

Representação e Processamento de Conhecimento (RPC)

1. Instalar a biblioteca “RDFLib” (já inclui o SPARQL)

Para instalar a biblioteca RDFLib (versão 4.2.2) e o Python (v3.6.4, testada) siga as indicações em:

- `_instalar_Python_distribute_RDFLib.txt`.

Para testar a instalação do interpretador SPARQL execute `a01_SPARQL.py` em `x_testInstall`. Se no fim a execução apresentar “ok-1, ok-2, ok-3, ok-4, ok-5” a instalação ficou correta.

Sobre documentação adicional Python:

- `aByteOfPython.pdf`; `OReilly_Learning_Python_3rd_Edition_Oct_2007.pdf`.

Sobre documentação adicional RDFLib:

- Exemplos usando a “Python Shell” (<alt>p e <alt>n navegam nos comandos executados):
 - o `import rdflib` #importar a biblioteca
 - o `dir(rdflib.Graph)` #conteúdo da classe Graph
 - o `help(rdflib.Graph)` #comentários escritos no código de Graph
 - o `help(rdflib.Graph.parse)` #comentários em Graph.parse()
 - o `help(rdflib.Graph.query)` #comentários em Graph.query()
- Mais informação sobre estas funções na documentação Python e também nas páginas 52-55 de `oEssencialDaLinguagemPython.pdf`.
- Se necessário recorra ao código fonte disponível na pasta: `rdflib-4.2.2\rdflib\`

O ambiente de desenvolvimento IDLE que está incluído na distribuição do Python é suficiente para a utilização nas aulas práticas. O ambiente é simples e permite que a atenção se foque nos exercícios (e não em compreender o próprio ambiente de desenvolvimento). Posteriormente poderá usar um ambiente mais completo, e.g., o “plug-in” Eclipse PyDev (<http://pydev.org/>). Fora do contexto Eclipse há várias referências a Spyder (<https://github.com/spyder-ide/spyder>).

2. Iniciar uma descrição RDF (grafo)

Considere o vocabulário FOAF (“Friend Of A Friend”) especificado em “`z_FOAFspec_v0_99.pdf`”. Veja também o contexto do projeto FOAF em “`http://xmlns.com/foaf/spec/`”.

- Desenhe, no papel, um grafo com três pessoas: `eu`, `amigoA` e `amigoB`. Cada pessoa é do tipo (`rdf:type`) `FOAF Person` e tem um nome (ver que propriedade FOAF utilizar).
- Estenda o grafo anterior indicando que a pessoa `eu` conhece e é conhecida pelas outras duas e que existe uma pessoa que é conhecida pelas outras mas que não conhece uma delas; a propriedade FOAF `knows` é usada para descrever asserções sobre “quem conhece quem”.
- Estenda o grafo anterior indicando para a pessoa `eu`: o endereço de e-mail e o cognome (“nickname”); para um amigo o seu endereço de e-mail (ver que propriedades FOAF utilizar).

Representação e Processamento de Conhecimento (RPC)

3. ... uma “novidade”

O projeto “VOWL: *Visual Notation for OWL Ontologies*” tem estado, ao longo dos últimos tempos, em evolução (cf., <http://vowl.visualdataweb.org/>). Atualmente, este projeto já disponibiliza alguns conteúdos interessantes. Embora ainda não tenhamos “tratado” o tema do OWL (e noção de ontologia) podemos começar já a usufruir dos conteúdos oferecidos neste projeto (VOWL).

- a) Tem especial interesse (para esta aula) a visualização gráfica (e interativa) do FOAF disponível em: <http://www.visualdataweb.de/webvowl/>. Explore esta informação navegando graficamente e compare com a informação textual disponível na especificação.
- b) Para entender melhor a notação gráfica aceda a <http://vowl.visualdataweb.org/v2/> e note a informação apresentadas nas tabelas do documento. Grande parte desta informação será posteriormente detalhada; no entanto podemos ficar já com uma ideia dos vários aspetos.

4. Construir um grafo e associar um “namespace” aos recursos RDF

Atenção: o código disponível foi testado na versão **Python 3.6.4**.

- a) Analise o ficheiro “foaf.rdf” gerado pela execução de `a01_criarFOAF.py` e altere o código para o “namespace” associado ao vocabulário FOAF ser “`http://xmlns.com/foaf/0.1/`”.

Copie o ficheiro “foaf.rdf” obtido para “foaf_a.rdf” (vai precisar dele posteriormente).

5. Construir um sujeito com BNode ou qualificador[valor]

Leia com atenção a função `adicionarPessoa` em “`a01_criarFOAF.py`”; BNode é “Blank Node”.

- a) Altere aquela função de modo a que o “namespace” “`http://meuMS/#`” fique associado a cada nova pessoa (`idPessoa`) que é construída. Note que deve alterar as linhas indicadas com `>ALTERAR<` em: i) bloco onde estão definidos os “namespace”, e ii) função `adicionarPessoa`.
- b) Analise o novo ficheiro “foaf.rdf” e compare com o anterior “foaf_a.rdf” (este novo ficheiro deve conter “`rdf:about="http://meuMS/#eu">`”, e não só!).
- c) Copie o ficheiro “foaf.rdf” obtido para “foaf_b.rdf” (pode precisar dele posteriormente).

6. Construir um triplo <sujeito, predicado, objecto>

Leia com atenção a função `s_knows_o` em “`a01_criarFOAF.py`”.

- a) Altere as linhas indicadas com `>ALTERAR<` de modo a que a função construa e devolva um triplo `<s, FOAF:knows, o>` onde `s` e `o` são, respectivamente, sujeito e objecto (parâmetros formais).
- b) Apenas conseguirá testar as alterações depois de terminar o próximo exercício; por isso avance!

Representação e Processamento de Conhecimento (RPC)
--

7. Usar uma ontologia já existente e descrever novos indivíduos

Volte a olhar para a folha de papel onde desenhou o grafo (cf. questão 2).

- Em “a01_criarFOAF.py” altere as linhas indicadas com >ALTERAR< nos blocos comentados com “Criar uma pessoa: eu”, “Criar um amigo: amigoA” e “Criar um amigo: amigoB”. As alterações devem transformar o grafo desenhado no papel em código que usa a RDFLib.
- Analise o novo ficheiro “foaf.rdf” e confirme que a questão anterior ficou bem implementada..
- Copie o ficheiro “foaf.rdf” obtido para “foaf_c.rdf” (será que precisa dele posteriormente?).

8. Percorrer o grafo à procura de relações

Recorde que o grafo é um conjunto de triplos $\langle s, p, o \rangle$ e analise, em “a01_criarFOAF.py”, o ciclo que itera em todos os s (sujeitos).

- Altere as linhas indicadas com >ALTERAR< de modo a: “apresentar, para cada pessoa (i.e., sujeito do tipo `FOAF:Person`) o seu nome, `FOAF:name`, e valor da propriedade “`FOAF:mbox`”.

9. Construir representações RDF (“serialize”) XML, N3 e Triples

Analise as diversas formas de construir representações RDF percorrendo as últimas linhas de código do “a01_criarFOAF.py”.

- Garanta que alguma das suas representações RDF tem exactamente a mesma estrutura que em “foaf_FINAL.rdf” (o texto pode ter outra sequência mas tem o mesmo padrão de construção).
- Em “a01_criarFOAF.py” altere o ponto “Serializar RDF com N-Triples” para gerar um ficheiro “foaf.nt” com a serialização do grafo em N-Triples.
- Em “a01_criarFOAF.py” altere o ponto “Serializar RDF com N3” para gerar um ficheiro “foaf.n3” com a serialização do grafo em N-3.

10. Reconstruir o grafo a partir de uma representação RDF/XML

Leia com atenção “a02_interrogarRDF_grafo.py”.

- Analise a função `construirGrafo` e perceba como se constrói um grafo a partir de uma descrição RDF. Note que, por omissão, a função “`parse`” trata o formato XML (posteriormente se verá como tratar os restantes formatos).
- Adicione manualmente informação ao seu ficheiro “foaf.rdf” (e.g., mais um endereço email) e utilize `construirGrafo` para construir o grafo RDF que daí resulta.

Representação e Processamento de Conhecimento (RPC)
--

11. Reconstruir o grafo a partir de qualquer representação RDF

Considere que: `Graph().parse(nomeDoFicheiro, format=formatoRDF)`, onde `formatoRDF` pode tomar os valores `"xml"`, `"n3"` ou `"nt"` constrói um grafo representado, respectivamente, por XML, "notation 3" ou "N-Triples" (lido a partir de `nomeDoFicheiro`).

- Altere a função `construirGrafo` de modo a receber como parâmetro o formato da representação RDF de modo a transformar qualquer representação num grafo.
- Altere a invocação da função `"construirGrafo"` (em `"a02_interrogarRDF_grafo.py"`) de modo a usar o grafo representado no ficheiro `"foaf.nt"`; volte a executar `"a02_interrogarRDF_grafo.py"`.
- Repita o exercício da alínea anterior mas agora usando o ficheiro `"foaf.n3"`.
- Com os exercícios das alíneas anteriores note que `obterConhecimentoMutuo` devolve a mesma informação independentemente do formato adoptado para representar a estrutura RDF.

12. Construir percursos sobre o grafo RDF

Leia com atenção a função `obterConhecimentoMutuo` em `"a02_interrogarRDF_grafo.py"`.

- Pretende-se percorrer o grafo de conhecidos (cf. questão 2) de modo a obter apenas aqueles que se conhecem mutuamente. Para isso altere em `obterConhecimentoMutuo`, as linhas indicadas com `>ALTERAR<`. Note que deve também fazer um ajuste em `apresentar_FOAFname`.