

COMP 474 UU,COMP 6741 UU 2204

[Home](#) / [My courses](#) / [COMP-474-2204-UU](#) / 24 January - 30 January / [Lab Session #2](#)

Lab Session #2

Introduction

Welcome to the second lab. This time, we'll start programming knowledge graphs using Python.

Task #1: RDF

Your first task is to translate some of the knowledge graphs you developed on last week's Worksheet #1 into a real RDF graph. Write down the triples using the Turtle format discussed in the lecture. Then, validate your graph by:

1. Using a browser, go to <http://ttl.summerofcode.be> and paste your Turtle code into the designated text area.
2. Click the "Validate!" button.
3. Examine the results of parsing the input. Correct any mistakes that you might have made accordingly.
4. If no mistakes are found in the input, you should see a message that reads "Congrats! Your syntax is correct."

There are a number of other RDF-related tools online; for example, try out the RDF converter at <http://rdfvalidator.mybluemix.net/> and convert your Turtle file (.ttl) into JSON-LD and RDF/XML to get an idea how these formats look like. The validator at <https://www.w3.org/RDF/Validator/> only accepts RDF/XML, but it can additionally draw you a graph corresponding to your triples (under the "Display Result Options", select "Triples and Graph"). Convert your RDF file from Turtle to RDF/XML and visualize it in form of a graph.

Note: you will probably encounter references to `rdfs:` (RDF Schema) in examples you find online; we will cover the details of RDFS in this week's lecture.

Task #2: RDFlib

For working with RDF and related standards, there are a multitude of libraries available. For example, a popular open source framework for Java is [Apache Jena](#). Here, we will use the [RDFlib](#) for Python ([documentation](#)).

Install RDFlib and try loading the graph you prepared in Task #1. Print out your whole graph `g` using

```
for s,p,o in g:
    print s,p,o
```

Next, try to export your graph in N-Triples (N3) format and check the output.

Go through the following four tasks in the [RDFlib documentation](#) under "Getting Started":

- [Getting started with RDFlib](#)
- [Loading and saving RDF](#)
- [Creating RDF triples](#)
- [Navigating Graphs](#)

(SPARQL will be the topic of lecture #4).

Task #3: Hello, Eliza!

In the first lecture, you've seen Eliza, which is probably the oldest example of a chatbot/intelligent agent:

- Find an online version of Eliza and try it out (note that not all versions you will find will use the original DOCTOR script);

- Make sure you understand [how Eliza works](#);
- Find a source code version of Eliza that you can modify and run locally. Add "Concordia" as a new keyword, together with some suitable decomposition & reassembly rules and try it out.

That's all for this lab!

Last modified: Tuesday, 9 February 2021, 12:47 PM

[◀ Worksheet #02](#)

Jump to...



[Lecture Slides #04 ▶](#)

You are logged in as [Mitalee Chandrakant Naik](#) ([Log out](#))

[COMP-474-2204-UU](#)

[Academic Integrity](#)

[Academic Assessment Tool](#)

[English \(en\)](#)

[Deutsch \(de\)](#)

[English \(en\)](#)

[Español - Internacional \(es\)](#)

[Français \(Canada\) \(fr_ca\)](#)

[Français \(fr\)](#)

[Italiano \(it\)](#)

[العربية \(ar\)](#)