

Evaluation of INFO216

Please take some time to answer this survey!

The feedback received from this survey is used to improve the INFO216 course. For instance, if you felt some labs were easier or tougher than they should have been, then this feedback is very much appreciated.

TABLE OF CONTENTS

01

Exam 2023

General info...

02

Previous Exams

Proposed solutions, explanations...

03

SPARQL

Overview, resources

04

XSD, RDF, RDFS, SHACL & OWL

Overview, resources...

05

MISC

Lab presentations, old lab tasks...



EXAM 2023

General Info

You can find information about the exam here

- 4 hour school exam
- Date: 8. June 2023
- Start time 09:00 or 15:00 check 2 weeks before the exam
- Digital exam (bring your computer)
- No resources available (no notes, books etc.)

Information about exams at UIB is available **here**

General Info

Andreas has published some information about the exam **here**.

Last year's exam can be found **here**, and other exams can be found **here**.

The best way to practice is through doing, so try to complete the previous exams (particularly last year's and 2021), **in addition** to the labs. All solutions to the labs can be found **here** and **here** (lab 3).

NOTE: Some subjects are curriculum this year, where previous years it hasn't been (e.g. SHACL).



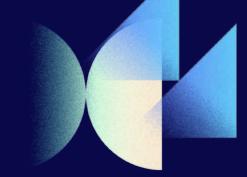
Previous Exams

Exam 2021 - Turtle & SPARQL

The answers to the turtle and SPARQL questions can be found under the following <u>discussion thread</u> (below information about the exam).

Otherwise, the exam has a lot of multiple choice questions, which includes good practice for general knowledge of the course, and questions on vocabularies.

Exam 2022 - Info



Task 1 - General Knowledge

Task 2 - Ontology Building

Task 2a - Ontology Rules

Task 2b - Ontology Individuals

Task 3 - Vocabularies

Task 4 - RDF, RDFS & OWL

Task 4a - RDF, RDFS & Basic OWL

Task 4b - Complex OWL

Task 5 - Reasoning

Task 6 - SPARQL

Task 6a - SPARQL Queries

Task 6b - SPARQL Updates

Task 7 - RDFLIB Understanding

78 tasks in total, 44 multiple choice & 34 programming

Exam 2022 - Task 7

How many errors can you spot in the following screenshot?

You can find the file containing errors <u>here</u> and my solution <u>here</u>.

```
from rdflib import Graph, Namespace, RDF, RDFS, OWL, XSD, Literal
# Define paths and file names
BASE = 'http://ex.org#'
ONT FILE = 'ontology.ttl'
PAPER FILE = 'papers.txt'
AUTHOR FILE = 'authors.txt'
KG FILE = 'created kg.ttl'
# Create rdflib graph
Graph()
# Read ontology from file
g.parse(ONT FILE, format='owl')
def str to id(str):
    """Make sure that the string is a legal part of a URL."""
    return str.replace(' ', ' ')
def add paper (q, line):
    """Add information about a paper to the graph."""
    if line.strip() == '0':
        return
    title, author names, source, pub name, year = line.split(';')
    paper = BASE['publ '+str to id(title.strip())]
    g.add(paper, RDF.type, BASE.Paper)
    g.add(paper, BASE.title, title.strip())
    g.add(paper, BASE.year, int(year))
```



Exam 2022 - Answers

If you spot any errors let me know!



SPARQL

SPARQL Queries

```
PREFIX ns1: <a href="http://example.org#">http://example.org#>

SELECT ?invs (COUNT(?invs) as ?count)
WHERE {
    ?s ns1:investigation ?invs;
    ns1:outcome ns1:indictment .
}

GROUP BY ?invs
HAVING(?count > 1)
ORDER BY DESC(?count)
```

Example of SELECT query

Question: What's the difference between ?s and ns1:indictment

SELECT: returns table

ASK: returns yes/no

DESCRIBE: returns a graph

CONSTRUCT: returns a graph



SPARQL Updates

INSERT DATA

DELETE DATA

INSERT WHERE

DELETE WHERE

DELETE INSERT WHERE:

DELETE {?paper dct:contributor kg:auth_456}

INSERT {?paper dct:contributor kg:auth_654}

WHERE {?paper dct:contributor kg:auth_456}



```
SELECT DISTINCT (SUM(?lore) as ?age) WHERE{
     #Triples
     FILTER {?age >= "18"^^xsd:integer}
     BIND (year(?age) as ?por)
     OPTIONAL {}
     {} UNION {}
     :subject (:property / rdf:rest* /rdf:first) ?object .
GROUP BY ?age
HAVING (?age > 5)
ORDER BY DESC ?age
LIMIT 2
```

This is <u>not</u> a definitive list! These are just some examples, there are plenty more. For example, instead of SUM you could have COUNT, AVG, MIN and/or MAX.

SPARQL Resources

Documentation

- Lecture Notes
- Query Documentation
- Update Documentation
- Property Path
- Expressions and Functions
- RDFLIB SPARQL
- Readings list for Lecture 3

Tasks

- Lab 3 (SPARQL)
- Lab 4 (Programming SPARQL)

Solutions

- Lab 3 Solution
- Lab 4 Solution



XSD, RDF, RDFS, SHACL & OWL

XSD (Datatypes)

xsd:string, xsd:boolean, xsd:decimal, xsd:integer, xsd:float, xsd:double, xsd:dateTime, xsd:dateTimeStamp, xsd:time, xsd:date, xsd:gYearMonth, xsd:gYear, xsd:gMonthDay, xsd:gDay, xsd:gMonth, xsd:duration, xsd:yearMonthDuration, xsd:dayTimeDuration, xsd:hexBinary, xsd:base64Binary, xsd:anyURI, xsd:normalizedString, xsd:token, xsd:language, xsd:NMTOKEN, xsd:Name, xsd:NCName, xsd:positiveInteger, xsd:nonPositiveInteger, xsd:negativeInteger, xsd:long, xsd:int, xsd:short, xsd:byte, xsd:nonNegativeInteger, xsd:unsignedLong, xsd:unsignedInt, xsd:unsignedShort, xsd:unsignedByte, xsd:minInclusive, xsd:maxInclusive

NOTE: There are more than the ones listed, this is not a definitive list of what to remember and not to remember, just an overview of some of the ones used!



RDF & RDFS

RDF:

RDF.type

RDFS:

RDFS.subClassOf, RDFS.subPropertyOf, RDFS.domain, RDFS.range

RDFS.label, RDFS.comment, RDFS.seeAlso, RDFS.isDefinedBy, RDFS.Datatype

NOTE: There are more than the ones listed, this is not a definitive list of what to remember and not to remember, just an overview of some of the ones used!



SHACL

General Concepts

- sh:NodeShape
- sh:targetClass
- sh:property
- sh:path

Others

- sh:class
- sh:minCount
- sh:maxCount
- sh:datatype
- sh:or
- sh:and

- sh:hasValue
- sh:value
- sh:pattern
- sh:in
- sh:nodeKind

Check **table 5.3** under **5.6.3 Constraint Components** in the documentation for more core concepts of SHACL

<u>NOTE</u>: There are more than the ones listed, this is not a definitive list of what to remember and not to remember, just a short overview of some of the concepts used



Basic OWL

(sameAs, differentFrom, AllDifferent, equivalentClass, disjointWith, AllDisjointWith, equivalentProperty, propertyDisjointWith, AllDisjointProperties, inverseOf)

(SymmetricProperty, AsymmetricProperty, ReflexiveProperty, IrreflexiveProperty, TransitiveProperty, FunctionalProperty, InverseFunctionalProperty, propertyChainAxiom)

NOTE: There are more than the ones listed, this is not a definitive list of what to remember and not to remember, just an overview of some of the ones used!

Complex OWL

(oneOf, unionOf, intersectionOf. complementOf)

(Restriction, Class, onProperty, onClass, hasSelf, onDatatype)

(someValuesFrom, allValuesFrom, hasValue)

(cardinality, minCardinality, maxCardinality)

(qualifiedCardinality, minQualifiedCardinality, maxQualifiedCardinality)

(NegativePropertyAssertion, sourceIndividual, assertionProperty, targetIndividual, withRestriction)

NOTE: There are more than the ones listed, this is not a definitive list of what to remember and not to remember, just an overview of some of the ones used!



Resources

RDF & RDFS

Documentation

- <u>Lecture 1 Notes</u>
- Lecture 2 Notes
- Lecture 7 Notes
- Reading list lecture 1, 2 & 7
- Many more, check reading list & useful materials

Tasks:

- Lab 1 Intro to RDF
- Lab 2 More RDF
- Lab 5 CSV to RDF
- Lab 7 RDFS
- Solutions

OWL

Documentation

- Lecture 8 Notes
- Lecture 10 Notes
- Reading list for Basic OWL
- Reading list for complex OWL

Tasks:

- Lab 8 Basic OWL
- Lab 9 Protégé OWL
- Lab 10 Protégé OWL 2
- Solutions

SHACL

Documentation

- Lecture 7 Notes
- SHACL Documentation
- SHACL Playground
- pySHACL

Tasks:

- Lab 6 SHACL
- Lab 6 Solution



Miscellaneous

Lab Presentations & Examples

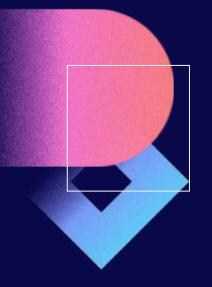
Lab presentations for lab 6, 7, 8 & 9 can be found <u>here</u> and combined <u>here</u>.

The INFO216 folder on GitHub can be found here

Old Labs

You can find old labs in the course by checking the history tab in the labs section. However, most of the new labs you have gone through are more relevant and better than the old ones.

This <u>lab</u> can be useful to practice writing complex owl in turtle and/or RDFLIB. The proposed solution can be found here (RDFLIB) & here (turtle).







Remember we have an <u>INFO216 Discord</u> you can use to help each other!





