

## RDF(S) – Quick Intro

An example triple:

MarlonBrando actsIn TheGodfather .

S P O

Each element of a triple has to be uniquely identified – it has to have its URI (Uniform Resource Identifier)

PREFIX ex: <http://ai.fon.bg.ac.rs/osnovne/inteligentni-sistemi/vocabularies/example/>

ex:MarlonBrando ex:actsIn ex:TheGodfather .

Defining the type of a resource:

ex:MarlonBrando rdf:type ex:Actor . # rdf stands for RDF namespace

ex:TheGodfather rdf:type ex:Movie .

ex:Actor rdf:type rdfs:Class . # rdfs stands for RDF Schema namespace

ex:Movie rdf:type rdfs:Class .

Defining a subclass relation:

ex:Actor rdfs:subClassOf ex:Person .

ex:Movie rdfs:subClassOf ex:Artwork .

# subclass based inferences:

ex:MarlonBrando rdf:type ex:Actor . => ex:MarlonBrando rdf:type ex:Person .

Defining domain and range of a property:

ex:actsIn rdfs:domain ex:Actor ;

rdfs:range ex:Movie .

# domain- and range-based inferences, e.g.,

X ex:actsIn Y .

#X and Y are unknown resources; based on the above defined domain and range, the following inferences would be made:

X rdf:type ex:Actor . # also X rdf:type ex:Person .

Y rdf:type ex:Movie . # also Y rdf:type ex:Artwork .

Defining subproperty relation:

```
ex:principleActorIn rdfs:subPropertyOf ex:actsIn .
```

# subproperty-based inferences:

```
ex:AlPacino ex:principleActorIn ex:TheGodfather .
```

=>

```
ex:AlPacino ex:actsIn ex:TheGodfather
```

Additional triples to build a graph:

```
ex:TheGodfather owl:sameAs dbr:The_Godfather . # dbr: stands for http://dbpedia.org/resource/
```

```
dbr:The_Godfather dbo:director dbr:Francis_Ford_Coppola . # dbo stands for http://dbpedia.org/ontology/
```

```
dbr:Francis_Ford_Coppola dbo:child dbr:Sofia_Coppola .
```

```
dbr:Sofia_Coppola dbo:director dbr:Lost_in_Translation_(film).
```

```
dbr:Sofia_Coppola dbo:birthPlace dbr:New_York_City .
```

```
dbr:New_York_City owl:sameAs wikidata:wiki/Q60 #wikidata: stands for http://wikidata.org/
```