

RDF

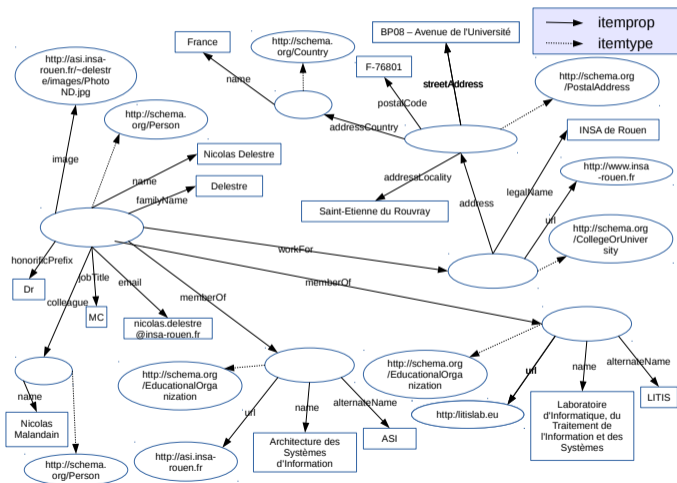
Cours « Recherche d'Information et Graphe de données »

Nicolas Delestre

Plan...

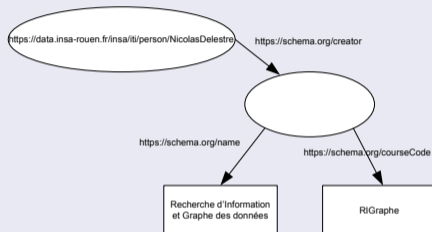
- 1 Vers le Web des données
- 2 URI
- 3 RDF
- 4 Les classes
- 5 La réification
- 6 La publication
- 7 Wikidata
- 8 Conclusion

Les métadonnées de <https://delestre.pages.insa-rouen.fr/siteweb/>



Constats

- Certains éléments ne sont pas nommés alors qu'ils ont un rôle, ils représentent quelque chose (une personne, un établissement, une organisation pédagogique, etc.)
- Alors que la page décrit une personne ces métadonnées décrivent plus de choses
- Il y a deux types de liens alors que les deux servent à caractériser une entité
- Le même fichier, le même URL, rassemble des informations destinées aux humains et aux machines



Définition

- Le Web des données est le Web pour les machines :
 - « le Web » implique l'utilisation (et son extension) des technologies du Web (URI, HTTP) et des principes du Web (informations décentralisées et interconnectées)
 - « pour les machines » implique informations structurées

Le web des données 2 / 2

Plus exactement

	WEB du Texte	WEB des Données	WEB Sémantique
Standards et Outils	URL HTTP HTML	URI HTTP RDF SPARQL	URI HTTP RDF SPARQL
	DTD XSD Moteur de recherche	RDFS Moteur de requêtes	OWL Moteur de requêtes
Techniques	Recherche d'Information basée sur des index de mots Monde des documents	Mise en correspondance de graphes Monde des bases de données	Logique Monde des bases de connaissances

URL, URI et IRI

Des URL aux URI

- Les URL permettent de localiser une ressource (que l'on nomme très souvent document) sur le Web
- Les URI (et IRI) permettent de référencer une ressource (au sens large : un objet, un concept, une idée, etc.) sur le Web
- Si à partir d'un URI on peut obtenir (à l'aide du requête HTTP GET) un document qui décrit la ressource, on dit que cet URI est déréférençable

International Resource Identifier

- Les URI sont limités à un ensemble de caractères (ASCII)
- Les IRI permettent d'utiliser des caractères Unicode
- Exemple : `http://www.example.com/id/cole`

RDF

Définition

- RDF (*Resource Description Framework*) est un formalisme de représentation de métadonnées proposé par le W3C en 1999
- RDF permet de faire du Web des données, c'est à dire de structurer les données sur le Web pour les rendre compréhensibles par les machines

Un peu d'histoire

- En 1997, Tim Berners-Lee propose une première version du RDF (Resource Description Framework) pour permettre de faire du Web des données
- En 1999, le W3C publie la première recommandation RDF
- En 2004, le W3C publie la recommandation RDF version 1.0
- En 2014, le W3C publie la recommandation RDF version 1.1
- En 2025-2026, le W3C publie le *draft* de la recommandation RDF version 1.2



Documentations disponibles

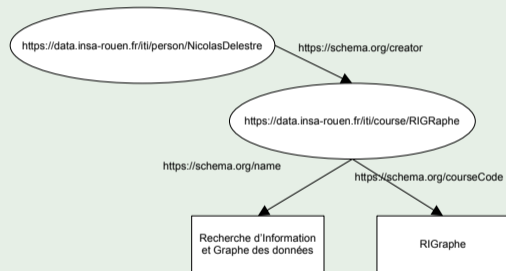
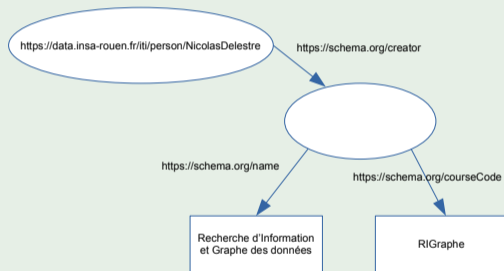
- *Primer* présentation du RDF pour les débutants avec des exemples concrets
- *Concept an Abstract Data Model* présentation du modèle de données RDF (graphes RDF et éléments des graphes)
- *Semantics* présentation de la sémantique du RDF (interprétation logique des graphes RDF)
- *Schema* présentation du RDF Schema (RDFS) pour la description de schémas RDF
- **XML Syntax, Turtle, TriG, N-Triples, N-Quads, JSON-LD, RDFa** présentation des différentes syntaxes de sérialisation du RDF

Concepts de base

- Un triplé composé de :
 - Sujet ou Ressource : URI de la ressource sur laquelle s'applique la métadonnée
 - Prédicat ou Propriété : URI de la métadonnée appliquée à la ressource
 - Objet ou valeur : valeur de la métadonnée (peut être une ressource ou un littéral)



Possibilité de structurer l'information



L'élément XML de base : *Description*

- Ressource : Attribut *about*
- Prédicat : Élément XML inclus provenant d'un schéma (voir RDFS)
- Valeur : Spécifiée dans l'élément inclus

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:schema="http://schema.org">
  <rdf:Description rdf:about="https://data.insa-rouen.fr/iti/person/NicolasDelestre">
    <schema:name>Nicolas Delestre</schema:name>
  </rdf:Description>
</rdf:RDF>
```

- On peut avoir plusieurs descriptions faisant référence à plusieurs schémas RDFS

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:schema="http://schema.org/"
  xmlns:foaf="http://xmlns.com/foaf/0.1/">
  <rdf:Description rdf:about="https://data.insa-rouen.fr/iti/person/NicolasDelestre">
    <schema:name>Nicolas Delestre</schema:name>
  </rdf:Description>
  <rdf:Description rdf:about="https://data.insa-rouen.fr/iti/person/NicolasDelestre">
    <foaf:name>Nicolas Delestre</foaf:name>
  </rdf:Description>
</rdf:RDF>
```

- Les valeurs peuvent être des littéraux ou des ressources
- L'attribut *datatype* permet de typer les littéraux (possibilité d'utiliser les types *xsd*)
- Possibilité d'utiliser les *ENTITY* des DTD

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [

```

- Les attributs *nodeID* et *id* permettent de créer des nœuds vides

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:schema="http://schema.org/">
  <rdf:Description rdf:about="https://data.insa-rouen.fr/iti/person/NicolasDelestre">
    <schema:name>Nicolas Delestre</schema:name>
    <schema:creator rdf:nodeID="rigraphe"/>
  </rdf:Description>
  <rdf:Description rdf:nodeID="rigraphe">
    <schema:name>Recherche d'Information et Graphe de données</schema:name>
    <schema:courseCode>RI Graphe</schema:courseCode>
  </rdf:Description>
</rdf:RDF>
```

Bag, sequence et alternative

- Il est fréquent de devoir faire référence à plusieurs ressources (par exemple « un livre écrit par plusieurs auteurs »)
- RDF propose trois conteneurs :
 - bag** Liste non ordonnée de ressources ou de littéraux (élément *bag*)
 - sequence** Liste ordonnée de ressources ou de littéraux (élément *seq*)
 - alternative** Liste de ressources ou de littéraux qui représentent des alternatives pour une valeur unique (élément *alt*)
- Les éléments du conteneur sont listés à l'aide de l'élément *li* ou *_i* avec ($i > 0$)
 - Pour alternative il doit y avoir au moins un élément *_1* (élément par défaut)

Bag

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:schema="http://schema.org/"
  xmlns:foaf="http://xmlns.com/foaf/0.1/">
  <rdf:Description rdf:about="https://data.insa-rouen.fr/iti/person/NicolasDelestre">
    <schema:name>Nicolas Delestre</schema:name>
    <schema:creator>
      <rdf:Bag>
        <rdf:li rdf:resource="https://data.insa-rouen.fr/iti/course/Algo"/>
        <rdf:li rdf:resource="https://data.insa-rouen.fr/iti/course/Python"/>
        <rdf:li rdf:resource="https://data.insa-rouen.fr/iti/course/Compil"/>
        <rdf:li rdf:resource="https://data.insa-rouen.fr/iti/course/RIGraphe"/>
      </rdf:Bag>
    </schema:creator>
  </rdf:Description>
</rdf:RDF>
```

Turtle 1 / 2

Une autre méthode de sérialisation

- La syntaxe XML est verbeuse, peut être difficile à lire (par les humains)
- Une autre syntaxe *Turtle* (N3) a été proposée par Tim Berners-Lee pour être encore plus facile à lire
- Syntaxe simple :

```

@prefix ns: <URI>.
...

<ns:Sujet>
  <Predicat1> <Valeur1>;
  <Predicat2> <Valeur21>,<Valeur22>,...;
  ...
  <Predicatn> <Valeurn>.

[
  <Predicat1> <Valeur1>.
  <Predicat2> <Valeur21>,<Valeur22>,...;
  ...
  <Predicatn> <Valeurn>.
]

```



Turtle 2 / 2

XML RDF

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:schema="http://schema.org/"
  xmlns:foaf="http://xmlns.com/foaf/0.1/">
  <rdf:Description rdf:about="https://data.insa-rouen.fr/iti/person/NicolasDelestre">
    <schema:name rdf:datatype="&xsd;token">Nicolas Delestre</schema:name>
    <foaf:name>Nicolas Delestre</foaf:name>
  </rdf:Description>
</rdf:RDF>
```

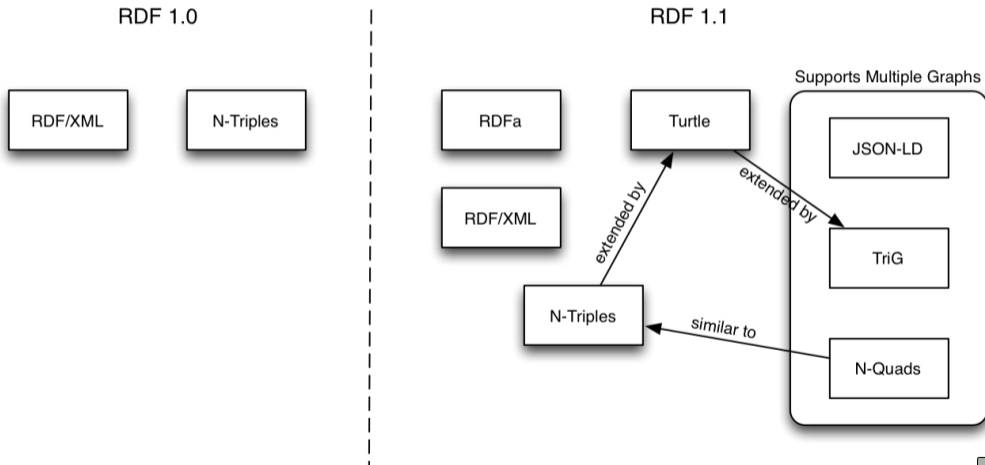
Turtle

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix schema: <http://schema.org/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

<https://data.insa-rouen.fr/iti/person/NicolasDelestre>
  schema:name "Nicolas Delestre"^^<http://www.w3.org/2001/XMLSchema#token> ;
  foaf:name "Nicolas Delestre" .
```



D'autres syntaxes



Classe 1 / 2

- En RDF il est possible d'associer à une ressource une ou plusieurs classes :
 - en utilisant la propriété `rdf:type`
 - en utilisant le nom de la classe pour la balise décrivant la ressource en remplacement `rdf:Description` (sérialisation XML)
 - en utilisant la propriété `a` (sérialisation N3)
- Cela permet de connaître l'ensemble des propriétés valides pour cette ressource, mais contrairement à la modélisation orientée objet :
 - on n'est pas obligé d'utiliser toutes les propriétés pour décrire un individu de la classe
 - on peut utiliser d'autres propriétés pour décrire un individu

Nicolas Delestre comme enseignant et chercheur

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix schema: <http://schema.org/> .
@prefix aiiso-roles: <http://purl.org/vocab/aiiso-roles/schema#> .

<https://data.insa-rouen.fr/insa/iti/person/NicolasDelestre>
  rdf:type aiiso-roles:Lecturer ,
          aiiso-roles:Research_Associate ;
  schema:name "Nicolas Delestre" .
```

Principe

- Parfois, il est nécessaire de faire référence à une déclaration RDF (par exemple pour indiquer la source de cette déclaration)
- La réification permet de faire référence à une déclaration RDF en la transformant en ressource RDF

Avant RDF 1.2

- Pour réifier une déclaration RDF, on crée une ressource RDF qui a pour propriétés :
 - `rdf:type` avec comme valeur `rdf:Statement`
 - `rdf:subject` avec comme valeur le sujet de la déclaration RDF
 - `rdf:predicate` avec comme valeur le prédicat de la déclaration RDF
 - `rdf:object` avec comme valeur l'objet de la déclaration RDF

La réification 2 / 4

Les différents noms du département

```

@prefix rdf:      <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix schema:  <http://schema.org/> .
@prefix prov:    <http://www.w3.org/ns/prov#> .
@prefix xsd:     <http://www.w3.org/2001/XMLSchema#> .
@prefix insarouen: <https://data.insa-rouen.fr/> .
@prefix iti:     <https://data.insa-rouen.fr/iti/> .

iti:departement a schema:EducationalOrganization .

insarouen:insa a schema:CollegeOrUniversity ;
  schema:name "INSA Rouen" .

# Les triplets "simples"
iti:departement schema:name "ASI" .
iti:departement schema:name "ITI" .
iti:departement schema:name "ISIA" .

# Réification du triplet : departement schema:name "ASI"
_:stmt1 a rdf:Statement ;
  rdf:subject iti:departement ;
  rdf:predicate schema:name ;
  rdf:object "ASI" ;
  prov:wasAttributedTo insarouen:insa ;
  prov:generatedAtTime "1999"^^xsd:gYear .

```

Les différents noms du département

```
# Réification du triplet : departement schema:name "ITI"
_:stmt2 a rdf:Statement ;
  rdf:subject iti:departement ;
  rdf:predicate schema:name ;
  rdf:object "ITI" ;
  prov:wasAttributedTo insarouen:insa ;
  prov:generatedAtTime "2015"^^xsd:gYear .

# Réification du triplet : departement schema:name "ISIA"
_:stmt3 a rdf:Statement ;
  rdf:subject iti:departement ;
  rdf:predicate schema:name ;
  rdf:object "ISIA" ;
  prov:wasAttributedTo insarouen:insa ;
  prov:generatedAtTime "2025"^^xsd:gYear .
```

Depuis RDF 1.2

- La ressource ou l'objet d'un triplé RDF peut être une déclaration RDF

Les différents noms du département

```
iti:departement a schema:EducationalOrganization .
```

```
insarouen:insa a schema:CollegeOrUniversity ;  
  schema:name "INSA Rouen" .
```

```
<< iti:departement schema:name "ASI" >>  
  prov:wasAttributedTo insarouen:insa ;  
  prov:generatedAtTime "1999"^^xsd:gYear .
```

```
<< iti:departement schema:name "ITI" >>  
  prov:wasAttributedTo insarouen:insa ;  
  prov:generatedAtTime "2015"^^xsd:gYear .
```

```
<< iti:departement schema:name "ISIA" >>  
  prov:wasAttributedTo insarouen:insa ;  
  prov:generatedAtTime "2025"^^xsd:gYear .
```

Comment publier des données sur le Web ?

Première méthode : une vision centrée sur les documents

- Tout ce que l'on veut décrire est désigné par un URI
- Les informations formelles sur cet URI sont les triplés dont le sujet est cet URI
- Publier des données sur le Web des données revient à publier des documents qui contiennent des triplés RDF qui ont pour sujet les URI que l'on veut décrire

Questions ?

- Comment obtenir ce document ?
- Comment faire pour que les humains et les machines puissent obtenir des descriptions différentes de la même ressource ?
- Qu'est-ce qu'un bon URI ?

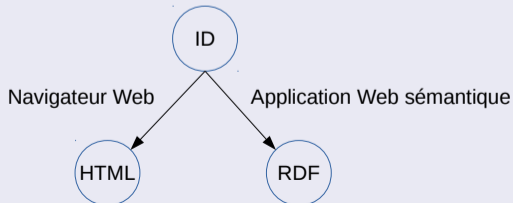
Qu'est ce qu'un bon URI ?

Critères

D'après <https://www.w3.org/TR/cooluris/>, un bon URI doit respecter les critères suivants :

- 1 Être sur le Web (déréférençable) : les humains ET les machines doivent pouvoir obtenir une description qui leur sont compréhensibles, documents HTML pour les humains, documents structurés pour les machines (utilisation des attributs `accept` et `content-type`)
- 2 Être non ambigu : **un URI ne peut pas être à la fois la référence à une ressource et la description de cette ressource**

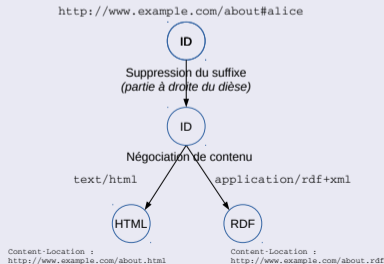
Identifiant d'une ressource (URI) déréférençable



Comment créer des bons URI? 1 / 3

Première solution : les *hash URIs* (Cf. <https://www.w3.org/TR/cooluris/>)

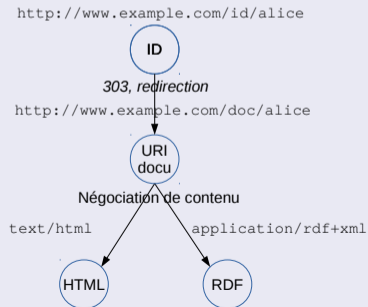
- Pour les petites structures, organismes, entreprises, etc. : un même document décrit toutes les ressources
- Exemple :
 - `http://www.example.com/about#exampleinc`
 - `http://www.example.com/about#alice`
- Mécanisme :



Comment créer des bons URI? 2 / 3

Deuxième solution : 303 See Other (Cf. <https://www.w3.org/TR/cooluris/>)

- Pour les grandes structures, organismes, entreprises, etc. : un document par ressource
- Exemple :
 - <http://www.example.com/id/exampleinc>
 - <http://www.example.com/id/alice>
- Premier cas : il n'y a pas de différences informationnelles entre le document pour les humains du document pour les machines
 ⇒ un seul URI pour désigner ce document + deux versions du même document

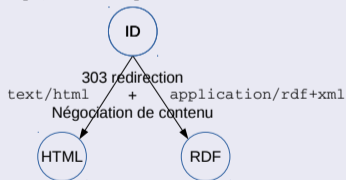


Comment créer des bons URI? 3 / 3

Deuxième solution : *303 See Other* (Cf. <https://www.w3.org/TR/cooluris/>)

- Pour les grandes structures, organismes, entreprises, etc. : un document par ressource
- Exemple :
 - <http://www.example.com/id/exampleinc>
 - <http://www.example.com/id/alice>
- Deuxième mécanisme, lorsque il y a des différences informationnelles entre le document pour les humains du document pour les machines
⇒ deux documents

<http://www.example.com/id/alice>



Content-Location :
<http://www.example.com/people/alice.html>

Content-Location :
<http://www.example.com/data/alice.rdf>



Wikidata, un exemple de base de données RDF

Wikidata

- Wikidata est une base de données collaborative et libre qui contient des données structurées sur une grande variété de sujets
- Wikidata est un projet de la Wikimedia Foundation, la même organisation qui gère Wikipédia
- Wikidata utilise le format RDF pour présenter les données qu'elle contient (mais n'utilise pas `rdf:type`)
- Toutes les déclarations (`statement`) peuvent être justifiées (réification)
- En 2026, Wikidata contient plus de 120 millions d'items et plus de 1 milliard de déclarations RDF

Les préfixes

Les deux préfixes les plus souvent utilisés

- wd (<http://www.wikidata.org/entity/>) : pour les items de Wikidata
- wdt (<http://www.wikidata.org/prop/direct/>) : pour les propriétés de Wikidata

Construction d'un URI wikidata

- Les items de Wikidata ont des URI de la forme <http://www.wikidata.org/entity/Q42> (URI de l'item Q42)
- Les propriétés de Wikidata ont des URI de la forme <http://www.wikidata.org/prop/direct/P31> (URI de la propriété P31)

URI déréférencables

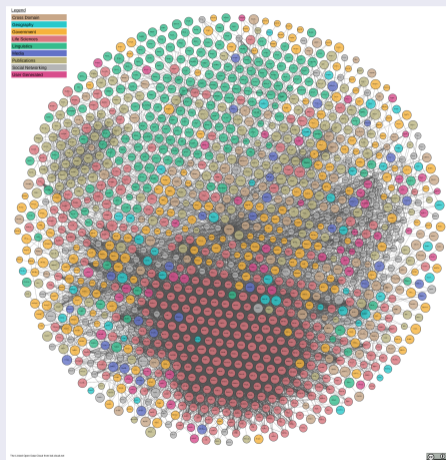
```
wget --header="Accept: application/rdf+xml" http://www.wikidata.org/entity/Q1665112
```



L'Open data défini par Tim Berners-Lee en 2006



Le web des données aujourd'hui source : <http://lod-cloud.net/> (2026)



1358 jeux de données

Ce que nous avons vu

- Les bases du RDF : triplé, conteneur, classes, réification
- Deux exemples de sérialisation : XML et Turtle
- Comment bien publier des données sur le Web avec une vision documentaire

Ce qui n'a pas été présenté

- Les autres méthodes de sérialisation
- La notion de graphe nommé et le *Global Giant Graph*
- **Comment publier des triplés RDF avec un point de vue base de données**
- **Comment créer des schémas, comment vérifier la validité de données RDF et comment inférer de nouveaux triplés RDF automatiquement**