Semantic Web handout including:  
lecture questions and practical sessions

In this document, you must provide your answers to the questions asked during the course **and** to the questions of the practical sessions; everything in one document.

The questions of the course have been repeated here; **do not delete the questions** but provide your answer to each question just below the question. You can use screenshots when appropriate as an answer to a question but keep your answers and this file as small and concise as possible.

At the end, you must generate and submit only one final PDF file based on this template.

In questions where you are asked to create, invent or use your own data, make sure they are different from other student’s.

First name: <ANSWER HERE/>

Family name: <ANSWER HERE/>

Email: <ANSWER HERE/>

**TABLE OF CONTENTS:**

[**QUESTIONS FROM THE COURSES** 2](#_Toc116378480)

[Questions from the course on Linked Data. 2](#_Toc116378481)

[Questions from the course on RDF. 5](#_Toc116378482)

[Questions from the course on SPARQL. 9](#_Toc116378483)

[Questions from the course on Ontologies. 14](#_Toc116378484)

[Questions from the course on RDFS 15](#_Toc116378485)

[Questions from the course on OWL. 18](#_Toc116378486)

[Questions from the course on Vocabularies. 22](#_Toc116378487)

[Questions from the course on other data formats. 27](#_Toc116378488)

[**LAB SESSIONS** 31](#_Toc116378489)

[Lab session on RDF. 31](#_Toc116378490)

[Lab session on SHACL. 34](#_Toc116378491)

[Lab session on SPARQL. 36](#_Toc116378492)

[Lab session on RDFS. 43](#_Toc116378493)

[Lab session on OWL. 49](#_Toc116378494)

# QUESTIONS FROM THE COURSES

# Questions from the course on Linked Data.

## Q1.1 Practice XML replace missing parts

<archi\_book>

<short\_title>Architecture Now</short\_title>

<main\_author>Jodidio, Philip<main\_author>

<ID isbn10="3822840912"/>

< archi\_book >

## Q1.2 Provide 10 first lines

Get 10 first lines of the five results for:

<http://www.wikidata.org/entity/Q23014205>  
<http://www.wikidata.org/entity/Q23014205.json>  
<http://www.wikidata.org/entity/Q23014205.rdf>  
<http://www.wikidata.org/entity/Q23014205.ttl>  
<http://www.wikidata.org/entity/Q23014205.nt>

<ANSWER HERE/>

## Q1.3 DBpedia

1. Find “London” on DBpedia.org; e.g. Google: "london site:dbpedia.org“  
   make sure you are on the English chapter (dbpedia.org) as there are many others (fr.dbpedia.org, de.dbpedia.org)
2. Find dbp:populationDemonym and give its value
3. Find rdf:type
4. Click on value yago:WikicatCapitalsInEurope
5. Find “Vienna” and get its URI  
   (careful: with content negotiation and redirection, the URL of the page you are currently viewing may be different from the URI of the resource it describes)
6. ISO code of the Vienna region?

Londoner, <http://dbpedia.org/resource/Vienna>, AT-9

## Q1.4 WHO.IS?

1. contact for inria.fr
2. contact for fabien.info
3. contact for lemonde.fr

Florian DUFOUR, REDACTED FOR PRIVACY, SOCIETE EDITRICE du monde

## Q1.5 CURL (or WGET)

1. Ten first lines:

curl -o Paris.html -L -H "Accept: text/html" http://dbpedia.org/resource/Paris

curl -o Paris-rdf.xml -L -H "Accept: application/rdf+xml" http://dbpedia.org/resource/Paris

1. Ten first lines for HTML and RDF http://ns.inria.fr/fabien.gandon#me
2. Ten first lines for HTML and RDF for ‘Vienna’ on Dbpedia
3. Ten first lines for the “URI of the name of Victor Hugo” in the Library of Congress:  
   http://id.loc.gov/authorities/names/n79091479
4. Ten first lines for HTML and RDF  
   https://purl.uniprot.org/uniprot/P43121
5. What is the topic and format of data obtained with  
   curl -o data.json -L -H "Accept: application/json" https://www.wikidata.org/wiki/Special:EntityData/Q551861
6. What is the topic and format of data obtained with  
   curl -o data.ttl -L -H "Accept: text/turtle" http://dx.doi.org/10.1007/3-540-45741-0\_18
7. curl -o Paris.html -L -H "Accept: text/html" http://dbpedia.org/resource/Paris

<!DOCTYPE html>

<html

prefix="

dbp: http://dbpedia.org/property/

dbo: http://dbedia.org/ontology/

dct: http://purl.org/dc/terms/

dbd: http://dbpedia.org/datatype/

og: https://ogp.me/ns#

"

## Q1.6 Find the URIs of «  Pedro Almodóvar » on the Spanish Dbpedia and on Wikidata.

<ANSWER HERE/>

## Q1.7 Spotlight demo

Reproduce the demo:

1. Copy a text from Wikipedia (e.g. Muse Band page)
2. Find the DBpedia Spotlight service page
3. Paste the text and run the detection
4. Try with other texts and copy-paste one of the results you get.

<ANSWER HERE WITH THE COPY-PASTED ANOTATED TEXT OR SCREENSHOT/>

# Questions from the course on RDF.

## Q2.0 Fill the blanks

## “Jen is an engineer woman, 42-year old, married to Seb who is a man with whom she had two children: Anny who is a woman and Steffen who is a man". For each person we also explicitly specify the name.

## To fill the blanks we use the values: :Seb, :Steffen, voc:name, voc:hasChild, voc:age, voc:hasSpouse, rdf:type, voc:Engineer, voc:Man, "Jen", "Seb", "Anny", "Steffen" For each person we also explicitly specify the name

<ANSWER HERE BY REPLACING ALL THE QUESTION MARKS/>:

42

?

?

?

?

?

?

?

?

?

?

?

?

?

?

?

?

?

?

?

## Q2.1 What is missing to say that “doc.html as for authors Catherine and Fabien and is about Music and Piano” ?

<http://inria.fr/rr/doc.html> <http://inria.fr/schema#author>  
 <http://ns.inria.fr/fabien.gandon#me> .  
  
<http://inria.fr/rr/doc.html> <http://inria.fr/schema#theme> "Music" .

<ANSWER HERE/>

## Q2.2 Fill the blanks (N3/Turtle)

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

@prefix voc: <http://www.unice.fr/voc#> .

@prefix xml: <http://www.w3.org/XML/1998/namespace> .

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

<http://www.unice.fr/data#Jen> a voc:Engineer , voc:Woman;

voc:age "42"^^xsd:string ;

voc:hasChild <http://www.unice.fr/data#Anny>, < http://www.unice.fr/data#Steffen >;

voc:hasSpouse <http://www.unice.fr/data#Seb> ;

voc:name "Jen" .

<http://www.unice.fr/data#Seb> a voc:Man ;

voc:hasChild <http://www.unice.fr/data#Anny>,

<http://www.unice.fr/data#Steffen> ;

voc:name "Seb" .

<http://www.unice.fr/data#Anny> a voc:Woman ;

voc:name "Anny" .

<http://www.unice.fr/data#Steffen > a voc:Man ;

voc:name "Steffen" .

## Q2.3 Fill the blanks (RDF/XML)

<?xml version="1.0" encoding="UTF-8"?>

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE rdf:RDF [ <!ENTITY vocab "http://www.unice.fr/voc"> <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#"> ]>

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:voc="&vocab;#" xml:base="http://www.unice.fr/data">

<voc:Woman rdf:about="#Jen">

<voc:name>Jen</voc:name>

<voc:age rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">AAA </voc:age>

<BBB rdf:resource="#Seb"></BBB>

<voc:hasChild rdf:resource="#Steffen"></voc:hasChild>

<voc:hasChild>

<rdf:Description rdf:about="#Anny">

<voc:name>Anny</voc:name>

<rdf:type CCC="&vocab;#Woman"></rdf:type>

</rdf:Description>

</voc:hasChild>

<DDD rdf:resource="&vocab;#Engineer"></DDD>

</voc:Woman>

<EEE rdf:about="#Seb">

<voc:name>Seb</voc:name>

<voc:hasChild rdf:resource="#Steffen"></voc:hasChild>

<voc:hasChild rdf:resource="#Anny"></voc:hasChild>

</EEE>

<voc:Man rdf:about="#Steffen">

<voc:name>Steffen</voc:name>

</voc:Man>

</rdf:RDF>

## Q2.4 Visit me please Get the RDF data from: [http://ns.inria.fr/fabien.gandon#me](http://ns.inria.fr/fabien.gandon)

1. Get the RDF data from: [http://ns.inria.fr/fabien.gandon#me](http://ns.inria.fr/fabien.gandon)
2. What is the syntax used?
3. Validate it and see the graph:  
   <http://www.w3.org/RDF/Validator/>
4. Translate into Turtle/N3:  
   <http://www.easyrdf.org/converter>   
   <http://rdf.greggkellogg.net/distiller>  
   <https://issemantic.net/rdf-converter>   
   <http://rdf-translator.appspot.com/>
5. Visualize it also with:  
   <http://cltl.nl/visualrdf/>   
   <http://www.easyrdf.org/converter> (PNG, SVG)  
   <https://www.ldf.fi/service/rdf-grapher>
6. Adapt to your data and do it again

<ANSWER HERE/>

## Q2.5 what is the meaning of this RDF? What is this description saying? <?xml version="1.0"?>

## <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:exs="http://example.org/schema#">

## <rdf:Description rdf:about="http://example.org/doc.html">

## <rdf:type rdf:resource="http://example.org/schema#Report"/>

## <exs:theme rdf:resource="http://example.org#Music"/>

## <exs:theme rdf:resource="http://example.org#Danse"/>

## <exs:nbPages rdf:datatype="http://www.w3.org/2001/XMLSchema#int">73</exs:nbPages>

## </rdf:Description>

## </rdf:RDF>

<ANSWER HERE/>

## Q2.6 Visit to Victor Hugo

1. See HTML data from:  
   <http://id.loc.gov/authorities/names/n79091479.html>
2. Get RDF data from:  
   <http://id.loc.gov/authorities/names/n79091479.rdf>
3. What is the syntax?
4. Translate into Turtle/N3:  
   <http://www.easyrdf.org/converter>   
   <http://rdf.greggkellogg.net/distiller>  
   <https://issemantic.net/rdf-converter>   
   <http://rdfvalidator.mybluemix.net/>   
   <http://rdf-translator.appspot.com/>
5. Any remark about the values of the properties of Victor Hugo?

<ANSWER HERE/>

## Q2.7 What is the syntax of the following RDF statement? What does it mean? @prefix dcterms: <http://purl.org/dc/terms/>.

## GRAPH <http://inria.fr/data/algebra>

## {

## <http://inria.fr/rr/doc.html>

## dcterms:subject

## <http://data.bnf.fr/ark:/12148/cb121105993> .

## }

<ANSWER HERE/>

## Q2.8 Visit Leukocyte surface antigen CD53

1. See HTML data from:  
   <http://www.uniprot.org/uniprot/Q61451>
2. Get RDF data from:  
   <http://www.uniprot.org/uniprot/Q61451.rdf>
3. What is the syntax?
4. Translate into Turtle/N3:  
   <http://www.easyrdf.org/converter>   
   <http://rdf.greggkellogg.net/distiller>   
   <https://issemantic.net/rdf-converter>   
   <http://rdfvalidator.mybluemix.net/>   
   <http://rdf-translator.appspot.com/>
5. Any remark about the structure of the data?

<ANSWER HERE/>

# Questions from the course on SPARQL.

## Q3.1 Test SPARQL online Connect to: <https://corese.inria.fr/srv/tutorial/sparql>

## Answers to the query:

## prefix v: <http://www.inria.fr/2015/humans#>

## select \* where { ?x a v:Person . }

<ANSWER HERE/>

## Q3.2 Test SPARQL online Connect to

<http://dbpedia.org/snorql/> or  
<http://fr.dbpedia.org/sparql> or …  
<http://wiki.dbpedia.org/Internationalization/Chapters>

## Answers to the query:

## SELECT ?x ?p ?v WHERE {

## ?x rdfs:label "Paris"@fr .

## ?x ?p ?v .

## }

## LIMIT 10

<ANSWER HERE/>

## Q3.3 Test SPARQL online Connect to: <https://query.wikidata.org/>

## What does this query retrieve?

## SELECT distinct ?p ?n WHERE

## { wd:Q30 p:P6 [ ps:P6 ?p ].

## ?p rdfs:label ?n .

## FILTER (lang(?n)="en") }

## Discover wd:Q30 using the namespace attached to wd: PREFIX wd: <http://www.wikidata.org/entity/>

## Discover p:P6 using the namespace attached to p: PREFIX p: <http://www.wikidata.org/prop/>

## PREFIX ps: <http://www.wikidata.org/prop/statement/> Find q-name of the property “given name” <https://www.wikidata.org/wiki/Wikidata:List_of_properties>

<ANSWER HERE/>

## Q3.4 SPARQL query to return 20 persons at most (use type foaf:Person)

<ANSWER HERE/>

## Q3.5 SPARQL query to return 20 persons (at most), after the 10th result i.e. from 11th to 30th

<ANSWER HERE/>

## Q3.6 You have two properties: c:name and c:age 1.Find the age of resources whose name is ‘Fabien’

<ANSWER HERE/>

## 2.Find the name of resources whose age is less than 50

<ANSWER HERE/>

## 3. Find properties and their values for resources whose name is ‘Fabien’ and whose age is less than 50

<ANSWER HERE/>

## 4.Find other names of resources with name ‘Fabien’, at least, a second name

<ANSWER HERE/>

## 5.Find resources which have two different properties with the same value

<ANSWER HERE/>

## 6.Find resources which have the same property with two different values

<ANSWER HERE/>

## Q3.7 Could this query return ex:a c:memberOf ex:b and why ?

## select \* where {

## ?x c:memberOf ?org . minus { ex:a c:memberOf ex:b }

## }

<ANSWER HERE/>

## Q3.8 get the members of organizations (c:memberOf) but remove the resources author of a document (c:author) by using ‘not exists’

<ANSWER HERE/>

## Q3.9 what is retrieving this query ?

## prefix ex: <http://example.org/>

## select ?x (count(?doc) as ?c) where { ?x ex:author ?doc }

## group by ?x

## order by desc(count(?doc))

<ANSWER HERE/>

## Q3.10 What expression should we use to find the ?x related to ?y by paths composed of properties foaf:knows and/or rdfs: seeAlso?

## ?x (foaf:knows | rdfs:seeAlso)+ ?y

## ?x foaf:knows+ | rdfs:seeAlso+ ?y

## ?x (foaf:knows / rdfs:seeAlso)+ ?y

<ANSWER HERE/>

## Q3.11 what is this query retrieving?

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

## select ?x (if (bound(?n), ?n, "John Doe") as ?m)

## where {

## ?x foaf:knows ?y

## optional { ?y foaf:name ?n }

## }

<ANSWER HERE/>

## Q3.12 what is this query retrieving?

## prefix ex: <http://example.org/>

## select ?x (avg(?a) as ?b)

## where {

## ?x ex:knows ?y .

## ?y ex:age ?a

## }

## group by ?x

<ANSWER HERE/>

## Q3.13 You have two properties: c:name and c:study and the resources c:Informatics and c:Mathematics

## 1. Find resources that study informatics or mathematics

## 2. In addition return the name of the resource if it has a name

## 3. In addition return the graph where the name is given

<ANSWER HERE/>

## Q3.14 On which graph(s) is calculated ?x ?p ?y

## On which graph(s) is calculated graph ?g { ?y ?q ?z }

## prefix ex: <http://example.org/>

## select \*

## from ex:g1

## from named ex:g2

## where {

## ?x ?p ?y .

## graph ?g { ?y ?q ?z} }

<ANSWER HERE/>

## Q3.15 Write a query to change foaf:name into rdfs:label

<ANSWER HERE/>

## Q3.16 what is this query performing?

## prefix ex: <http://example.org/>

## delete { ?x ex:age ?a }

## insert { ?x ex:age ?i }

## where {

## select ?x ?a (xsd:integer(?a) as ?i)

## where {

## ?x ex:age ?a

## filter(datatype(?a) = xsd:string)

## }

## }

<ANSWER HERE/>

## Q3.17 Which clauses could you use to obtained results as RDF triples following a specific pattern?

## SELECT … WHERE {…} …

## CONSTRUCT { } WHERE {…} …

## DESCRIBE <…> DESCRIBE … {…}

## ASK {…}

## DELETE { … } INSERT { … } WHERE {…} …

<ANSWER HERE/>

## Q3.18 What is the difference between these two queries?

## prefix ex: <http://example.org/>

## insert { ?x a ex:Parent }

## where { ?x ex:hasChild ?y }

## prefix ex: <http://example.org/>

## construct { ?x a ex:Parent }

## where { ?x ex:hasChild ?y }

<ANSWER HERE/>

# Questions from the course on Ontologies.

## Q4.0 Choose among the following assertions one or more you consider to be true:

## an ontology is necessarily formalized in first-order logic

## an ontology may allow inferences on data that uses it

## conceptual graphs can represent an ontology

## a shared ontology promotes interoperability

## description logics can represent an ontology

<ANSWER HERE/>

## Q4.1 work alone for 10 minutes

## From real example of ontology engineering: you are designing a system to assist (data) management of species presented in a museum of natural sciences. In particular you need to organize at least the species below. Structure them and add categories as needed.

## Species: animal, bird, cat, cicada, clam, crocodile, dog, dragonfly, fish, frog, fungus, insect, kiwi, octopus, ostrich, shark, snail, snake, spider, thing, trout, whale

<ANSWER HERE/>

# Questions from the course on RDFS

## Q4.2 RDFS contains primitives to (several answers possible)…

## describe classes of resources

## describe formulas of calculation for values of properties

## describe types of properties of resources

## document definitions in natural language

## sign and authenticate the authors of the definitions of classes and properties

<ANSWER HERE/>

## Q4.3. What is defined and derived from these definitions?

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

## @prefix : <http://inria.fr/devices#>

## :Phone rdfs:subClassOf :Device .

## :Computer rdfs:subClassOf :Device .

## :Smartphone rdfs:subClassOf :Computer .

## :Smartphone rdfs:subClassOf :Phone .

<ANSWER HERE/>

## Q4.4. What is defined and derived from these definitions?

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

## @prefix : <http://inria.fr/member#>

## :employeeOf rdfs:subPropertyOf :proRelationWith .

## :hasControlOver rdfs:subPropertyOf :proRelationWith . :isShareholderOf rdfs:subPropertyOf :hasControlOver .

## :isCEOof rdfs:subPropertyOf :employeeOf, :hasControlOver .

<ANSWER HERE/>

## Q4.5. Download the ontology Schema.org founded by Google, Microsoft, Yahoo and Yandex: <https://schema.org/version/latest/schemaorg-current-https.ttl>

## Find the Class schema:AboutPage and identify its super-class

## Can a document be of type schema:abstract ?

<ANSWER HERE/>

## Q4.6. What can be said about the types of the resources that will be linked by the properties defined below?

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

## @prefix : <http://inria.fr/humans#>

## :driverOf rdfs:subPropertyOf :isControling .

## :piloteOf rdfs:subPropertyOf :isControling .

## :isControling rdfs:domain :Human ; rdfs:range :Object .

## :driverOf rdfs:range :Car .

## :piloteOf rdfs:domain :Adult ; rdfs:range :Plane .

<ANSWER HERE/>

## Q4.7. What could we add to this schema (several answers are possible)?

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

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

## @ prefix : <http://ns.inria.fr/humans/schema#>

## :p1 a rdf:Property ; rdfs:label "age"@en .

## :c1 a rdfs:Class ; rdfs:comment "a human being"@en .

## :p1 rdfs:label "firstname"@en, "prénom"@fr .

## :c1 rdfs:comment "un être humain"@en .

## :c1 rdfs:label "person"@en, "personne"@fr .

## :p1 rdfs:label "âge"@fr .

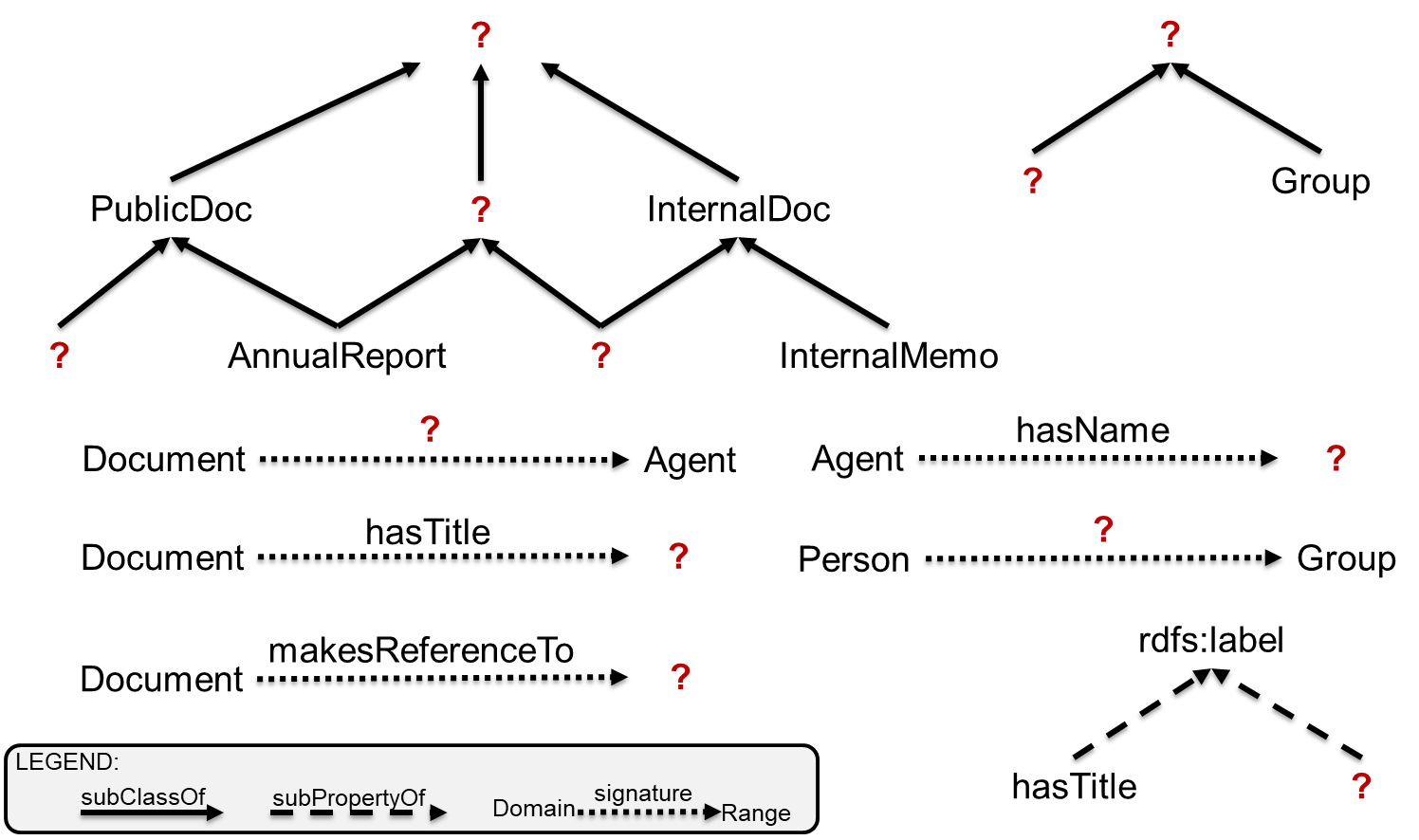
## :c1 rdfs:label "woman"@en .

## :c1 rdfs:label "persona"@es .

## :p1 rdfs:comment "the length of time something has existed."@en .

<ANSWER HERE/>

## Q4.8. (a) Fill the blanks with: Document, PublicDoc, PressArticle, Report, AnnualReport, InternalDoc, SecretReport, InternalMemo, Agent, Person, Group, hasTitle, hasAuthor, makesReferenceTo, hasName, isMemberOf + rdf / rdfs primitives. (b) Write it in RDFS and validate the RDF.



# Questions from the course on OWL.

## Q5.1 What is asserted and what can we deduce?

## ex:Man owl:intersectionOf (ex:Male ex:Human) . ex:Woman owl:intersectionOf (ex:Female ex:Human) . ex:Human owl:unionOf (ex:Man ex:Woman) .

## ex:Jane a ex:Human . ex:John a ex:Man .

## ex:James a ex:Male .

## ex:Jane a ex:Female .

<ANSWER HERE/>

## Q5.2 What are we defining and inferring?

## @prefix ex: <http://example.org/>

## ex:GrandFather rdfs:subClassOf [

## a owl:Class ;

## owl:intersectionOf ( ex:Parent ex:Man )

## ] .

## ex:Jim a ex:Man, ex:Parent .

## ex:Jack a ex:GrandFather .

<ANSWER HERE/>

## Q5.3 What is asserted and what can we deduce?

## ex:hasSpouse a owl:SymmetricProperty . ex:hasChild owl:inverseOf ex:hasParent . ex:hasParent rdfs:subPropertyOf ex:hasAncestor . ex:hasAncestor a owl:TransitiveProperty . ex:Jim ex:hasChild ex:Jane . ex:Jane ex:hasSpouse ex:John . ex:Jim ex:hasParent ex:James .

<ANSWER HERE/>

## Q5.4 What is asserted and what can we deduce?

## ex:Human owl:equivalentClass foaf:Person . foaf:name owl:equivalentProperty ex:name . ex:JimmyPage a ex:Human ; owl:sameAs ex:JamesPatrickPage . ex:JimmyHendrix owl:differentFrom ex:JimmyPage .

<ANSWER HERE/>

## Q5.5 What are we defining and inferring?

## ex:UnhappyPerson owl:equivalentClass [

## a owl:Class ;

## owl:intersectionOf (

## ex:Person

## [ a owl:Class ; owl:complementOf ex:Happy ]

## )

## ] .

<ANSWER HERE/>

## Q5.6 What is asserted and what can we deduce?

## ex:Human rdfs:subClassOf [ a owl:Restriction ;

## owl:onProperty ex:hasParent ;

## owl:allValuesFrom ex:Human ] .

## ex:Tom a ex:Human . ex:Tom ex:hasParent ex:James, ex:Jane.

<ANSWER HERE/>

## Q5.7 What are we defining and inferring?

## @prefix ex: <http://example.org/>

## ex:PersonList rdfs:subClassOf

## [

## a owl:Restriction ;

## owl:onProperty rdf:first ;

## owl:allValuesFrom ex:Person

## ] , [

## a owl:Restriction ;

## owl:onProperty rdf:rest ;

## owl:allValuesFrom ex:PersonList

## ] .

## ex:value rdfs:range ex:PersonList .

## ex:abc ex:value (ex:a ex:b ex:c) .

<ANSWER HERE/>

## Q5.8 What are we defining and inferring?

## @prefix ex: <http://example.org/>

## ex:Human rdfs:subClassOf [

## owl:intersectionOf (

## [

## a owl:Restriction ;

## owl:onProperty ex:hasBiologicalFather ;

## owl:maxCardinality 1

## ] , [

## a owl:Restriction ;

## owl:onProperty ex:hasBiologicalMother ;

## owl:maxCardinality 1

## ] )

## ] .

## ex:Jane a ex:Human ; ex:hasBiologicalFather ex:James , ex:Jhon .

<ANSWER HERE/>

## Q5.9 Visit the financial Industry Business Ontology (FIBO)

## <https://spec.edmcouncil.org/fibo/>

## with a lot of companies using it: <https://edmcouncil.org/page/listofmembersreview>

## Negociate the turtle version of the part about governments: <https://spec.edmcouncil.org/fibo/ontology/BE/GovernmentEntities/GovernmentEntities/>

## Explain the formal definitions of Government, NationalGovernment, FederalGovernment, and isJurisdictionOf.

<ANSWER HERE/>

# Questions from the course on Vocabularies.

## Q6.0 Visit schema.org and

## Check the documentation of schema:Accommodation <https://schema.org/Accommodation> and identify the more specific types in the page

## Download the current version in Turtle (.ttl): <https://schema.org/docs/developers.html> extract the Turtle definition of schema:Recipe and find the name of the class it inherits from

<ANSWER HERE/>

## Q6.1 What do you think of the annotation?

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

## <#B-A-Ba> a skos:Concept ;

## skos:prefLabel "B.A.-BA"@en , "b.a.-ba"@en ;

## skos:altLabel "B-A-BA"@en , "b-a-ba"@en ;

## skos:hiddenLabel "BABA"@en , "baba"@en .

<ANSWER HERE/>

## Q6.2 practice:

## Using the site prefix.cc find back the namespace usually associated to the SKOS prefix

## Access the URL of the namespace and find the RDF source file defining the SKOS vocabulary

## Find the definition of the property narrowMatch and give all the relations it has with other properties

<ANSWER HERE/>

## Q6.3 practice:

## Find and open the source file of Dublin Core Terms: <https://dublincore.org/schemas/rdfs/> Look at the definition of the class FileFormat and find the class it inherits from.

## Choose your preferred book on Amazon, Fnac, etc. and describe it in an RDF annotation using as many DC primitives as necessary.

## Add the most restrictive CC license to your preferred book ; is this license appropriate?

<ANSWER HERE/>

## Q6.4 practice:

## Get the source of the FoaF schema: <http://xmlns.com/foaf/spec/index.rdf>

## Find the property weblog

## What are the types of this property?

## Does it inherit from other properties?

## What is its signature?

<ANSWER HERE/>

## Q6.5 practice:

## Find the FOAF-a-Matic web page

## Use this tool to generate your FOAF profile in RDF/XML

## Translate it into Turtle, save and give the result in your answers.

## Add five specific relationships to your FOAF file using RELATIONSHIPS: <http://purl.org/vocab/relationship/>

<ANSWER HERE/>

## Q6.6 What does this mean?

## :BioRDF2DBLP a void:Linkset;

## void:target :BioRDF;

## void:target :DBLP;

## void:linkPredicate skos:exactMatch;

## void:triples 8936 .

<ANSWER HERE/>

## Q6.7 practice:

## Connect to the Void Store SPARQL endpoint: <http://lod.openlinksw.com/sparql/>

## Find the void:Dataset with a property dct:subject and retrieve their URI and subjects.

## Find the void:Dataset with a dct:subject "covid19".

<ANSWER HERE/>

## Q6.8 What does this mean?

## ex:plot prov:used ex:stats1998 .

## ex:bar-chart prov:wasGeneratedBy ex:plot .

## ex:stats1998 a dcat:Distribution ;

## dcat:format [ rdfs:label "CSV" ] ;

## dcat:mediaType "text/csv" .

<ANSWER HERE/>

## Q6.9 What does this mean?

## @prefix dcat: <http://www.w3.org/ns/dcat#> .

## @prefix void: <http://rdfs.org/ns/void#> .

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

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

## @prefix prov: <http://www.w3.org/ns/prov#> .

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

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

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

## @prefix : <http://inria.fr/data#> .

## :db-employ

## a dcat:Distribution ;

## dcat:downloadURL <http://wimmics.inria.fr/docs/employ-2014.sql> ;

## dct:title "SQL Dump of the employees" ;

## dct:spatial <http://www.geonames.org/6640252> ;

## dct:issued "2015-01-12"^^xsd:date ;

## dct:temporal <http://reference.data.gov.uk/id/year/2014> ;

## dct:publisher <http://inria.fr> ;

## dcat:mediaType "application/sql" ;

## dcat:format [ rdfs:label "SQL" ] ;

## dct:language <http://id.loc.gov/vocabulary/iso639-1/fr> ;

## dcat:byteSize "38729"^^xsd:decimal .

## 

## :R2RTransform12 prov:used :db-employ ;

## prov:used :R2R-employ-mapping ;

## prov:used <http://xmlns.com/foaf/0.1/> .

## :FoaFDump a void:Dataset;

## void:feature <http://www.w3.org/ns/formats/RDF\_XML>;

## void:dataDump <http://wimmics.inria.fr/docs/employ-2014.rdf>;

## void:exampleResource <http://ns.inria.fr/fabien.gandon#me> ;

## void:vocabulary <http://xmlns.com/foaf/0.1/>;

## void:triples 12875;

## dct:title "RDF Dump of the employees" ;

## prov:wasGeneratedBy :R2RTransform12 ;

## prov:generatedAtTime "2015-01-14T11:38:27"^^xsd:dateTime ;

## prov:wasDerivedFrom :db-employ .

<ANSWER HERE/>

## Q6.10 practice:

## Here is the namespace of Ontology for Media Resources: [http://www.w3.org/ns/ma-ont#](http://www.w3.org/ns/ma-ont) Negotiate the turtle version of the ontology with curl/wget

## Study the ontology to know if I can use the property hasLanguage on a Person

<ANSWER HERE/>

## Q6.11 practice:

## Connect to the LOV directory: <https://lov.linkeddata.es/>

## Search for schemas talking about “music artist”.

## What is the top ontology you find?

## What is its version number?

## Is it reused by other ontologies?

## How many classes and properties does it have?

## What expressivity does it use? (RDFS, OWL)

<ANSWER HERE/>

# Questions from the course on other data formats.

## Q7.1 What are the triples produced with this mapping and this table?

## :My\_Table rdf:type rr:TriplesMap ;

## rr:subjectMap [ rr:template "https://www.ietf.org/rfc/rfc{NUM}.txt"; ];

## rr:predicateObjectMap [

## rr:predicateMap [ rr:predicate dc:title ];

## rr:objectMap [ rr:column "ttl" ]

## ].

|  |  |  |
| --- | --- | --- |
| **ID** | **NUM** | **ttl** |
| 87 | 2616 | Hypertext Transfer Protocol -- HTTP/1.1 |
| 88 | 2396 | Uniform Resource Identifiers (URI): Generic Syntax |

<ANSWER HERE/>

## Q7.2 What are the triples encoded in this HTML?

## <div vocab="http://xmlns.com/foaf/0.1/" resource="#cathy" typeof="Person">

## <p> <span property="name">Catherine Faron</span>

## (mail: <span property="mbox">faron@i3s.unice.fr</span>) is a friend of

## <span property="knows" resource="http://ns.inria.fr/fabien.gandon#me">Fabien Gandon</span>

## </p>

## </div>

<ANSWER HERE/>

## Q7.3 practice:

## Look at the Web Page <https://www.w3.org/TR/xhtml-rdfa-scenarios/scenario-2.html>

## Call the translator on this Web page to get Turtle: <https://www.easyrdf.org/converter> <http://rdf.greggkellogg.net/distiller> <https://issemantic.net/rdf-converter>

## What does the extracted triple say?

## Repeat the question with the page: <https://coffeecode.net/rdfa/codelab/exercises/check_3b.html>

## Test the page: <http://ns.inria.fr/humans/humans_rdfa.html>

<ANSWER HERE/>

## Q7.4 Use the online tool to play with RDFa adding for instance a “creator” property <https://rdfa.info/play/>

<ANSWER HERE/>

## Q7.5 IMDB uses RDFa – OGP for the I like button

## Choose a movie on IMDB http://www.imdb.com

## Copy the URL of the page of the movie

## Go to the RDFa 1.0 RDFa Distiller and Parser: <https://www.w3.org/2007/08/pyRdfa/>

## Open the URI option, past the URL of the movie page and configure and perform the extraction to get Turtle

## Try also the transformation on a translator: <http://www.easyrdf.org/converter> <http://rdf.greggkellogg.net/distiller> <https://issemantic.net/rdf-converter> <http://rdfvalidator.mybluemix.net/> <http://rdf-translator.appspot.com/>

<ANSWER HERE/>

## Q7.6 Test JSON-LD online

## Transform your FOAF profile in JSON-LD with the translator: <http://www.easyrdf.org/converter> <http://rdf.greggkellogg.net/distiller> <https://issemantic.net/rdf-converter> <http://rdfvalidator.mybluemix.net/> <http://rdf-translator.appspot.com/>

## Use the following online tool to generate different variations of JSON-LD of your profile (expanded, collapsed, flattened, etc.) <http://json-ld.org/playground/>

<ANSWER HERE/>

## Q7.7 To provide the metadata of a CSV file I can...

## include them in a special column of the CSV.

## put them in a file with the same name plus “-metadata.json”.

## put them in the first line of my CSV file.

## put them in a file called “csv-metadata.json” in the same directory.

## add the URL of the metadata file to the content of my CSV file.

<ANSWER HERE/>

## Q7.8 TV Catalog : Imagine we submit the following call to an LDP platform

## GET /catalog/tv/ HTTP/1.1

## Host: example.org

## Accept: text/turtle; charset=UTF-8

## and we receive the following answer:

## HTTP/1.1 200 OK

## Content-Type: text/turtle; charset=UTF-8

## Link: <http://www.w3.org/ns/ldp#Resource>; rel="type", <http://www.w3.org/ns/ldp#DirectContainer>; rel="type"

## Allow: OPTIONS,HEAD,GET,POST,PUT

## Accept-Post: text/turtle, application/ld+json

## Content-Length: 232

## ETag: W/"90231678"

## @prefix ldp: <http://www.w3.org/ns/ldp#> .

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

## @prefix cat: <http://example.org/vocab/catalog#> .

## <> a ldp:DirectContainer; ldp:membershipResource <#cat>; ldp:hasMemberRelation cat:hasProduct;

## dcterms:title "Container of the TV descriptions";

## ldp:contains <tv1>, <tv2> .

## <#cat> a cat:Catalog; dcterms:title "Catalog of TVs"; cat:hasProduct <tv1>, <tv2> .

## Which ones of the following statements are true?

## the container is just a basic container.

## the container is a direct container.

## the container is an indirect container.

## the platform accepts the GET calls.

## the platform accepts the PATCH calls.

## the platform accepts RDF/XML format.

## the platform accepts RDF Turtle.

## the platform accepts RDF JSON-LD.

## a link hasProduct is automatically created between the resource #cat and the resources of this container

<ANSWER HERE/>

# LAB SESSIONS

**Remember: just like for programming, it is a good practice to write & validate step by step, incrementally, and to start from copy-pasted examples from the course.**

# Lab session on RDF.

## Software requirements

* A real text editor (e.g. Notepad++, Gedit, Sublime Text, Emacs, etc.) do not use Word!
* The RDF XML online validation service by W3C: <https://www.w3.org/RDF/Validator/>
* An RDF online translator:

* + <http://www.easyrdf.org/converter>
  + <http://rdf.greggkellogg.net/distiller>
  + <https://issemantic.net/rdf-converter>
  + <http://rdfvalidator.mybluemix.net/>
  + <http://rdf-translator.appspot.com/>
* The SPARQL Corese engine (CORESE-GUI) : <https://project.inria.fr/corese/>

## Understand existing data

1, If you haven’t do it yet during the course, get the RDF/XML about <http://ns.inria.fr/fabien.gandon#me> and translate the RDF/XML into Turtle/N3 syntax using one of the online translators.

Code of validated RDF in N3 syntax:

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

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

<http://ns.inria.fr/fabien.gandon>

a foaf:PersonalProfileDocument ;

foaf:maker <http://ns.inria.fr/fabien.gandon#me> ;

foaf:primaryTopic <http://ns.inria.fr/fabien.gandon#me> .

<http://ns.inria.fr/fabien.gandon#me>

a foaf:Person ;

foaf:name "Fabien Gandon" ;

foaf:title "Dr" ;

foaf:givenname "Fabien" ;

foaf:family\_name "Gandon" ;

foaf:nick "Bafien" ;

foaf:mbox <mailto:fabien.gandon@inria.fr> ;

foaf:homepage <http://fabien.info> ;

foaf:depiction <http://www-sop.inria.fr/members/Fabien.Gandon/common/FabienGandonBackground.jpg> ;

foaf:phone <tel:0492387788> ;

foaf:workplaceHomepage <http://www.inria.fr/> ;

foaf:workInfoHomepage <http://fabien.info> ;

foaf:schoolHomepage <http://www.insa-rouen.fr> ;

foaf:knows [

a foaf:Person ;

foaf:name "Olivier Corby" ;

foaf:mbox <mailto:olivier.corby@inria.fr> ;

rdfs:seeAlso <http://www-sop.inria.fr/members/Olivier.Corby/>

], [

a foaf:Person ;

foaf:name "Catherine Faron-Zucker" ;

foaf:mbox <mailto:faron@polytech.unice.fr> ;

rdfs:seeAlso <http://www.i3s.unice.fr/~faron/>

] .

In the RDF Turtle file, can you identify all the links between the two resources <http://ns.inria.fr/fabien.gandon> and <http://ns.inria.fr/fabien.gandon#me> ? What do they represent?

maker & primaryTopic

It’s a profile document who talks about Fabien and was made by Fabien. The resource <http://ns.inria.fr/fabien.gandon> is the profile document and the resource <http://ns.inria.fr/fabien.gandon#me> is Fabien

2, Get the Turtle data of Paris on DBpedia.org then in the file find the triple that declares it as a capital in Europe.

curl -o Paris.ttl -L -H "Accept: text/turtle" <http://dbpedia.org/resource/Paris>

The triple is:

@prefix dbr: <http://dbpedia.org/resource/> .

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

@prefix dbc: <http://dbpedia.org/resource/Category:> .

dbr:Paris dcterms:subject dbc:Capitals\_in\_Europe

## Humans Knowledge Graph and its namespace [http://ns.inria.fr/humans/data#](http://ns.inria.fr/humans/data)

The major part of this practical session is using a small dataset about a few persons.

The namespace of the dataset is [http://ns.inria.fr/humans/data#](http://ns.inria.fr/humans/data)

1. With your Web browser, visit that namespace and spot the age of Gaston in the graph

<ANSWER HERE/>

1. Use the command curl or wget to obtain the **XML version** of this dataset from the same address and download it in a file named “**humans\_data.xml**” then use the W3C RDF online validation service to validate the RDF/XML and see the triples and the graph.
2. Use the command curl or wget to obtain the **Turtle version** of this dataset from the same address and download it in a file named “humans\_data.ttl”
3. What is the namespace used for instances created / resources described in this file?

<ANSWER HERE/>

1. In the file how is the association between resources described (Gaston, Laura, etc.) and their namespace done i.e. how is the namespace of these resources specified?

<ANSWER HERE/>

1. What is the namespace of the vocabulary used to describe the resources (hasParent, name, etc.) in the dataset and how is it specified?

<ANSWER HERE/>

1. Find *everything* about John in the turtle file, all available information:

<ANSWER HERE/>

## Create RDF

Here is a statement extracted from the course:

“Jen is an engineer woman, 42-year old, married to Seb who is a man with whom she had two children: Anny who is a woman and Steffen who is a man”.

1. Use your text editor and write the above statements in RDF in N3 syntax inventing your own vocabulary. Save you file as “Jen.ttl”
2. Use your favorite text or XML editor and write the above statements in RDF in XML syntax reusing the same vocabulary “Jen.rdf”
3. Use the RDF XML online validation service to validate your XML and see the triples <https://www.w3.org/RDF/Validator/>
4. In the validator use the option to visualize the graph
   * Use the RDF online translator to validate your N3 and translate it into RDF/XML:  
     <http://www.easyrdf.org/converter>
   * <http://rdf.greggkellogg.net/distiller>
   * <https://issemantic.net/rdf-converter>
   * <http://rdfvalidator.mybluemix.net/>
   * <http://rdf-translator.appspot.com/>
5. Compare your RDF/XML with the result of the N3 translation
6. Translate in other formats to see the results.

Code of validated RDF in N3 syntax:

<ANSWER HERE/>

Code of validated RDF in XML syntax:

<ANSWER HERE/>

## Query your data

Download the Corese.jar library and start it as a standalone application: Run the command " java -jar -Dfile.encoding=UTF8 " followed by the name of the “.jar” archive. Notice that you need java on your machine and proper path configuration.

This interface provides several tabs: (1) the System tab for traces of execution, (2) a SHACL editor tab (3) a Turtle Editor tab and (4) A “+” tab to create as many queries as you want. Load the annotations contained in the file “Jen.rdf” you created and validated before. Click on the “+” tab to create a new query and the interface contains a default SPARQL query:

select \* where { ?x ?p ?y}

The SPARQL language will be presented in the next course. Just know that this query can find all the triples of your data. Launch the query and check the results.

# Lab session on SHACL.

## Software requirements

* A real text editor (e.g. Notepad++, Gedit, Sublime Text, Emacs, etc.)
* The RDF XML online validation service by W3C: <https://www.w3.org/RDF/Validator/>
* The SPARQL Corese engine (Corese-GUI jar file): <https://project.inria.fr/corese/>
* The human dataset file [http://ns.inria.fr/humans/data#](http://ns.inria.fr/humans/data)
* The SHACL file <http://ns.inria.fr/humans/humans_shape.ttl>

## What is that shape

With you text editor open the file humans\_shape.ttl and look at the content

What is the qualified name of the main shape being defined:

<ANSWER HERE/>

What is the type of that shape:

<ANSWER HERE/>

What is the target of that shape:

<ANSWER HERE/>

Explain in English the constraint it places on the focus node:

<ANSWER HERE/>

What is the severity level of that constraint?

<ANSWER HERE/>

In Corese load the dataset humans\_data.ttl (menu “file > load > Dataset”) and this shape (menu “file > load > SHACL”) and run the validation in a query tab (button “SHACL” in a query tab). Explain in English what the report is saying:

<ANSWER HERE/>

## Add your constraints

Extend the shape to add a constraint of severity level “Warning” enforcing that a Person should have an age:

<ANSWER HERE/>

In Corese load the dataset humans\_data.ttl (menu “file > load > Dataset”) and this shape (menu “file > load > SHACL”) and run the validation in a query tab (button “SHACL” in a query tab). Explain in English what the report is saying:

<ANSWER HERE/>

Extend the shape to add a constraint of severity level “Info” enforcing that a person’s name should be in English:

<ANSWER HERE/>

In Corese load the dataset humans\_data.ttl (menu “file > load > Dataset”) and this shape (menu “file > load > SHACL”) and run the validation in a query tab (button “SHACL” in a query tab). Explain in English what the report is saying:

<ANSWER HERE/>

# Lab session on SPARQL.

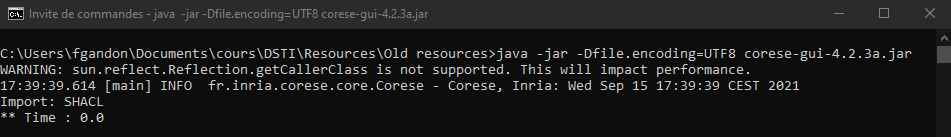
## Software requirements

* The RDF XML online validation service by W3C: <https://www.w3.org/RDF/Validator/>
* The RDF online translator: <http://rdf-translator.appspot.com/>
* The SPARQL Corese engine (Corese-GUI jar file): <https://project.inria.fr/corese/>

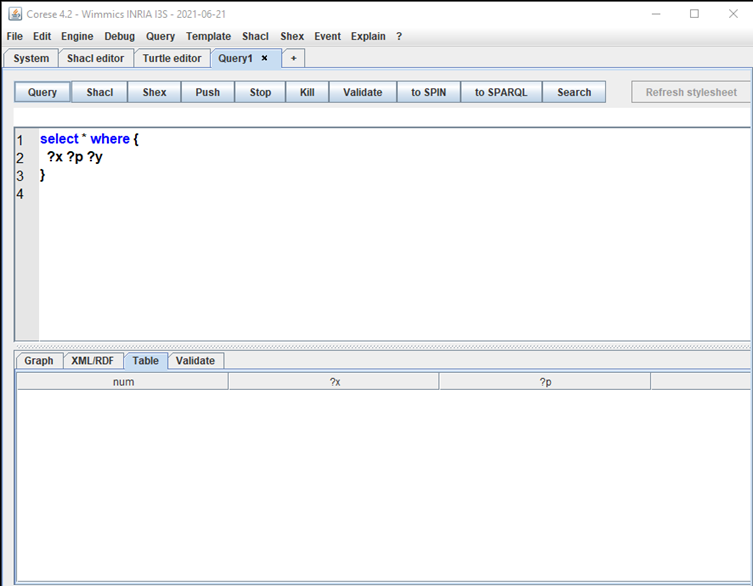
## Basic query on RDF humans dataset

If you haven’t done it yet download the SPARQL Corese engine.

Run the command " java -jar -Dfile.encoding=UTF8 " followed by the name of the “.jar” archive. Notice that you need java on your machine and proper path configuration. Example:



This interface provides several tabs: (1) the System tab for traces of execution, (2) a SHACL editor tab (3) a Turtle Editor tab and (4) A “+” tab to create as many queries as you want.



You should have the dataset humans\_data.ttl from the previous practical session.

Load the file humans\_data.ttl as RDF data in CORESE.

**NB:** CORESE reads all the formats/syntaxes of RDF.

## Question 1:

Create a new tab to enter the following query and explain what it does and the results you get. This is a good way to familiarize yourself with the data.

CONSTRUCT { ?s ?p ?o } WHERE { ?s ?p ?o }

Explanation:

<ANSWER HERE/>

Screenshot:

<ANSWER HERE/>

## Question 2:

Create a new tab to enter the following query:

prefix h: <http://ns.inria.fr/humans/schema#>

select \* where { ?x a ?t . filter(strstarts(?t, h:)) }

Translate this query in plain English.

<ANSWER HERE/>

Run this query. How many answers do you get?

<ANSWER HERE/>

Find John and his types in the answers.

John’s types:

<ANSWER HERE/>

## Question 3:

In the previous answer, locate the URI of John.

1. formulate a SELECT query to find all the properties of John, using his URI

Query

<ANSWER HERE/>

Results:

<ANSWER HERE/>

1. request a description of John using the SPARQL clause for this.

Query

<ANSWER HERE/>

Results:

<ANSWER HERE/>

## Question 4

Create a new tab to enter the following query:

prefix h: <http://ns.inria.fr/humans/schema#>

select \* where { ?x h:hasSpouse ?y }

Translate this query in plain English.

<ANSWER HERE/>

Run this query. How many answers do you get?

<ANSWER HERE/>

## Question 5:

In the RDF file, find the name of the property that is used to give the shoe size of a person.

1. Deduce a query to extract all the persons (h:Person) with their shoe size.

Query:

<ANSWER HERE/>

Result:

<ANSWER HERE/>

1. Change this query to retrieve all the persons and, if available, their shoe size.

Query:

<ANSWER HERE/>

Result:

<ANSWER HERE/>

1. Change this query to retrieve all the persons whose shoe size is greater than 8 or whose shirt size is greater than 12.

Query:

<ANSWER HERE/>

Result:

<ANSWER HERE/>

## Question 6:

In the RDF file, find the name of the property that is used to indicate the children of a person.

1. Formulate a query to find the parents who have at least one child.

Query:

<ANSWER HERE/>

How many answers do you get? How many duplicates do you identify in these responses?

<ANSWER HERE/>

1. Find a way to avoid duplicates.

Query:

<ANSWER HERE/>

How many answers do you get then?

<ANSWER HERE/>

1. Rewrite a query to find the Persons, Men and Women who have no child.

Query:

<ANSWER HERE/>

## Question 7

In the RDF file, find the name of the property that is used to give the age of a person.

1. Formulate a query to find persons with their age.

Query:

<ANSWER HERE/>

Result:

<ANSWER HERE/>

1. Formulate a query to find person who are not adults (here and Adult is a person at least 18 years old).

Query:

<ANSWER HERE/>

How many answers do you get?

<ANSWER HERE/>

1. Use the appropriate query clause to check if Mark is an adult; use the proper clause statement for this type of query to get a true or false answer.

Query:

<ANSWER HERE/>

1. Write a query that indicates for each person if her age is even (true or false).

Query:

<ANSWER HERE/>

## Question 8

1. **Construct** the symmetric of all hasFriend relations using the good SPARQL statement (ex. When finding Thomas hasFriend Fabien, your query should construct Fabien hasFriend Thomas)

Query:

<ANSWER HERE/>

1. **Insert** the symmetric of all hasFriend relations using the adequate SPARQL statement but check the results with a select query before and after.

Query:

<ANSWER HERE/>

## Question 9

Choose and edit one of the SELECT WHERE queries previously written to transform them into a CONSTRUCT WHERE query (retaining the same WHERE clause) in order to visualize the results as a graph.

Query:

<ANSWER HERE/>

Result:

<ANSWER HERE/>

## Question 10

Edit the file to add your own annotation (about you) to the RDF file reusing the properties of the file. Build queries to verify and visualize the annotations you added.

screenshots:

<ANSWER HERE/>

## Question 11

1. Formulate a query to find the persons who share the same shirt size.

Query:

<ANSWER HERE/>

1. Find the persons who have the same size shirt and construct seeAlso relationship between them.

Query:

<ANSWER HERE/>

1. Change the query into an insert.
2. Visualize the resources connected by seeAlso (use the CONSTRUCT clause).

screenshot:

<ANSWER HERE/>

1. Adapt the first query to find persons who have the same shoe size and insert a seeAlso relationship between them.

Query:

<ANSWER HERE/>

1. Visualize the resources connected by seeAlso (use the CONSTRUCT clause)

screenshot:

<ANSWER HERE/>

1. Change the query to find the resources connected by a path consisting of one or several seeAlso relationships.

Query:

<ANSWER HERE/>

1. Reload the engine (option reload in the menu) and rerun the last visualization query.

## Question 12

1. Find the largest shoe size

Query:

<ANSWER HERE/>

1. Find persons who have the biggest size of shoe (subquery + aggregate)

Query:

<ANSWER HERE/>

1. Calculate the average shoe size using the appropriate aggregation operator

Query:

<ANSWER HERE/>

1. Check the average with your own calculation using sum() and count()

Query:

<ANSWER HERE/>

## Question 13

Find couples without children

Query:

<ANSWER HERE/>

## Question 14

Using INSERT DATA, create a new person with its properties. Then, check that it has been created.

Insert:

<ANSWER HERE/>

Screenshot result:

<ANSWER HERE/>

## Question 15

Find the persons connected by paths of any family links. Construct an arc seeAlso between them to visualize the result.

query:

<ANSWER HERE/>

screenshot:

<ANSWER HERE/>

## Question 16

Run the following query:

prefix db: <http://dbpedia.org/ontology/>

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

prefix h: <http://ns.inria.fr/humans/schema#>

construct { ?x h:name ?nx . ?y h:name ?ny . ?x h:hasSpouse ?y }

where {

service <http://fr.dbpedia.org/sparql/> {

select \* where {

?x db:spouse ?y .

?x foaf:name ?nx .

?y foaf:name ?ny .

}

limit 20

}

}

Explain what it does

<ANSWER HERE/>

modify it to insert new persons in the base and check the results.

query:

<ANSWER HERE/>

# Lab session on RDFS.

## Software requirements

* The RDF XML online validation service by W3C: <https://www.w3.org/RDF/Validator/>
* The RDF online translator: <http://rdf-translator.appspot.com/>
* The SPARQL Corese engine (Corese-GUI jar file): <https://project.inria.fr/corese/>

## About the humans schema

1. If you don’t have the human schema file yet use the command curl or wget to obtain the Turtle version of this schema from its namespace and download it in a file named “humans\_schema.ttl”  
     
    [http://ns.inria.fr/humans/schema#](http://ns.inria.fr/humans/schema)
2. What is the namespace of this ontology? How was it specified?

<ANSWER HERE/>

1. Locate the use of the terms of the RDF (S) language: Class, Property, label, comment, range, domain, subClassOf, subPropertyOf, etc. What are their namespaces?

<ANSWER HERE/>

1. What are the classes of resources that can have the age property? Explain

<ANSWER HERE/>

1. Look at the beginning of the file and draw the subgraph of the hierarchy containing the classes Animal, Man and Woman.

Drawing of hierarchy:

<ANSWER HERE/>

## Query the schema itself

Reset or relaunch the standalone Corese search engine interface and load the file humans\_**schema.**ttl (and only this one).

1. Write a query to find all the classes of the ontology.

query:

<ANSWER HERE/>

1. Write a query to find all the links subClassOf in the ontology.

query:

<ANSWER HERE/>

1. Write a query to find the definitions and translations of "shoe size" (*other* labels and comments in different languages for the resource labeled "shoe size").

query:

<ANSWER HERE/>

answers:

<ANSWER HERE/>

1. Write a query to find the synonyms in French of the word 'personne' in French (*other* labels in the same language for the same resource/class/property). What are the answers?

query:

<ANSWER HERE/>

answers:

<ANSWER HERE/>

1. Write a query to find the different meaning of the term "size" (disambiguation using the different comments attached to different resources/classes/properties having the label "size"). What are the answers?

query:

<ANSWER HERE/>

answers:

<ANSWER HERE/>

1. Write a query to find the properties that use the class Person in their signatures?

query:

<ANSWER HERE/>

1. Make CORESE draw the graph of the hierarchy of Classes using a CONSTRUCT query considering only the classes in the humans schema

query:

<ANSWER HERE/>

screenshot:

<ANSWER HERE/>

1. To the previous CONSTRUCT add the signatures of the relations.

query:

<ANSWER HERE/>

screenshot:

<ANSWER HERE/>

**You now know how to query schemas on the semantic Web!**

## Query data augmented by an RDFS schema

## Question 1

1. Reset the Corese engine and load only the data (humans\_**data**.ttl)
2. Write a query to find the Persons.

Query:

<ANSWER HERE/>

Number of results before:

<ANSWER HERE/>

1. Load the schema (humans\_**schema**.ttl)
2. Rerun the query to find the Persons and explain the result.

New number of results after and your explanation:

<ANSWER HERE/>

## Question 2

1. Write a query to find Males and their wives. How many answers do you get? Explain this result.

Query:

<ANSWER HERE/>

Number of results and explanation:

<ANSWER HERE/>

1. In the data declare that Lucas has father Karl. Reset Corese, reload the ontology and the data, and then rerun the query to find Males and their wives. Explain the new result.

 Line added in RDF:

<ANSWER HERE/>

Number of results before and after and explanation:

<ANSWER HERE/>

## Question 3

1. Write a query to find the Lecturers and their types. How many answers do you get? See how this typing is declared in the data and explain the result.

Query:

<ANSWER HERE/>

Number of results and your explanation:

<ANSWER HERE/>

1. Write a query to find common resources both of type Person and of type Male (instances of both classes). See how this typing is declared in the data and explain the presence of Jack.

 Query:

<ANSWER HERE/>

Your explanation of the result:

<ANSWER HERE/>

## Question 4

Write a query to find the hasAncestor relations. Explain the result after checking where this property is used in the data.

 Query:

<ANSWER HERE/>

Your explanation of the result:

<ANSWER HERE/>

## Question 5

1. Write a query to find the family cores (couples and their children) using a SELECT

 Query:

<ANSWER HERE/>

1. Modify it to display the result with a CONSTRUCT query

 Query:

<ANSWER HERE/>

## Question 6

1. Declare the olderThan relationship in the schema to indicate between two persons which is eldest and construct the arcs between persons with a SPARQL query

 Addition to schema:

<ANSWER HERE/>

 Query:

<ANSWER HERE/>

1. Find a query that generates only the minimum number of links without redundancy with olderThan transitivity.

 Query:

<ANSWER HERE/>

## Question 7

Write a query to find for John the properties which label contains the string "size" and the value of these properties.

 Query:

<ANSWER HERE/>

## Question 8

Use the ontology to document your answers in natural language: write a query to find the types and properties of Laura in French.

 Query:

<ANSWER HERE/>

## Create your own schema Family schema (can be done after the OWL practical session too if you are running out of time)

* Write the RDF schema that you used in the description of Jen in a RDFS Turtle (or in RDF/XML and then translate it) and save the RDFS Turtle in a file called “Family\_schema.ttl” (or “Family\_schema.xml”). Of course, this assumes that the URIs for the classes and properties declared/used must match in both files. You may have to update the files Jen.rdf and Jen.ttl to use your ontology.

Your schema:

<ANSWER HERE/>

* Check that your RDF schema and RDF files are valid using the W3C's RDF validation service or other converter/ translators services.
* Launch the standalone interface of Corese and load your files Family\_schema.ttl and Jen.ttl
* The interface contains a default SPARQL query:   
  Select ?x ?t where {?x rdf:type ?t}   
  Launch the query and look at the results.

Screenshot:

<ANSWER HERE/>

* Modify your ontology to declare the classes of Man and Woman as sub classes of Human (don’t change the data), reload the schemas and data and search for the humans to see the results

Screenshot:

<ANSWER HERE/>

Explanation:

<ANSWER HERE/>

* Modify your ontology to declare the properties hasChild and hasSpouse as sub properties of familyLink (don’t change the data), reload the schemas and data and search for the family links to see the results.

Screenshot:

<ANSWER HERE/>

Explanation:

<ANSWER HERE/>

* Modify your ontology to declare the class FamilyMember and use it to specify the signature of the property familyLink (don’t change the data) then reload the schemas and data and search for the family members.

Screenshot:

<ANSWER HERE/>

Explanation:

<ANSWER HERE/>

# Lab session on OWL.

## Software requirements

* The RDF XML online validation service by W3C: <https://www.w3.org/RDF/Validator/>
* The RDF online translator: <http://rdf-translator.appspot.com/>
* The SPARQL Corese engine (Corese-GUI jar file): <https://project.inria.fr/corese/>

## A, Query data augmented by an OWL schema

Make a copy of the humans\_schema.ttl file, name it humans\_owl\_schema.ttl and use it for the rest of the session. For each of the following statements, specify a SPARQL query that shows that the difference before and after running the OWL inferences: you will find that answers to these queries are different depending on whether you load the ontology humans\_schema.ttl or the humans\_owl\_schema.ttl you modified.

**Important:** The “Engine Menu” allows you to control and witness the result of the inferences. If nothing is selected and if you run no rules you will just have the graph you loaded. If you start applying RDFS or OWL you will see new inferred results being added. For this practical session make sure to apply (unselect and reselect) “OWL RL” in the engine menu of Corese to run the rules to see the addition of results. Depending on the version of the CORESE-GUI you use you may have to repeat this operation several times to see all results.

1. Declare that hasSpouse is a symmetrical property and do the same for hasFriend .

Code added to the schema:

<ANSWER HERE/>

Query:

<ANSWER HERE/>

Result before addition to the schema:

<ANSWER HERE/>

Result after addition to the schema (reload then unselect and reselect “OWL RL”):

<ANSWER HERE/>

Explanation:

<ANSWER HERE/>

1. Declare that hasChild is the inverse property of the hasParent property.

Code added to the schema:

<ANSWER HERE/>

Query:

<ANSWER HERE/>

Result before addition to the schema:

<ANSWER HERE/>

Result after addition to the schema (reload then unselect and reselect “OWL RL”):

<ANSWER HERE/>

Explanation:

<ANSWER HERE/>

1. Declare hasAncestor as transitive property.

Code added to the schema:

<ANSWER HERE/>

Query:

<ANSWER HERE/>

Result before addition to the schema:

<ANSWER HERE/>

Result after addition to the schema (reload then unselect and reselect “OWL RL”):

<ANSWER HERE/>

Explanation:

<ANSWER HERE/>

1. Declare and define the chain property hasSibling has a super-property of the existing properties hasBrother and hasSister.

Code added to the schema:

<ANSWER HERE/>

Query:

<ANSWER HERE/>

Result before addition to the schema:

<ANSWER HERE/>

Result after addition to the schema (reload then unselect and reselect “OWL RL”):

<ANSWER HERE/>

Explanation:

<ANSWER HERE/>

1. Declare and define the chain properties: hasUncle and hasAunt and in the data declare that Jack and Pierre are brothers and vice-versa.

Code added to the schema:

<ANSWER HERE/>

Query:

<ANSWER HERE/>

Result before addition to the schema (reload then unselect and reselect “OWL RL”):

<ANSWER HERE/>

Result after addition to the schema:

<ANSWER HERE/>

1. Declare the disjunction between Male and Female. Violate the constraint in the data, check the results and then remove the violation you created.

Code added to the schema:

<ANSWER HERE/>

Query:

<ANSWER HERE/>

Result before addition to the schema:

<ANSWER HERE/>

Result after addition to the schema (reload then unselect and reselect “OWL RL”):

<ANSWER HERE/>

Explanation:

<ANSWER HERE/>

1. Declare that the class Professor is the intersection of the class Lecturer and Researcher class.

Code added to the schema:

<ANSWER HERE/>

Query:

<ANSWER HERE/>

Result before addition to the schema:

<ANSWER HERE/>

Result after addition to the schema (reload then unselect and reselect “OWL RL”):

<ANSWER HERE/>

Explanation:

<ANSWER HERE/>

1. Declare that the Academic class is the union of classes Lecturer and Researcher.

Code added to the schema:

<ANSWER HERE/>

Query:

<ANSWER HERE/>

Result before addition to the schema:

<ANSWER HERE/>

Result after addition to the schema (reload then unselect and reselect “OWL RL”):

<ANSWER HERE/>

Explanation:

<ANSWER HERE/>

1. Create a class Organization and its sub class University. Create a new property mainEmployer, with domain Person and range Organization. Use a restriction to declare that any Professor has for main employer a University.

Code added to the schema (new property, new classes and new restriction):

<ANSWER HERE/>

Code added to the data (just declare the main employer of a Professor):

<ANSWER HERE/>

Query:

<ANSWER HERE/>

Result before addition to the schema:

<ANSWER HERE/>

Result after addition to the schema (reload then unselect and reselect “OWL RL”):

<ANSWER HERE/>

Explanation:

<ANSWER HERE/>

1. Use a restriction to declare that any person must have a parent who is a woman. For this last statement, you need to run the rule engine after loading the ontology and data.

Code added to the schema:

<ANSWER HERE/>

Query:

<ANSWER HERE/>

Result before addition to the schema:

<ANSWER HERE/>

Result after addition to the schema (reload then unselect and reselect “OWL RL”):

<ANSWER HERE/>

Explanation:

<ANSWER HERE/>

## B, Make your own OWL models:

For each one of the following OWL primitives imagine a definition that could use it and provide that definition in OWL using your preferred syntax (RDF/XML or N3/Turtle). For instance a possible definition using owl:TransitiveProperty would be a definition of the Ancestor property. For each primitive in the following list you imagine the definition of a class or property that was not given in the course and you give that definition in English and in OWL.

1. owl:oneOf <YOUR EXAMPLE HERE/>
2. owl:unionOf <YOUR EXAMPLE HERE/>
3. owl:intersectionOf <YOUR EXAMPLE HERE/>
4. owl:complementOf <YOUR EXAMPLE HERE/>
5. owl:disjointWith <YOUR EXAMPLE HERE/>  
    or owl:AllDisjointClasses  
    or owl:disjointUnionOf
6. owl:ObjectProperty <YOUR EXAMPLE HERE/>
7. owl:DatatypeProperty <YOUR EXAMPLE HERE/>
8. owl:SymmetricProperty <YOUR EXAMPLE HERE/>  
   or owl:AsymmetricProperty
9. owl:inverseOf <YOUR EXAMPLE HERE/>
10. owl:TransitiveProperty <YOUR EXAMPLE HERE/>
11. owl:propertyDisjointWith <YOUR EXAMPLE HERE/>
12. owl:ReflexiveProperty <YOUR EXAMPLE HERE/>  
     or owl:IrreflexiveProperty
13. owl:propertyChainAxiom <YOUR EXAMPLE HERE/>
14. owl:FunctionalProperty <YOUR EXAMPLE HERE/>
15. owl:InverseFunctionalProperty <YOUR EXAMPLE HERE/>
16. owl:hasKey <YOUR EXAMPLE HERE/>
17. owl:allValuesFrom <YOUR EXAMPLE HERE/>
18. owl:someValuesFrom <YOUR EXAMPLE HERE/>
19. owl:hasValue <YOUR EXAMPLE HERE/>
20. owl:maxCardinality <YOUR EXAMPLE HERE/>  
    or owl:minCardinality
21. owl:qualifiedCardinality <YOUR EXAMPLE HERE/>