

# ナレッジグラフ 推論チャレンジ

Knowledge Graph Reasoning Challenge

学生向け！

2021/08/30

技術勉強会  
～ナレッジグラフ利用技術の基礎と推論/説明への活用例～

ナレッジグラフ/LOD  
利用技術の入門（後編）

発表者：江上周作

# コンテンツ

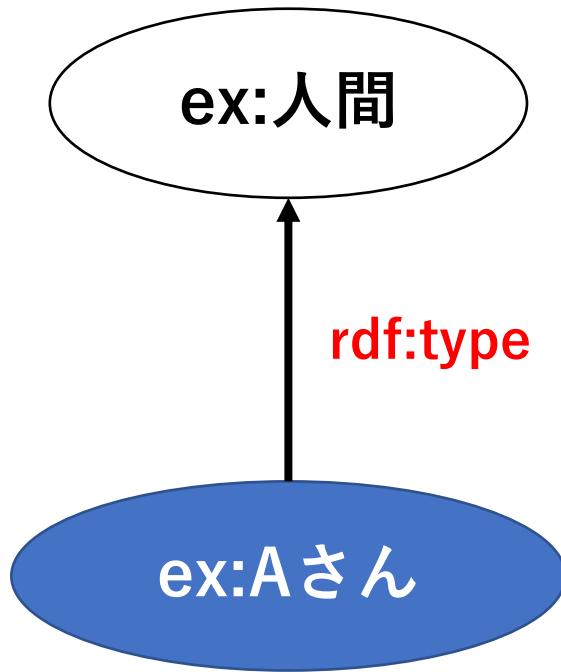
---

- (補足：ナレッジグラフの基礎)
- RDFデータベース（トリプルストア）の導入方法
- Pythonライブラリ「RDFLib」によるRDF利用プログラミング

# **ナレッジグラフ (RDF) の基礎 (続き)**

クラス, インスタンス, サブクラス, サブプロパティ

# クラスとインスタンス



「人間」さんは実在しない。  
概念、カテゴリ、グループのニュアンス。  
これを**クラス (Class)** という

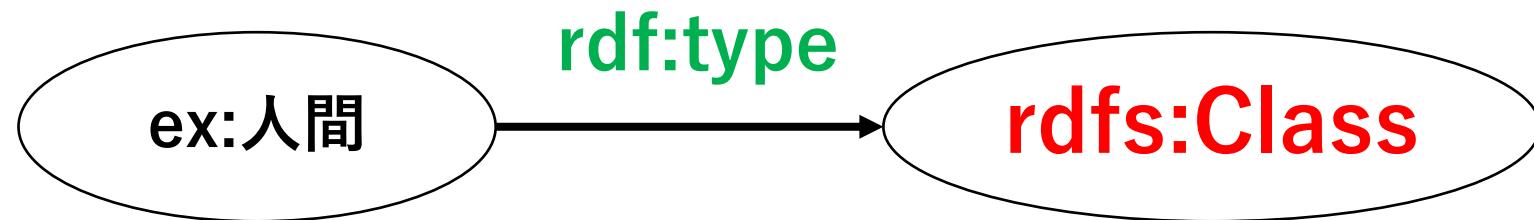
「Aさん」は実在する。  
人間を具体化させたものの一例。  
人間クラスのインスタンスという。

クラス-インスタンスの関係は**rdf:type**プロパティを使用する

# クラスとインスタンス

「ex:人間」をクラスとして定義する

書きたい情報 「ex:人間はクラスである」



「rdfs」は「<http://www.w3.org/2000/01/rdf-schema#>」

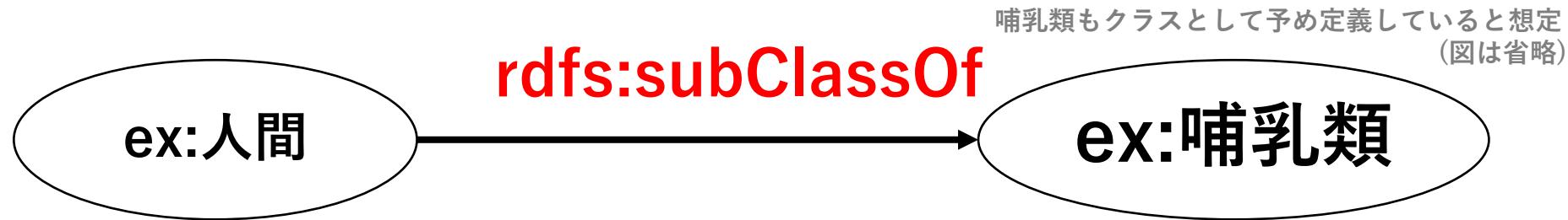
RDFSはクラスを定義するための用語をRDFで提供している。  
↑アクセスしてみるとRDFSのTurtleが見える

クラスの指定には「rdf:type」を使う決まり

# クラスの階層関係

「ex:人間」を「ex:哺乳類」のサブクラスとして定義する

書きたい情報 「ex:人間はex:哺乳類のサブクラスである」

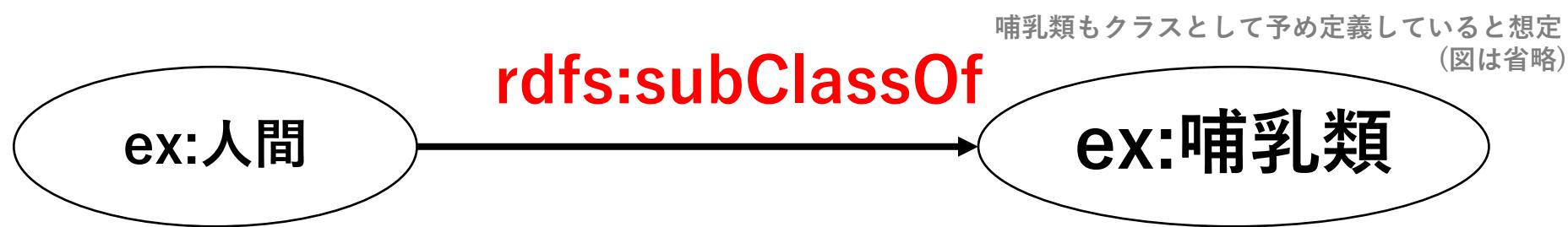


階層関係を意味するプロパティとして  
rdfs:subClassOfが一般的に使用される。

# クラスの階層関係

「ex:人間」を「ex:哺乳類」のサブクラスとして定義する

書きたい情報 「ex:人間はex:哺乳類のサブクラスである」

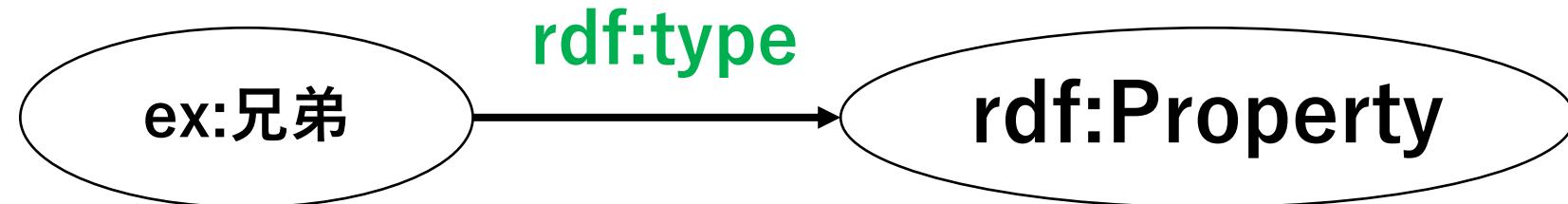


この定義をした上で  
「ex:Aさん rdf:type ex:人間 .」と書くと,  
「ex:Aさん rdf:type ex:哺乳類 .」を  
記述しなくても当然導ける = 【推論可能】

# プロパティ

「ex:兄弟」をプロパティとして定義する

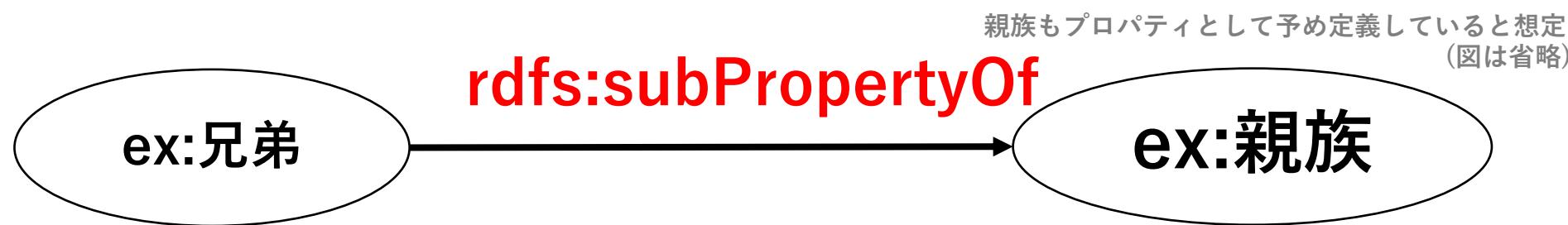
書きたい情報 「ex:兄弟はプロパティである」



# プロパティの階層関係

「ex:兄弟」を「ex:親族」のサブプロパティとして定義する！

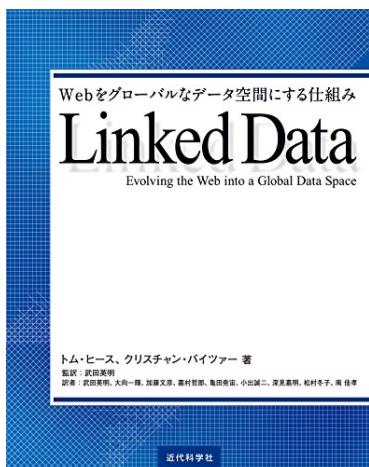
書きたい情報 「ex:兄弟はex:親族のサブプロパティである」



兄弟プロパティを使って「A ex:兄弟 B.」と書くと,  
「A ex:親族 B.」が当然導ける = 【推論可能】

# Linked Dataの基本原則

1. あらゆる事物にURIを付与すること
2. 誰でも事物の内容が確認できるように、URIはHTTP経由で参照できること
3. URIを参照した時は、標準の技術（RDFやSPARQL等）を使用して関係する有用な情報を利用できるようにすること
4. より多くの事物を発見できるように、他のURIへのリンクを含めること



詳しく理解するには下記の本がおすすめ

Linked Data : Webをグローバルなデータ空間にする  
仕組み

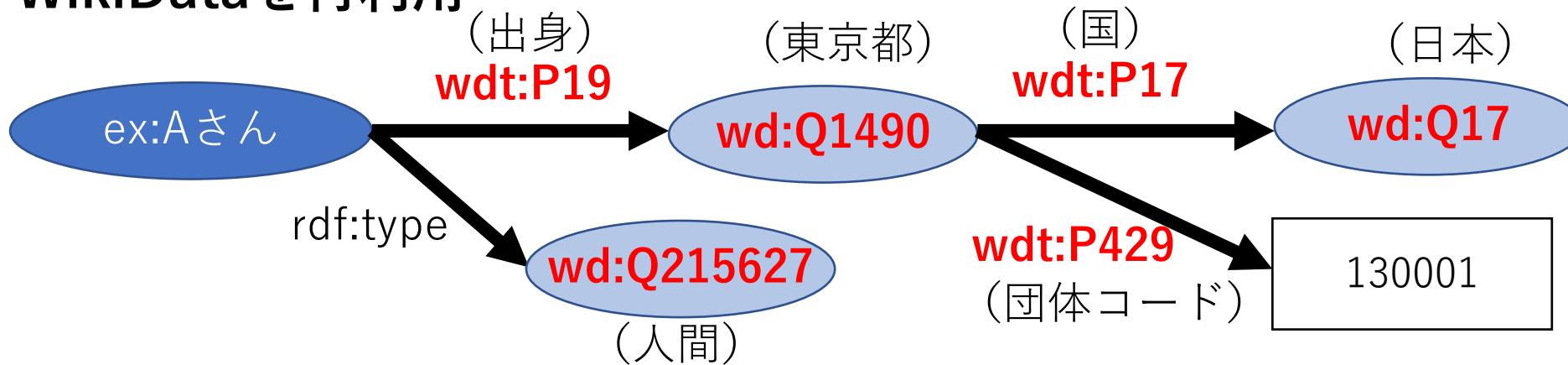
トム・ヒース、クリスチャン・バイツァー 著

監訳：武田英明

訳者：武田英明、大向一輝、加藤文彦、嘉村哲郎、亀田亮宙、小出誠二、深見嘉明、松村冬子、南 佳孝

# 既存のLODを利用

## WikiDataを再利用



```
@prefix ex: <http://example.com/terms/> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix wdt: <http://www.wikidata.org/prop/direct/> .  
@prefix wd: <http://www.wikidata.org/entity/> .  
ex:Aさん  
    rdf:type wd:Q215627 ;  
        wdt:P19 wd:Q1490 .  
  
wd:Q1490  
    wdt:P17 wd:Q17 ;  
        wdt:P429 "130001" .
```

```
C:\Users\s-egami\Downloads\test.ttl - EmEditor  
ツール ファイル(F) 切り取り(Shift+Delete, Ctrl+X) ホル(T) ウィンドウ(W) ヘルプ(H)  
test.ttl x  
1 @prefix ex: <http://example.com/terms/> .  
2 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
3 @prefix wdt: <http://www.wikidata.org/prop/direct/> .  
4 @prefix wd: <http://www.wikidata.org/entity/> .  
5 ex:Aさん  
6     rdf:type wd:Q215627 ;  
7         wdt:P19 wd:Q1490 .  
8  
9 wd:Q1490  
10     wdt:P17 wd:Q17 ;  
11         wdt:P429 "130001" .
```

# トリプルストアの導入

# RDFデータベース（トリプルストア）

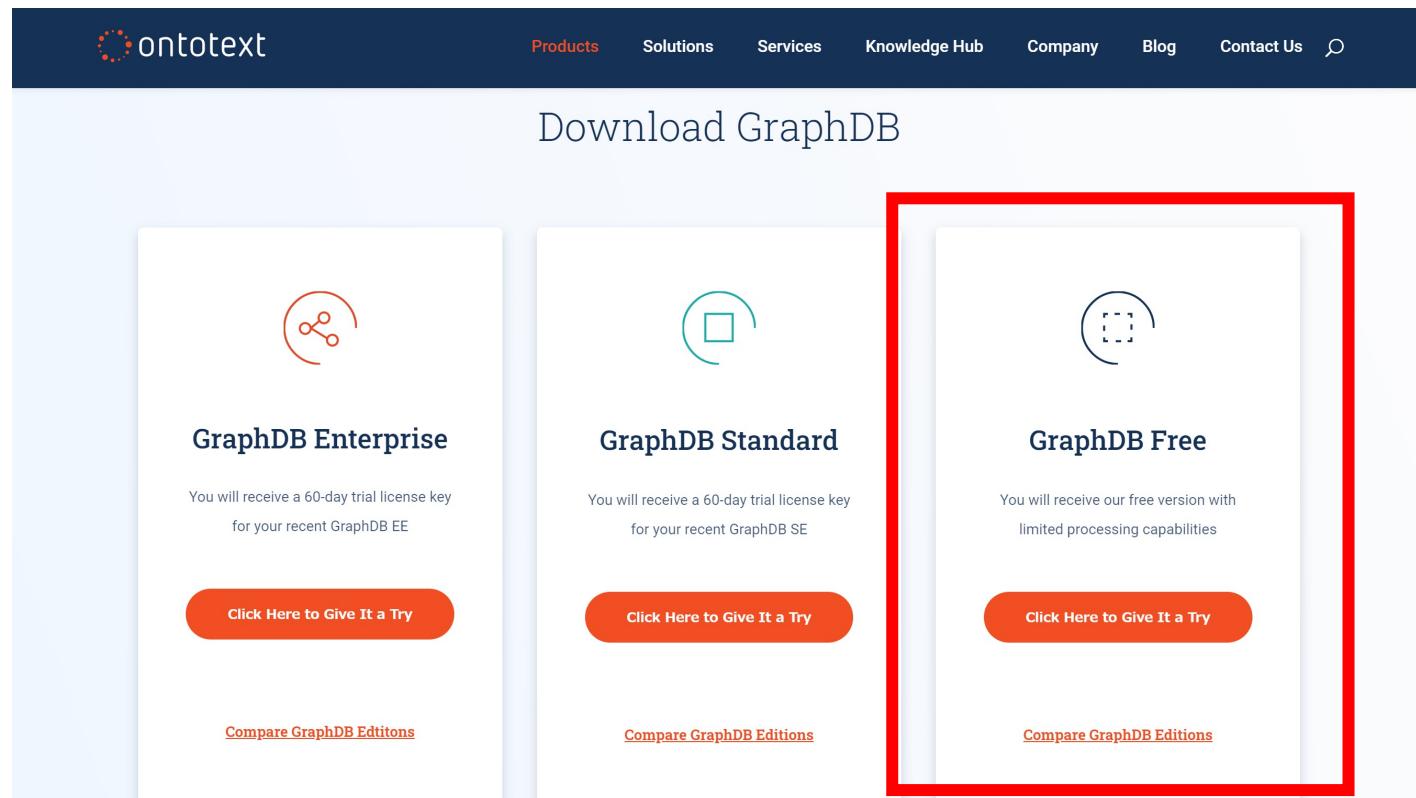
- RDFデータを格納し、SPARQLによる操作を可能にするデータベース
  - トリプルストア、RDFストアとも呼ばれる
- オープンソース、商用とともに様々なトリプルストアが存在
  - AllegroGraph
  - Amazon Neptune
  - Blazegraph
  - Dydra
  - GraphDB by Ontotext
  - MarkLogic
  - OpenLink Virtuoso
  - Oracle
  - Stardog
  - ...

詳しくはこちら

[https://en.wikipedia.org/wiki/Comparison\\_of\\_triplestores](https://en.wikipedia.org/wiki/Comparison_of_triplestores)

# RDFデータをトリプルストアに格納

- GraphDB by Ontotextのフリー版で説明
  - 下記ボタンの遷移先フォームから必要情報を入力して送信するとダウンロードリンクが送られてきます



<https://www.ontotext.com/products/graphdb/>

# リポジトリの作成

ここから実演します

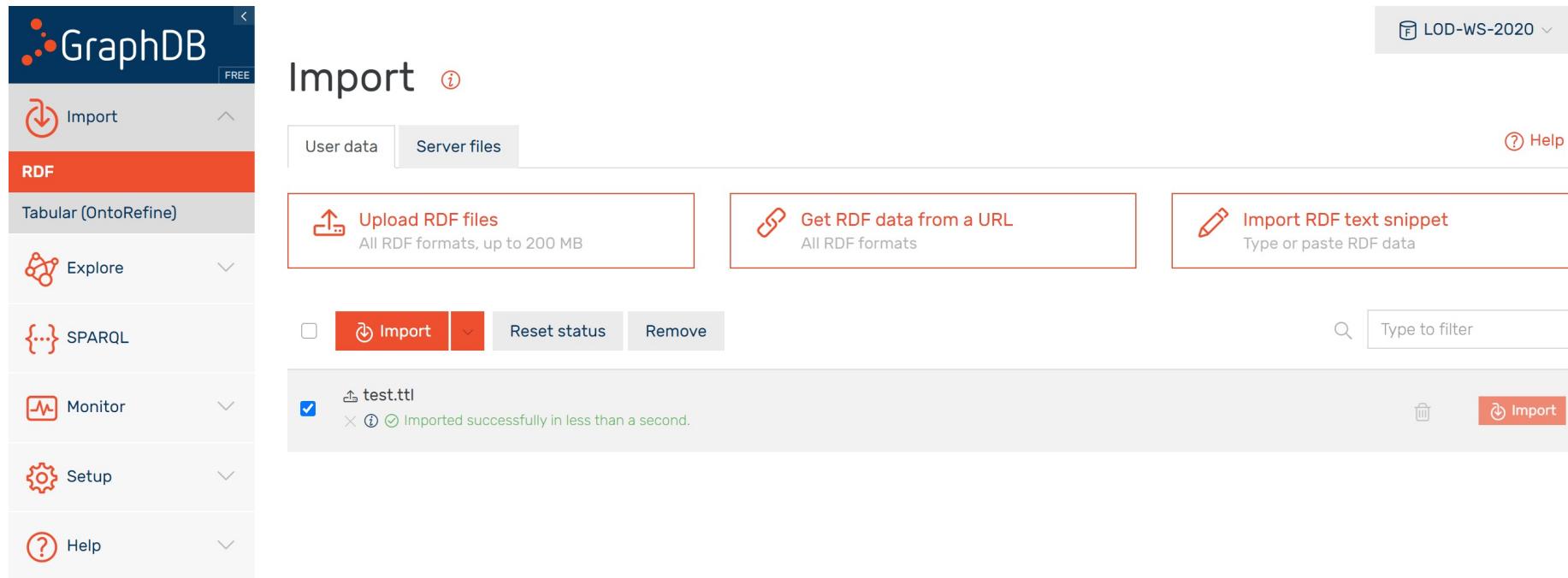
- Setup > Repositories > Create Repositories

The screenshot shows the GraphDB interface with the 'Create Repository' form. The left sidebar has a red highlight on 'Repositories'. The main form fields are:

- Repository ID\*: LOD-WS-2020
- Repository title: LOD-WS-2020
- Type: GRAPHDB-FREE
- Storage folder: storage
- Ruleset: RDFS-Plus (Optimized)
- Upload custom ruleset button
- Checkboxes:
  - Disable owl:sameAs
  - Supports SHACL validation
- Base URL: http://example.org/owlim#
- Entity index size: 10000000
- Checkboxes (continued):
  - Use predicate indices
  - Cache literal language tags
  - Use context index
  - Enable literal index
  - Check for inconsistencies
  - Throw exception on query time-out
  - Read-only
- Entity ID bit-size: 32

# インポート

- Import > RDF > Upload RDF Files > ファイル選択 > Import



# データの確認：トリプル

- Explore > Graph overview > The default graph

The screenshot shows the GraphDB interface with the following details:

- Left Sidebar:** Includes "Import", "Explore" (selected), "Graphs overview", "Class hierarchy", "Class relationships", "Visual graph" (selected), "Similarity", "SPARQL", "Monitor", "Setup", and "Help".
- Top Bar:** Shows "LOD-WS-2020" and a dropdown menu.
- Title:** "nil" with an edit icon.
- Source:** "Source: http://www.openrdf.org/schema/sesame#nil"
- Filter Bar:** Buttons for "subject", "predicate", "object", "context", "all" (selected), "Explicit only" dropdown, "Show Blank Nodes" (disabled), "Download as" dropdown, and "Visual graph" button.
- Data Table:** A table showing four triples:

	subject	predicate	object	context
1	ex:Aさん	rdf:type	wd:Q215627	http://www.ontotext.com/explicit
2	ex:Aさん	wdt:P19	wd:Q1490	http://www.ontotext.com/explicit
3	wd:Q1490	wdt:P17	wd:Q17	http://www.ontotext.com/explicit
4	wd:Q1490	wdt:P429		
- Bottom Right:** A "Visual graph" window showing a diagram with nodes "Aさん" (red circle), "Q1490" (blue circle), and "Q17" (light blue circle) connected by arrows labeled "P19" and "P17".
- Text:** "可視化も" (Visualization also).

# データの確認：グラフ

- Visual graph

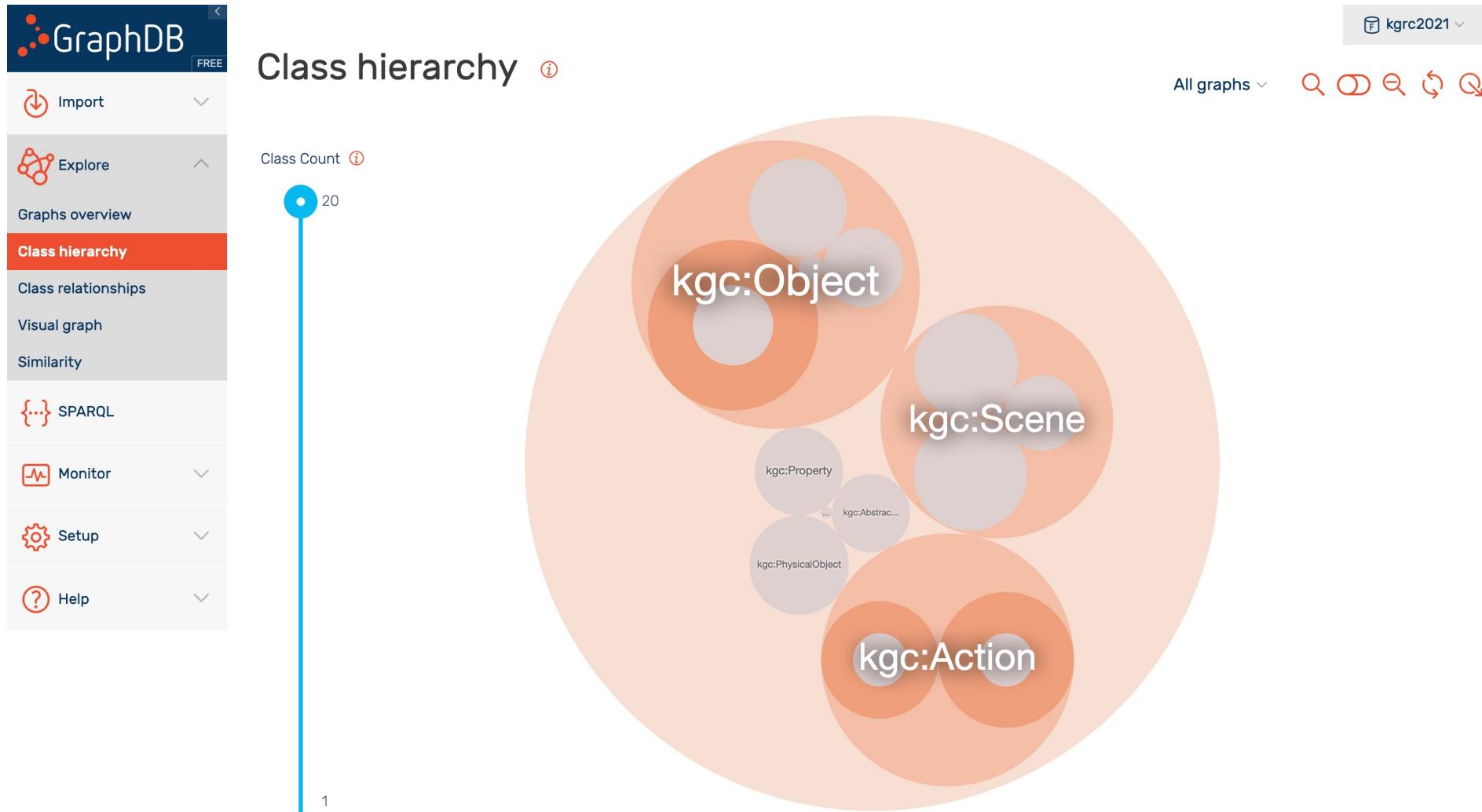
The screenshot shows the GraphDB interface. On the left, there's a sidebar with various navigation options like Import, Explore, SPARQL, Monitor, Setup, and Help. The main area is titled 'nil' and shows a table of triples from the source <http://www.openrdf.org/schema/sesame#nil>. The table has columns for subject, predicate, object, and context. A blue arrow points from the 'Visual graph' button in the top right of the main area to the right panel.

	subject	predicate	object	context
1	ex:Aさん	rdf:type	wd:Q215627	<a href="http://www.ontotext.com/explicit">http://www.ontotext.com/explicit</a>
2	ex:Aさん	wdt:P19	wd:Q1490	<a href="http://www.ontotext.com/explicit">http://www.ontotext.com/explicit</a>
3	wd:Q1490	wdt:P17	wd:Q17	<a href="http://www.ontotext.com/explicit">http://www.ontotext.com/explicit</a>
4	wd:Q1490	wdt:P429	130001	<a href="http://www.ontotext.com/explicit">http://www.ontotext.com/explicit</a>

The right panel shows the 'Visual graph' view with a title 'Visual graph'. It displays a network of nodes: a red node labeled 'Aさん', a light blue node labeled 'Q1490', and another light blue node labeled 'Q17'. Arrows indicate relationships: one arrow points from 'Aさん' to 'Q1490' labeled 'P19', and another arrow points from 'Q1490' to 'Q17' labeled 'P17'.

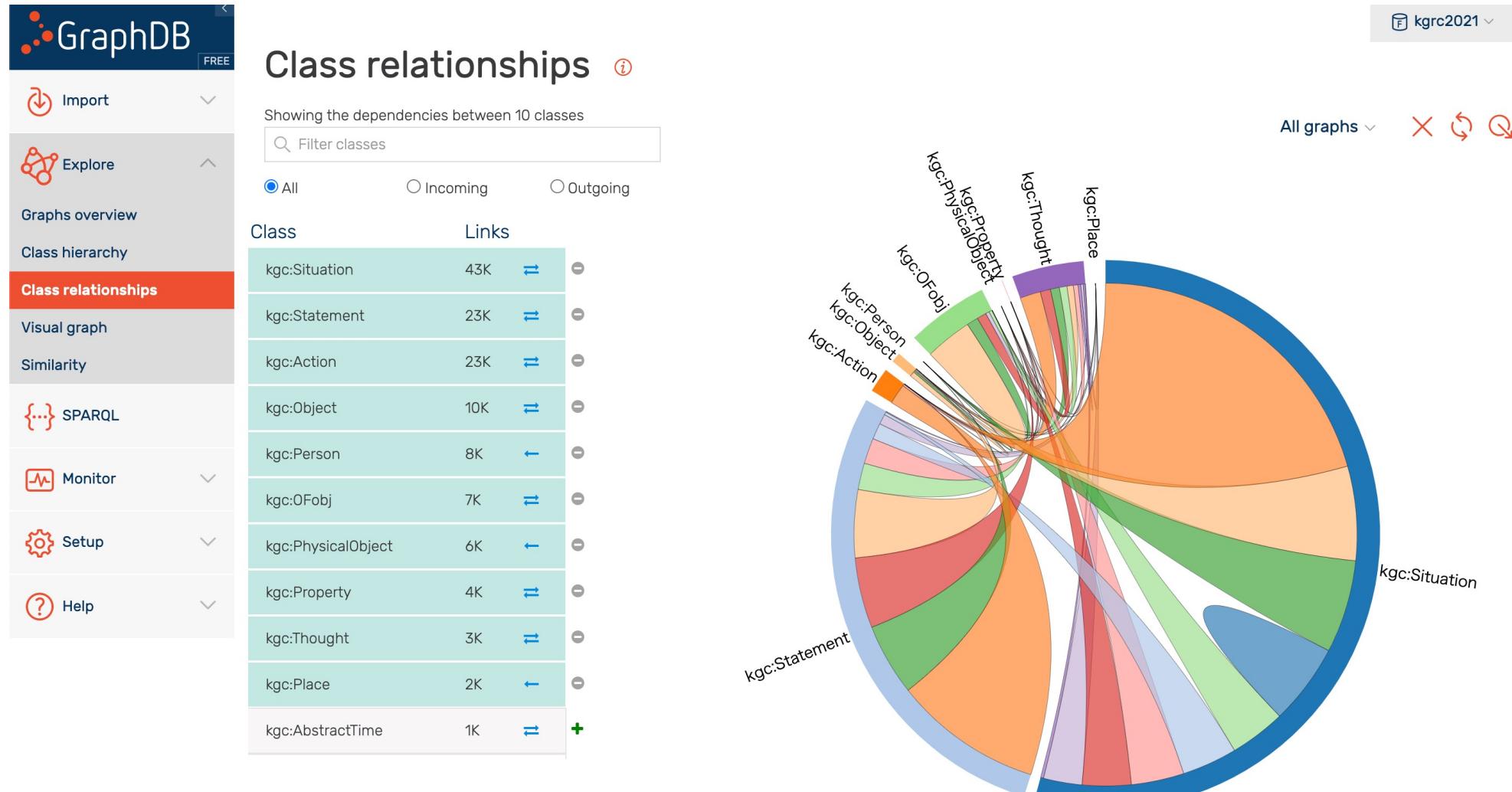
# データの確認：クラス階層

- Explore > Class hierarchy



# データの確認：クラス関係

- Explore > Class relationships



# SPARQL検索

- 検索が可能

The screenshot shows the GraphDB interface with the following details:

- Left Sidebar:** Includes "Import", "Explore" (with sub-options: Graphs overview, Class hierarchy, Class relationships, Visual graph, Similarity), "SPARQL" (selected and highlighted in orange), "Monitor", "Setup", and "Help".
- Top Bar:** Shows the dataset "LOD-WS-2020" and tabs for "Editor only", "Editor and results" (selected), and "Results only".
- Query Editor:** A code editor window titled "Unnamed" containing the following SPARQL query:

```
1 select * where {
2   ?s ?p ?o .
3 } limit 100
```
- Right Panel:** Contains several red icons: a save icon, a folder icon, a link icon, a double arrow icon, and a triple node icon.
- Run Button:** A red button labeled "Run" located at the bottom right of the query editor.
- Result View:** A table showing the results of the query. The table has columns for "s" and "p". The first row shows "rdf:type" in both columns. A tooltip for "rdf:type" indicates it is an "rdf:Property".
- Bottom Buttons:** "Table", "Raw Response", "Pivot Table", "Google Chart", and "Download as".
- Message Bar:** "Showing results from 1 to 80 of 80. Query took 0.2s, moments ago."
- Bottom Right:** "Keyboard shortcuts".

# SPARQL検索：可視化

- 検索結果のチャート表示も可能

Unnamed × Unnamed × 述語数カウント × +

```
PREFIX kgc: <http://kgc.knowledge-graph.jp/ontology/kgc.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
select ?label (count(?label) AS ?cnt) where {
  ?s kgc:hasPredicate ?o .
  ?o rdfs:label ?label .
} group by ?label order by desc(count(?label)) limit 20
```

Run Save

Table Raw Response Pivot Table Google Chart

Chart Editor

Start Charts Customize

Use 1st column as labels

Recommended charts - More »

label	cnt
say	~180
exist	~90
go	~70
see	~60
hear	~55
enter	~50
meet	~45
put	~40
open	~35
sit	~30

OK Cancel

# SPARQL検索：推論

- OWLやルールに基づく推論が可能

The screenshot shows the GraphDB interface with the following details:

- Left Sidebar:** Includes "Import", "Explore" (with "Graphs overview", "Class hierarchy", "Class relationships", "Visual graph", "Similarity"), "SPARQL" (selected), "Monitor", "Setup", and "Help".
- Top Bar:** Shows a dropdown for "LOD-WS-2020", tabs for "Editor only", "Editor and results" (highlighted in red), and "Results only", and a "Run" button.
- Query Editor:** A code editor titled "Unnamed" containing the following SPARQL query:

```
1 select * where {  
2   ?s ?p ?o .  
3 } limit 100  
4
```
- Right Panel:** A vertical sidebar with icons for saving, opening, linking, running, and expanding (highlighted with blue arrows). It also includes a "Run" button and a "Download as" dropdown.
- Bottom Panel:** A table showing the results of the query. The first row is highlighted with a blue border. The columns are labeled "s", "p", and "rdf:type". The "s" column contains "rdf:type", the "p" column contains "rdf:type", and the "rdf:type" column contains "rdf:Property".

Annotations on the right side of the interface:

- A blue arrow points to the "Run" icon in the sidebar with the text "推論機能ON" (Inference ON).
- A blue arrow points to the "owl:sameAs" icon in the sidebar with the text "owl:sameAsリンクの展開機能ON" (owl:sameAs link expansion function ON).

# SPARQL検索：推論（2）

SPARQL Query & Update ⓘ

Editor only Editor and results Results only

```
v 1 PREFIX ex: <http://example.com/terms>
v 2 select * where {
3   ?s a ex:哺乳類 .
4 }
```

Run

Press Alt+Enter to autocomplete

Table Raw Response Pivot Table Google Chart

Download as

No results. Query took 0.1s, moments ago.

Filter query results

s
No data available in table

推論OFF

SPARQL Query & Update ⓘ

Editor only Editor and results Results only

```
v 1 PREFIX ex: <http://example.com/terms>
v 2 select * where {
3   ?s a ex:哺乳類 .
4 }
```

Run

Press Alt+Enter to autocomplete

Table Raw Response Pivot Table Google Chart

Download as

Showing results from 1 to 1 of 1. Query took 0.1s, moments ago.

s
1 ex:Aさん

推論ON

# エクスポート

- リポジトリを好みの形式でエクスポート可能

The screenshot shows the GraphDB interface with the following details:

- Header:** GraphDB FREE, User: kgrc2021
- Left Sidebar:** Import, Explore, **Graphs overview** (highlighted in red), Class hierarchy, Class relationships, Visual graph, Similarity, SPARQL, Monitor, Setup, Help.
- Top Bar:** Search Graphs, Showing 1 - 9 of 9 results, Graphs per page: All.
- Main Content:** Export repository dropdown menu open, showing options: JSON, JSON-LD, RDF-XML, N3, N-Triples, N-Quads, Turtle, Turtle\*, TriX, TriG, TriG\*, Binary RDF. Each option has a URL below it: /kgc.knowledge-graph.jp/data/AbbeyGrange, /kgc.knowledge-graph.jp/data/SpecckledBand, /kgc.knowledge-graph.jp/data/DevilsFoot, /kgc.knowledge-graph.jp/data/ACaseOfIdentity, /kgc.knowledge-graph.jp/data/SilverBlaze, and http://kgc.knowledge-graph.jp/data/DancingMen. Each row also has a search icon and a delete icon.

# RDF利用プログラミング

# RDF操作ライブラリ

- Java
  - Apache Jena : <https://jena.apache.org/>
  - RDF4J : <https://rdf4j.org/>
- Python
  - RDFLib : <https://github.com/RDFLib/rdflib>
  - Owlready2 : <https://owlready2.readthedocs.io/en/latest/>
- PHP
  - EASYRDF : <https://www.easyrdf.org/>
- JS
  - RDFJS : <https://rdf.js.org/>
- Ruby
  - Ruby RDF : <https://github.com/ruby-rdf>
- C
  - Redland : <http://librdf.org/>
- Rust
  - Sophia : [https://github.com/pchampin/sophia\\_rs](https://github.com/pchampin/sophia_rs)

# Pythonライブラリ「RDFlib」の利用

- ・今回はPythonライブラリ「RDFlib」でRDFの利用方法を紹介
- ・Javaライブラリ「Jena」の利用方法は下記をご参照ください
  - ・「Javaのプログラムを作成してLODを効率的に処理してみよう  
<https://github.com/KnowledgeGraphJapan/LODws2nd/blob/master/ApacheJena%E3%83%8F%E3%83%B3%E3%82%BA%E3%82%AA%E3%83%B3.pdf>
- ・事前準備
  - ・どちらかを選択
    - ・Googleアカウントを用意しGoogle Colaboratoryを使用
    - ・Jupyter Notebookのインストールと実行環境の用意
- ・資料
  - ・<https://tinyurl.com/5xf8v6mv>

# Google Colaboratoryを利用

- 個人のドライブにコピーしてください

ここから実演します

The screenshot shows the Google Colaboratory interface with a notebook titled "kgrc-ws-2021-sample.ipynb". The left sidebar displays a file menu with options like "ドライブで探す", "Playground モードで開く", and "ノートブックを新規作成". The main area contains two code cells. The first cell shows the output of a command to install the Python library "RDFLib" using pip. The second cell shows the output of a command to install the "SPARQLWrapper" library. Both installations were successful.

```
pythonライブラリ「RDFLib」によるRDF利用プログラミング  
ノートブックは一般公開イベント  
ツイッグラフ推論チャレンジ2021「技術勉強会」～ナレッジグラフ利用技術の基礎と推論/説明への活用例～  
いて「ナレッジグラフ／L O D利用技術の入門」の説明に使用されるPythonチュートリアルです。
```

### 準備】必要なライブラリのインストール

```
!pip install rdflib  
Collecting rdflib  
  Downloading rdflib-6.0.0-py3-none-any.whl (376 kB)  
    ██████████ | 376 kB 4.5 MB/s  
Requirement already satisfied: pyparsing in /usr/local/lib/python3.7/dist-packages (from rdflib) (2.4.7)  
Collecting isodate  
  Downloading isodate-0.6.0-py2.py3-none-any.whl (45 kB)  
    ██████████ | 45 kB 3.4 MB/s  
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (from rdflib) (57.4.0)  
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from isodate->rdflib) (1.15.0)  
Installing collected packages: isodate, rdflib  
Successfully installed isodate-0.6.0 rdflib-6.0.0
```

```
!pip install SPARQLWrapper  
Collecting SPARQLWrapper  
  Downloading SPARQLWrapper-1.8.5-py3-none-any.whl (26 kB)  
Requirement already satisfied: rdflib>=4.0 in /usr/local/lib/python3.7/dist-packages (from SPARQLWrapper) (6.0.0)  
Requirement already satisfied: isodate in /usr/local/lib/python3.7/dist-packages (from rdflib>=4.0->SPARQLWrapper) (1.15.0)  
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (from rdflib>=4.0->SPARQLWrapper) (57.4.0)
```

# 今回説明しなかった関連技術

時間の都合上説明できなかつたので、より詳しく知りたい方は下記の資料をご覧ください

- RDFの細かな部分
  - RDF入門：<http://www.asahi-net.or.jp/~ax2s-kmtn/internet/rdf/rdf-primer.html>
  - RDF 1.1入門：<http://www.asahi-net.or.jp/~ax2s-kmtn/internet/rdf/NOTE-rdf11-primer-20140225.html>
- RDFスキーマ
  - RDFスキーマ -- リソース表現の語彙定義：<https://kanzaki.com/docs/sw/rdf-schema.html>
- オントロジー
  - オントロジーとは？：<https://www.slideshare.net/KoujiKozaki/ss-124638776>
- OWL
  - ウェブ・オントロジー言語OWL：<https://kanzaki.com/docs/sw/webont-owl.html>
- 記述論理
  - セマンティックWebと記述論理：<http://www.sw.cei.uec.ac.jp/kaneiwa/SemWeb.pdf>
- その他の推論
  - Reasoning: <http://graphdb.ontotext.com/documentation/standard/reasoning.html>
- SHACL
  - <https://www.w3.org/TR/shacl/>