"@fr-FR
"truck"@en-US
"lorry"@en-GB

"appartement"@fr-FR
"apartment"@en-US
"flat"@en-GB

"ascenseur"@fr-FR
"elevator"@en-US
"lift"@en-GB
```

Les étiquettes de langues ne sont pas cumulables avec la précision du type de données. Par exemple, on ne peut pas écrire :

```
"ascenseur"^^xsd:string@fr-FR
"elevator"^^xsd:string@en-US
"lift"^^xsd:string@en-GB
```

Le cours sur SPARQL donnera des exemples de requêtes pour filtrer des littéraux en fonction de la langue, pour récupérer l'année d'une date et pour changer le type d'un littéral.

Exercices

Exercice 1

Quelle est la classe de la ressource `<http://www.sudoc.fr/061950483/id>` ? Quelles sont les ontologies utilisées dans ces triplets ? Dans quelles syntaxes sont-elles exprimées ? Que signifie le prédicat `a` ?

```
@prefix bibo: <http://purl.org/ontology/bibo/> .
```

```
@prefix dc: <http://purl.org/dc/terms/> .
```

```
<http://www.sudoc.fr/061950483/id>
```

```
  a bibo:Document ;
```

```
  dc:subject <http://www.idref.fr/026956845/id> ;
```

```
  dc:bibliographicCitation "La Villemarqué : sa vie & ses œuvres / [Pierre Hersart de La
```

¹⁶ Cet exemple est tiré du livre de Bob Ducharme, Learning SPARQL (O'Reilly, 2013), p.170

Villemarqué] / Edition revue et augmentée / Paris : Librairie ancienne Honoré Champion, Édouard Champion , 1926" .

Exercice 2

Dans les triplets suivants, quelles sont les ressources ? Quels sont les littéraux ? Quelles précisions pourrait-on apporter aux littéraux ?

@prefix foaf: <http://xmlns.com/foaf/0.1/> .

@prefix skos: <http://www.w3.org/2004/02/skos/core#> .

@prefix dc: <http://purl.org/dc/terms/> .

<http://www.idref.fr/026956845/id> a foaf:Person .

<http://www.idref.fr/026956845/id> skos:prefLabel "La Villemarqué, Théodore Hersart (1815-1895 ; vicomte de)" .

<http://www.idref.fr/026956845> dc:created "1978-02-08" .

Exercice 3

Comment écrire en Turtle et l'[ontologie FOAF](#) que Yves Berthou est une personne, que son prénom est "Yves", que son nom de famille est "Berthou" et que Eskalibor et Kaledvourc'h sont ses pseudonymes bretons ? Utiliser la ressource RDF correspondant à Yves Berthou sur IdRef.

Exercice 4

A l'aide de [EasyRDF Converter](#), convertissez en Turtle les triplets RDF/XML de la notice de Théodore Hersart de la Villemarqué puis tentez de répondre aux questions suivantes :

1. Quel est le sujet principal de la notice de Théodore Hersart de la Villemarqué ?
2. Quelle est l'URI de la ressource RDF représentant Théodore Hersart de la Villemarqué ?
3. Quels sont les triplets correspondant à la date de naissance de Théodore ? Quelle ontologie a été utilisée pour exprimer cette information ?
4. Quel est le type de la ressource représentant la notice de Théodore Hersart de la Villemarqué ? Autrement dit, de quelle classe est-il l'instance ?
5. Quel est le type de la ressource représentant Théodore Hersart de la Villemarqué ? Autrement dit, de quelle classe est-il l'instance ?

Réponses

Réponse 1

@prefix bibo: <http://purl.org/ontology/bibo/> .

@prefix dc: <http://purl.org/dc/terms/> .

<http://www.sudoc.fr/061950483/id>

a bibo:Document ;

dc:subject <http://www.idref.fr/026956845/id> ;

dc:bibliographicCitation "La Villemarqué : sa vie & ses œuvres / [Pierre Hersart de La Villemarqué] / Edition revue et augmentée / Paris : Librairie ancienne Honoré Champion,

Édouard Champion , 1926" .

Comme indiqué par le triplet <<http://www.sudoc.fr/061950483/id>> a bibo:Document, (le prédicat a est équivalent en Turtle à rdf:type) cette ressource est une instance de la classe Document de l'ontologie BIBO. L'ontologie est décrite par des triplets en RDF/XML accessibles à l'URI déréférençable (lorsque l'on tente d'accéder à l'URI avec un navigateur, on obtient une réponse) <http://purl.org/ontology/bibo/>. Si on soumet à EasyRDF les triplets pour les convertir en Turtle pour plus de lisibilité, le nom complet de l'ontologie, The Bibliographic Ontology, et sa description, sont indiqués par les triplets ayant pour prédicats dc:title et dc:description :

```
bibo:
  a owl:Ontology ;
  owl:versionInfo "http://purl.org/ontology/bibo/1.3/" ;
  dc:title "The Bibliographic Ontology" ;
  dc:description ""The Bibliographic Ontology describes
bibliographic things on the semantic Web in RDF. This ontology can be
used as a citation ontology, as a document classification ontology, or
simply as a way to describe any kind of document in RDF. It has been
inspired by many existing document description metadata formats, and
can be used as a common ground for converting other bibliographic data
sources.""@en ;
  dc:creator bibo:bdarcus, bibo:fgiasson .
```

La deuxième ontologie utilisée est le Dublin Core, avec ses propriétés dc:subject et dc:bibliographicCitation. L'URI de cette ontologie est également déréférençable avec négociation de contenu¹⁷. C'est à dire que si vous saisissez l'adresse <http://purl.org/dc/terms/> dans un navigateur, vous accédez à une page HTML lisible par un humain alors que si utilisez un outil tel que EasyRDF, vous obtenez des triplets RDF exprimés avec la syntaxe Turtle. L'ontologie ne comprend pas de descriptif (pas de triplets avec le prédicat dc:description) mais son titre est précisé par le triplet suivant (DCMI = Dublin Core Metadata Initiative) :

```
dc: dc:title "DCMI Metadata Terms - other"@en .
```

Réponse 2

Dans les triplets, tous les sujets, prédicats et objets sont des ressources. Les littéraux sont "La Villemarqué, Théodore Hersart (1815-1895 ; vicomte de)" et la date "1978-02-08". Pour préciser que le nom a été écrit en français, on peut rajouter l'étiquette de langue (langage tag) @fr. Il est également préférable de préciser que 1978-02-08 est une date avec le suffixe ^^xsd:date, en précisant la valeur du préfixe xsd: au début des triplets.

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
```

¹⁷ Pour en savoir plus sur le concept de négociation de contenu (*content negotiation* en anglais) et les bonnes pratiques pour la publication de données RDF, voir [La négociation de contenu - HTTP](#), [Cool URIs for the Semantic Web](#) et [Best Practice Recipes for Publishing RDF Vocabularies](#).

```
@prefix dc: <http://purl.org/dc/terms/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://www.idref.fr/026956845/id> a foaf:Person .
<http://www.idref.fr/026956845/id> skos:prefLabel "La Villemarqué, Théodore Hersart
(1815-1895 ; vicomte de)"@fr .
<http://www.idref.fr/026956845> dc:created "1978-02-08"^^xsd:date .
```

Réponse 3

Notez l'utilisation de la virgule et des points-virgules :

```
<https://www.idref.fr/029665337/id> a foaf:Person ; foaf:familyName "Berthou"@fr ;
foaf:givenName "Yves"@fr; foaf:nick "Eskalibor"@br, "Kaledvourc'h"@br.
```

Réponse 4

La notice IdRef de Théodore Hersart de la Villemarqué, un document, au sens de l'ontologie BIBO, a pour principal sujet la personne Théodore Hersart de la Villemarqué, que l'Abes a choisi d'identifier par l'URI `<http://www.idref.fr/026956845/id>`, soit l'URI de la notice, suivie du préfixe `/id`. Ces déclarations, si elles étaient exprimées en triplets dans le langage courant, pourrait donner :

```
# La notice IdRef de Théodore a pour principal sujet Théodore (la personne)
```

```
<http://www.idref.fr/026956845> a pour principal sujet
  <http://www.idref.fr/026956845/id> .
```

```
<http://www.idref.fr/026956845/id> est une Personne .
```

L'Abes a utilisé l'ontologie FOAF (Friend Of A Friend) pour déclarer que Théodore est une personne (sa ressource est une instance de la classe `foaf:Person`) et que sa notice a pour principal sujet (la propriété `foaf:primaryTopic`) Théodore. Ce qui donne en Turtle :

```
<http://www.idref.fr/026956845> foaf:primaryTopic <http://www.idref.fr/026956845/id> .
<http://www.idref.fr/026956845/id> a foaf:Person .
```

Il n'existe pas de propriété dans l'ontologie FOAF (Friend of a Friend), qui permette par exemple d'écrire :

```
<http://www.idref.fr/026956845/id> foaf:birthDate "1815-07-07" .
```

La date de naissance est donc exprimée sous la forme d'un événement `<http://www.idref.fr/026956845/birth>` de type naissance (`ns2:Birth`) à l'aide de l'ontologie BIO. L'URI de BIO (<http://purl.org/vocab/bio/0.1/>) ne renvoie pas de triplets mais une page HTML décrivant ses classes et ses propriétés. En haut de cette page HTML, on trouve un lien vers sa version RDF/XML : <https://vocab.org/bio/schema.rdf>.

```
# Déclaration de la date de naissance de Théodore Hersart de la Villemarqué à l'aide de
```

l'ontologie BIO.

@prefix ns2: <http://purl.org/vocab/bio/0.1/> .

<http://www.idref.fr/026956845/id> ns2:event <http://www.idref.fr/026956845/birth> .

<http://www.idref.fr/026956845/birth>

a ns2:Birth ;

ns2:date "1815-07-07" .

La date de naissance n'est donc pas directement accessible comme une propriété de la ressource correspondant à Théodore, <http://www.idref.fr/026956845/id> , mais comme la propriété d'une autre ressource, instance de la classe bio:Birth.

Et le sémantique dans tout ça ?

Le Web sémantique n'a rien à voir avec la sémantique, l'étude scientifique et philosophique du langage.

Le Web sémantique¹⁸ est la diffusion sur le Web de données RDF en utilisant des ontologies exprimées avec le langage [OWL](#) (Web Ontology Language)¹⁹ et/ou RDFS.

La notion de **sémantique** est évoqué dès 1994, [dans une intervention de Tim Berners-Lee](#) lors de la première conférence internationale du World Wide Web²⁰. Tim Berners-Lee évoque le besoin d'un **Web sémantique** où le contenu des pages web et les liens qui existent entre elles puissent avoir du sens pour des machines. En effet, à moins de faire intervenir des logiciels d'intelligence artificielle, rien ne permet à un ordinateur de "savoir" que la notice de Théodore Hersart de la Villemarqué est un document au sujet d'une personne.

Linked data, Linked Open Data

Il est tout à fait possible d'utiliser les technologies du Web sémantique (RDF, OWL, RDFS, SPARQL) sans pour autant respecter les principes du Linked Data (Web des données, en français) définis en 2006 par Tim Berners-Lee, co-créateur du Web et président du W3C dans un [design issue](#) accessible en ligne. Comme le précise Tim Berners-Lee dans l'entête, il ne s'agit que de son point de vue personnel, imparfait mais toutefois public. Il ne s'agit donc pas d'une recommandation du W3C.

Le texte qui suit est une tentative de traduction et un résumé de ce document.

Pour TBL, le Web sémantique ne consiste pas seulement à publier des données sur le Web. Il consiste à créer des liens [entre les données] de manière à ce qu'une personne ou une

¹⁸ Voir la [définition du Web sémantique](#) sur Wikipédia en anglais.

¹⁹ Une ontologie, dans le sens du Web sémantique, est juste un document OWL (source : [OWL - Semantic Web Standards](#)).

²⁰ <https://www.iw3c2.org/>

machine puisse explorer le Web des données. Avec des données liées (linked data), vous pouvez trouver d'autres données liées.

Il suggère donc quatre règles pour publier des données sur le Web :

1. Use URIs as names for things (*Utiliser des URI pour nommer les choses*)
2. Use HTTP URIs so that people can look up those names. (*Utiliser des URI HTTP de manière à ce que les gens puisse y accéder*)
3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL) (Quand quelqu'un cherche à accéder à la ressource, lui fournir des informations utiles en utilisant les standards RDF et SPARQL.)
4. Include links to other URIs. so that they can discover more things. (créer des liens vers d'autres URI de manière à ce d'autres informations puissent être trouvées)

Puis TBL détaille les raisons pour lesquelles il a écrit ces règles. La deuxième règle vient du fait que des moyens d'identifier des ressources ont été inventés DOI (Digital Object Identifier), LSID (Life Science Identifiers), XRI (eXtensible Resource Identifier) là où les URI HTTP auraient très bien pu faire l'affaire.

Déduire des triplets : les inférences

Les ontologies servent à déclarer des classes et des propriétés mais elles permettent également d'**inférer** de nouvelles informations.

Par exemple, la propriété²¹ `owl:inverseOf` permettra de déclarer dans une ontologie "Vie quotidienne" que *soigné* est le contraire de *soigné par*²².

```
@prefix viqo: <http://viequotidienne.org/> .

viqo: a owl:Ontology ;
      dc:title "L'ontologie de la vie quotidienne" .

viqo:soigne a owl:ObjectProperty ;
      owl:inverseOf viqo:soigne_par .

viqo:soigne_par a owl:ObjectProperty .
```

La propriété `owl:SymmetricProperty` permettra de déclarer que le terme époux est symétrique : si A est l'époux de B alors B est l'époux de A.

```
@prefix viqo: <http://viequotidienne.org/> .
```

²¹ Les ontologies du Web sémantiques sont écrites avec des triplets RDF où les prédicats sont des propriétés des super-ontologies OWL et RDFS. Autrement dit, les ontologies permettent de définir des propriétés avec des triplets utilisant des propriétés des ontologies OWL et RDFS.

²² Ces exemples sont inspirés du livre de Bob Ducharme, Learning SPARQL (O'Reilly, 2013)

```
viqo:soigne a owl:ObjectProperty ;  
    owl:inverseOf viqo:soigne_par .  
  
viqo:soigne_par a owl:ObjectProperty .  
  
viqo:epoux a owl:SymmetricProperty .
```

Les propriétés **rdfs:domain** et **rdfs:range** permettront de préciser que le sujet d'un triplet ayant pour propriété viqo:soigne appartient de facto à la classe viqo:Medecin et son prédicat, à la classe viqo:Patient.

```
@prefix viqo: <http://viequotidienne.org/> .  
  
viqo: a owl:Ontology ;  
    dc:title "L'ontologie de la vie quotidienne" .  
  
viqo:Medecin a owl:Class .  
  
viqo:no_adeli a owl:ObjectProperty ;  
    rdfs:domain viqo:Medecin .  
  
viqo:soigne a owl:ObjectProperty ;  
    rdfs:domain viqo:Medecin ;  
    rdfs:range viqo:Patient .  
  
viqo:soigne a owl:ObjectProperty ;  
    owl:inverseOf viqo:soigne_par .  
  
viqo:soigne_par a owl:ObjectProperty .  
  
viqo:epoux a owl:SymmetricProperty .
```

Aussi si on publie les données concernant Martine avec cette ontologie :

```
@prefix vmart: <http://laviedemartine.org/> .  
@prefix viqo: <http://viequotidienne.org/> .  
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
  
vmart:pn001 a foaf:Person ;  
    foaf:name "Martine" ;  
    viqo:epoux vmart:pn002 ;  
    viqo:soigne_par vmart:pn003 .  
  
vmart:pn002 a foaf:Person ;
```



```
foaf:name "Bernard" .

vmart:pn003 a foaf:Person ;
foaf:name "Juliette" .
```

On pourra déduire (on dit aussi *inférer*) les triplets suivants :

```
@prefix vmart: <http://laviedemartine.org/> .
@prefix vico: <http://viequotidienne.org/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

vmart:pn002 vico:epoux vmart:pn001 . # Bernard est l'époux de Martine
vmart:pn003 vico:soigne vmart:pn001 . # Juliette soigne Martine
vmart:pn003 a vico:Medecin . # Juliette est un médecin, elle peut avoir un n°Adeli
(propriété vico:no_adeli)
```

Remarquez au passage qu'une ressource peut appartenir à plusieurs classes. Juliette appartient aux classes `foaf:Person` et `vico:Medecin`.

Les logiciels capables d'interpréter les ontologies pour faire des inférences sont appelés moteurs d'inférences (*inference engines*). GraphDB, le logiciel de base de données RDF que nous utiliserons dans le cours sur SPARQL intègre un moteur d'inférence. Il est désactivé par défaut, car il n'y a pas de miracle : le moteur ne "déduit" pas les inférences comme le ferait l'intelligence humaine, ni ne crée de nouveaux triplets à la demande, en fonction des besoins du moteur de requête SPARQL, mais il crée tous les triplets déductibles à partir des données saisies et des règles OWL et RDFS, ce qui peut augmenter de manière conséquente la taille de la base.

Réutiliser des ontologies existantes ou créer sa propre ontologie

RDFS et OWL sont des vocabulaires standards du W3C pour définir et décrire les classes et les propriétés de triplets RDF. Ils ne fonctionnent pas comme des modèles auxquels doivent se conformer les données, tel un schéma de base de données relationnelle. Par exemple, il n'est pas obligatoire de déclarer que Paul appartient à la classe `vico:Medecin` pour créer un triplet avec son n°Adeli²³.

Selon la complexité des données à exprimer, il peut être plus avantageux d'utiliser une ou plusieurs ontologies existantes. Le site [LOV](#) (Linked Open Vocabularies) permet de faire des recherches dans les classes et les propriétés de 714²⁴ vocabulaires.

²³ Un numéro Adeli est un identifiant national attribué à un médecin lors de son enregistrement en Agence régionale de santé.

²⁴ LOV référence 714 vocabulaires au 9 juillet 2020.

Une fois les ontologies choisies, il n'est pas toujours facile de s'y retrouver dans la multitude de classes et de propriétés. Pour naviguer visuellement dans une ontologie, vous pouvez utiliser [WebVOWL](#) ou [OWLGrEd](#).

Si aucune des ontologies existantes ne satisfait vos besoins, le logiciel [Protégé](#), développé par l'université de Stanford, vous évitera d'écrire votre ontologie à la main dans un fichier texte et de connaître par coeur toutes les subtilités de OWL (dont il existe plusieurs versions ²⁵).

Au départ, il y avait RDF, RDFS et OWL

A ce stade, si on ne devrait retenir que trois documents à consulter pour comprendre le RDF, ce serait les ontologies RDF, RDFS²⁶ et OWL dans lesquelles sont définies (avec des triplets Turtle RDF, bien entendu) les concepts de base du Web sémantique. Pour éditer les fichiers en liens dans le tableau ci-dessous, ajoutez l'extension .ttl et ouvrez-les avec le logiciel Notepad++ (sous Windows).

préfixe	nom officiel	URI	commentaire
rdf:	The RDF Concepts Vocabulary (RDF)	http://www.w3.org/1999/02/22-rdf-syntax-ns#	Cette ontologie définit notamment <code>rdf:type</code> et <code>rdf:Property</code> (qui est une instance de <code>rdfs:Class</code>). Elle définit aussi les concepts de déclaration, sujet, prédicat, objet (<code>rdf:Statement</code> , <code>rdf:subject</code> , <code>rdf:predicate</code> , <code>rdf:object</code>). La recommandation du W3C correspondante est RDF 1.1 Concepts .
rdfs:	The RDF Schema vocabulary (RDFS)	http://www.w3.org/2000/01/rdf-schema#	Cette ontologie définit la notion de ressource, classe, littéral (<code>rdfs:Resource</code> , <code>rdfs:Class</code> , <code>rdfs:Literal</code>), les propriétés <code>rdfs:label</code> et <code>rdfs:comment</code> ainsi que deux propriétés qui servent de fondement aux inférences : <code>rdfs:range</code> et <code>rdfs:domain</code> . La recommandation du W3C correspondante doit être RDF Schema 1.1 .
owl:	The OWL 2 Schema vocabulary (OWL 2)	http://www.w3.org/2002/07/owl#	Cette ontologie décrit les classes et les propriétés de la spécification OWL 2 Web Ontology Language RDF-Based Semantics . Elle définit notamment owl:Class et les propriétés <code>owl:sameAs</code> (vu précédemment, pour lier des ressources équivalentes), <code>owl:inverseOf</code> , <code>owl:SymmetricProperty</code> .

Pas facile de s'y retrouver avec les propriétés qui relèvent du domaine `rdf:` et celles qui

²⁵ OWL Lite, OWL DL, OWL Full, OWL 2 EL, OWL 2 QL et OWL 2 RL

²⁶ Voir aussi le très instructif [How are RDF and RDFS related?](#)

relèvent du domaine `rdfs:`. Par ex. écrit-on `rdf:label` ou `rdfs:label` ? Ce qu'il faut retenir c'est que parmi les propriétés les plus courantes des ontologies RDF et RDFS, seules **`rdf:type`** et **`rdf:Property`** proviennent de RDF toutes les autres propriétés sont définies dans RDFS, même `rdfs:label` :

- **`rdf:`**
 - `rdf:type`
 - `rdf:Property`
- **`rdfs:`**
 - `rdfs:label`
 - `rdfs:comment`
 - `rdfs:Class` (`owl:Class` est une instance de `rdfs:Class`)
 - `rdfs:domain`
 - `rdfs:range`

Comme méthode mnémotechnique, on peut se souvenir du triplet suivant qui déclare que `dc:creator` (*a pour créateur* selon l'ontologie Dublin Core) est une propriété (`rdf:Property`) :

```
dc:creator rdf:type rdf:Property .
```

Exercices

Exercice 1

Comment exprimer en Turtle et avec l'ontologie FOAF que Théodore Hersart de la Villemarqué connaît Alfred Potier de Courcy et que Théodore est membre de l'**Académie des sciences de Berlin** (utiliser les trois ressources IdRef correspondant à Théodore, Alfred et à l'Académie).

Exercice 2

A quelles classes appartiennent la ressource `<http://www.idref.fr/026956845/id>` ?

Réponses

Réponse 1

Selon la documentation de l'ontologie FOAF (<http://xmlns.com/foaf/spec/>), la propriété `foaf:knows` a pour *domain* une ressource de type `foaf:Group` et pour *range* une ressource de type `foaf:Agent`. C'est à dire que l'on ne peut pas écrire le triplet :

Théodore **membre** Académie des sciences de Berlin .

... mais le triplet :

Académie des sciences de Berlin **membre** Théodore .

Soit avec les ressources correctes :

```
<http://www.idref.fr/026956845/id> foaf:knows <https://www.idref.fr/029867495/id> .  
<http://www.idref.fr/077231082/id> foaf:member <http://www.idref.fr/026956845/id> .
```

Réponse 2

Quand on converti en Turtle avec EasyRDF les triplets de

<http://www.idref.fr/026956845/id>, on constate qu'il a pour type explicite foaf:Person (un triplet exprime explicitement l'appartenance de la ressource à la classe foaf:Person) et pour classe implicite owl:Thing (inféré par la propriété foaf:primaryTopic). On pourrait également dire que Théodore appartient également à la classe foaf:Agent, puisque foaf:Person est une sous-classe de foaf:Agent (voir http://xmlns.com/foaf/spec/#term_Person).

```
<http://www.idref.fr/026956845>  
  a bibo:Document ;  
  foaf:primaryTopic <http://www.idref.fr/026956845/id> .
```

```
<http://www.idref.fr/026956845/id> a foaf:Person .
```

Où sont les triplets ? SPARQL endpoint et bases de données de graphes

Les triplets peuvent être :

1. écrits dans des fichiers texte stockés sur votre ordinateur dans le vocabulaire de votre choix (**Turtle**, **RDF/XML**, etc.).
2. accessibles depuis l'**URI** d'une ressource RDF sous forme de RDF/XML comme les notices de l'Abes. Utile pour récupérer les triplets d'une notice mais moins pratique pour récupérer les triplets de toutes les notices (voir 5.).
3. intégrés dans le code HTML de pages avec du **RDFa**²⁷, une recommandation du W3C pour insérer des triplets RDF dans des pages Web en XHTML²⁸.
4. accessibles par une API qui délivre du JSON-LD. Le **JSON-LD** est une recommandation du W3C pour sémantiser des données au format JSON.
5. stockés dans une base de données de graphe qui supporte le RDF appelé **triple store** en anglais. L'avantage de cette dernière solution est qu'elle permet d'interroger l'intégralité d'un jeu de données RDF à travers un **SPARQL endpoint**, un service web qui accepte les requêtes en SPARQL.

Les triplets de IdRef sont accessibles par les URI des notices mais comment interroger toutes les notices à la fois ? Réponse : Avec le **SPARQL endpoint** de data.idref.fr.

Quand on accède à cette URL avec un navigateur, le SPARQL prend l'aspect d'une page web, mais il ne s'agit que de son interface utilisateur. Vous pouvez aussi lancer des requêtes

²⁷ [Introduction à RDFa](#)

²⁸ Voir aussi <https://schema.org/docs/gs.html> sur Schema.org

sur un SPARQL endpoint avec des logiciels tel que [ARQ](#), un processeur SPARQL en ligne de commande, ou depuis un autre SPARQL endpoint (selon le principe des recherche fédérées, que nous verrons dans le cours sur le SPARQL) ou depuis d'autres interfaces, telles que [Yasgui](#)²⁹, un éditeur SPARQL qui propose la coloration syntaxique, la validation de la syntaxe, l'autocomplétion et des outils de visualisation des résultats (graphes, cartes, timelines, etc.) ou [Sparklis](#), développé par l'enseignant-chercheur Sébastien Ferré et utilisé par [Persée](#), un portail de publications scientifiques.

💡 A noter :: Un SPARQL endpoint n'est pas la base de données de triplets mais le point d'accès pour lancer des requêtes en SPARQL dans une base de données de triplets RDF (*triple store*).

Dans le cours sur SPARQL, nous utiliserons la version gratuite (mais pas open source) du triple store [GraphDB](#), édité par la société bulgare Ontotext.

Liste non exhaustive de triple stores, pour une liste plus complète, voir [Comparison of triplestores](#) sur Wikipédia) :

- [GraphDB](#) (qui sera utilisé lors durant l'ANF. Existe en version gratuite non-open source et en version commerciale),
- [OpenLink Virtuoso](#) (utilisé par l'Abes),
- [TDB](#) du projet Apache Jena,
- [AllegroGraph](#),
- [MarkLogic Data Hub](#),
- [BlazeGraph](#) (utilisé par la base de connaissances Wikidata).

Quelques exemples de SPARQL endpoint

Il n'existe pas de Google du Web sémantique, ni d'annuaire des SPARQL endpoint. Les exemples ci-dessous ne sont donc pas forcément représentatifs des données sémantiques disponibles publiquement sur internet.

En France, la Bibliothèque nationale de France, l'Agence bibliographique pour l'enseignement supérieur, [Isidore](#) (le moteur de recherche en sciences humaines et sociales), Persée, les archives ouvertes Hal, diffusent des métadonnées en RDF à travers des SPARQL endpoint : [data.bnf.fr](#), [data.idref.fr](#), [http://data.persee.fr/explorer/sparql-endpoint/](#), [https://isidore.science/sqe](#), [https://data.archives-ouvertes.fr/](#).

A l'international, le SPARQL endpoint du British Museum, pionnier dans la diffusion de données RDF en 2011³⁰ n'est plus accessible à ce jour³¹. Mais le British Museum continue à développer des projets liés au Web sémantique tel que [ResearchSpace](#).

Au niveau européen, [Europeana](#), l'[Office européen des brevets](#), [European Data Portal](#) (le portail des données européennes) proposent d'accéder à leurs données / métadonnées par

²⁹ L'interface Yasgui de l'Abes est accessible à l'adresse [https://data.idref.fr/yasgui.html](#).

³⁰ [Experimenting with British Museum data. Overdue Ideas. Owen Stephens](#)

³¹ Le SPARQL endpoint du British Museum n'était pas disponible en été 2020.

des SPARQL endpoints.

Le projet DBpedia qui s'était donné comme objectif de transformer en RDF les cartouches d'informations des Wikipédia est toujours actif.

La base de connaissance Wikidata dispose d'un remarquable SPARQL endpoint, avec une interface multilingue, un assistant de requête et un service d'autocomplétion³² et surtout, la possibilité d'interroger de manière fédérée d'autres SPARQL endpoint³³.

Les alternatives au SPARQL et au Web sémantique

Dans le monde des développeurs informatiques, le Web sémantique est loin d'avoir été adopté.

Dans l'enquête 2020 de Stack Overflow³⁴, un site international d'entraide en développement informatique, SPARQL et RDF n'apparaissent même pas parmi les technologies utilisées ou apprises par les 65.000 personnes interrogées.

Au 3 septembre 2020, 4970 questions sur Stack Overflow avaient pour tag SPARQL pour 12.103 questions avec le tag GraphQL et 557.884 avec le tag SQL (2.080.486 avec le tag Javascript, 1.709.292 avec le tag Java, 1.527.042 questions avec le tag Python).

Ce désintérêt peut peut-être s'expliquer par le fait que le Web sémantique a été conçu par le W3C, qui s'il est un organisme reconnu par la communauté des développeurs web (HTML et CSS sont des recommandations du W3C ainsi que SVG, PNG, XML, etc.), la communauté plus large des développeurs suit certainement plus les technologies adoptées ou développées par Google, Apple, Facebook, Amazon et Microsoft. Or, le moins que l'on puisse dire, c'est que le Web sémantique ne fait pas partie de la stratégie des GAFAM.

Ces dernières années, de nouvelles technologies sont apparues telles que le langage et environnement d'exécution **GraphQL**³⁵, créé par Facebook en 2012 ou les langages de bases de données de graphes **Cypher**, créé par la société Neo4J Inc. en 2011, et **Gremlin** de Apache Tinkerpop³⁶.

Ces technologies n'ont pas la même ambition que le Web sémantique, dont le but ultime était de lier les données du web entre elles et de leur donner du sens pour les machines, grâce aux ontologies. Mais elles ont le mérite de proposer des solutions clé en main pour diffuser des données sur le web ou stocker des données sous forme de graphes.

³² https://www.wikidata.org/wiki/Wikidata:SPARQL_tutorial#Autocompletion

³³ data.idref.fr, le SPARQL endpoint de IdRef offre également la possibilité d'interroger de manière fédérée des SPARQL endpoint "amis" (BnF, Persée, Wikidata, Larhra, Hal, Isidore).

³⁴ [Stack Overflow Developer Survey 2020](#)

³⁵ Des sites tels que Allociné, Dailymotion, airbnb ont adopté GraphQL pour le partage de données en interne ou avec leurs partenaires. Voir [Who's Using GraphQL?](#) et [GraphQL Landscape](#).

³⁶ Apache Tinkerpop et Gremlin sont aux bases de données de graphes ce que JDBC et SQL sont aux bases de données relationnelles (source : [Gremlin \(query language\)](#) sur Wikipédia en anglais).

Si vous souhaitez diffuser les données de vos bases de données relationnelles, sans soucis d'**interopérabilité**, utilisez une API ou le GraphQL.

Si vous exploitez des données complexes structurées nativement sous forme de graphes, les systèmes de gestion de bases de données Neo4j (basé sur la langage de requête Cypher) ou OrientDB (basé sur le langage de requête Gremlin), seront peut-être plus adaptées.

Mais si vous avez des bases de données relationnelles dont vous souhaitez diffuser le contenu avec un soucis d'opérabilité avec des référentiels, en donnant la possibilité à vos utilisateurs de faire des requêtes complexes allant au delà de celles permises par des API, de déduire des nouvelles informations, alors le Web sémantique est peut-être fait pour vous.

Ce qui est certain, c'est qu'une technologie n'est pas meilleure qu'une autre. Chacune répond à des besoins spécifiques, à vous de déterminer les vôtres.

Pour en savoir plus

Il existe un seul livre entièrement dédié au SPARQL, paru chez l'éditeur américain O'Reilly : DuCharme, B. (2013). *Learning SPARQL : Querying and updating with SPARQL 1.1* (Second edition). O'Reilly Media ([site du livre](#) avec les triplets RDF et les requêtes SPARQL utilisées dans le livre). L'auteur, Bob DuCharme tient un blog : <http://www.bobdc.com/blog/>.

Sur FUN (France Université Numérique), le MOOC [Web sémantique et Web de données](#) de l'Inria est ouvert jusqu'au 31 janvier 2021. Ses auteurs ont publié un livre chez Dunod en 2012 : Gandon, F., Faron-Zucker, C., & Corby, O. (2012). *Le Web sémantique*. Dunod.