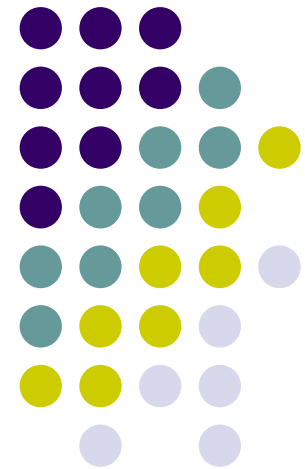


# SPARQL ([sparkle])

---

Dotazovací jazyk pro RDF





# Lehké zopakování RDF

- RDF = datový model = graf
- struktura RDF
  - triple = trojice (statement = tvrzení)
  - **?subject ?predicate ?object**
    - **?predicate** přiřazuje **?subject(u)** hodnotu **?object**
    - **?subject** je URI nebo Blank Node
    - **?predicate** je URI
    - **?object** je URI, Blank Node nebo Literál
  - URI (Uniform Resource Identifier)
    - globálně unikátní identifikátor
    - podmnožinou je URL (Uniform Resource Locator)
  - Blank Node
    - lokální identifikátor platný jen v rámci RDF grafu
  - Literál



# Více k literálům (1)

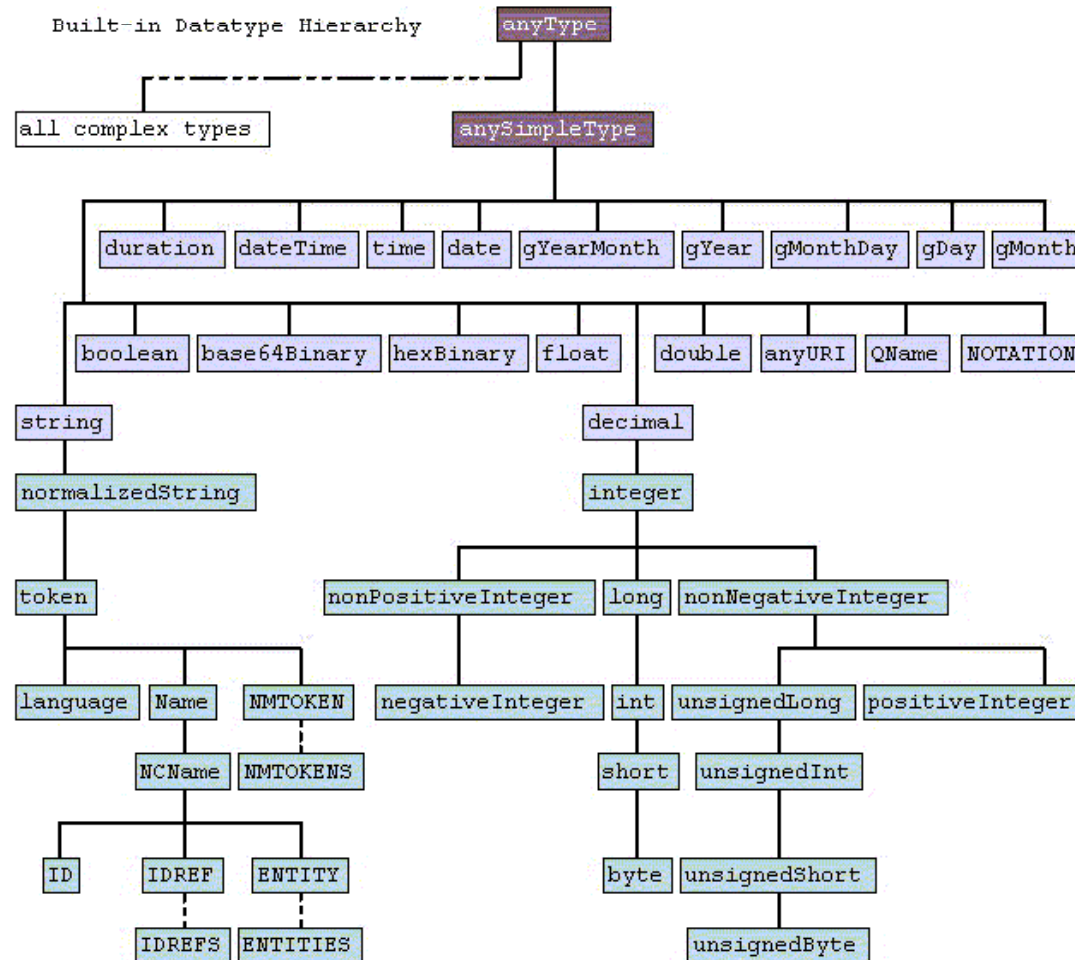
- bude se hodit při tvorbě SPARQL dotazů
- literály jsou **textové hodnoty**
- těmito hodnotami mohou být
  - řetězce
    - „Tim Berners-Lee“
  - čísla
    - „123456“
  - data
    - „2002-05-30T09:00:00“
  - atd.

# Více k literálům (2)



- literály se dělí na 2 skupiny
  - **plain** literals
    - čistě textové hodnoty
      - „čistě textová hodnota“
    - může být uveden jazyk (ve kterém je hodnota uvedena)
      - „čistě textová hodnota“@cs
      - syntaxe viz. dále
  - **typed** literals
    - specifikuje přesný typ literálu
      - syntaxe viz. dále
    - např. int, double, date, dateTime, atd.
      - k tomuto účelu se využívají **built-in datatypes** z XML

# XML built-in datatypes





# Serializace RDF

- RDF má spoustu serializací (RDF/XML, N3, N-Triples, Turtle, TriX, TriG, RDFa, eRDF)
  - některé slouží k přenosu a uchovávání RDF dat
    - **RDF/XML**
    - **TriX** (RDF Triples in XML)
      - speciální XML serializace pro Named Graphs („pojmenované grafy“)
        - Named Graphs viz. dále
  - jiné byly vytvořeny s ohledem na čitelnost
    - **N3 (Notation 3)**
      - „**non-XML serialization** of RDF models“, W3C standard
      - podmnožinou je **Turtle** (Terse RDF Triple Language)
        - nejčitejnější RDF serializace
        - není standardizován

# Turtle



# Popis RDF zdroje v RDF/XML



<?xml version="1.0"?>

- `<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:ex="http://example.org/stuff/1.0/">`
- `<rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar" dc:title="RDF/XML Syntax Specification (Revised)">`
  - `<ex:editor>`
    - `<rdf:Description ex:fullName="Dave Beckett">`
      - `<ex:homePage rdf:resource="http://purl.org/net/dajobe/" />`
    - `</rdf:Description>`
  - `</ex:editor>`
- `</rdf:Description>`
- `</rdf:RDF>`

# Týž popis RDF zdroje v Turtle



- @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
- @prefix dc: <http://purl.org/dc/elements/1.1/>.
- @prefix ex: <http://example.org/stuff/1.0/> .
- <http://www.w3.org/TR/rdf-syntax-grammar>
  - dc:title "RDF/XML Syntax Specification (Revised)" ;
  - ex:editor [ ex:fullname "Dave Beckett";  
ex:homePage http://purl.org/net/dajobe/  
] .



# SPARQL a Turtle

- **SPARQL** syntaxe vychází ze syntaxe **Turtle**
- SPARQL dotaz
  - PREFIX p: <<http://dbpedia.org/property/>>  
PREFIX dbpedia: <<http://dbpedia.org/resource/>>

```
SELECT ?movie
WHERE
{
  ?movie    p:wordnet_type
            <http://www.w3.org/2006/03/wn/wn20/instances/synset-movie-
            noun-1> .
  ?movie    p:starring      dbpedia:Angelina_Jolie .
  ?movie    p:released      ?released .
  FILTER (?released > "2005-01-01"^^xsd:date)
}
```



# Named Graphs (a TriG)

- Named Graph
  - tzv. „pojmenovaný graf“
    - RDF graf, který má přiřazenu URI
    - tímto jménem se lze na RDF graf odkazovat
- TriG
  - rozšiřuje Turtle syntaxi o možnost pojmenovat graf
  - ```
<http://mynamedgraph/> {  
  [vlastní RDF graf]  
}
```



# TriG - příklad

- @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .  
@prefix ex: <http://www.example.org/vocabulary#> .  
@prefix dc: <http://purl.org/dc/elements/1.1/>  
@prefix : <http://www.example.org/exampleDocument#> .
- **:G1** {  
:Monica ex:name "Monica Murphy" .  
:Monica ex:homepage <http://www.monicamurphy.org> .  
:Monica ex:email <mailto:monica@monicamurphy.org> .  
:Monica ex:hasSkill ex:Management  
}
- **:G2** {  
:Monica rdf:type ex:Person .  
:Monica ex:hasSkill ex:Programming  
}
- **:G1** dc:title „Monica’s Profile“@en



# SPARQL (1)

- SPARQL Protocol and RDF Query Language
- od 15. ledna 2008 W3C standard
- dotazovací jazyk pro RDF
- „read-only“
  - SPARUL – SPARQL Update Language
- jiné dotazovací jazyky pro RDF
  - (předchůdci SPARQL)
  - rdfDB, RDQL and SeRQL



## SPARQL (2)

- „SPARQL is to the Semantic Web (and, really, the Web in general) what SQL is to relational databases.“
  - Tim Berners-Lee
- „If we view the Semantic Web as a global collection of databases, SPARQL can make the collection look like one big database.“



# SPARQL (3)

- SPARQL je založen na W3C standardech
  - RDF
  - XML
  - HTTP
  - (WSDL, SOAP)



# SPARQL - dotaz

- <http://www.sparql.org/sparql.html>
- RDF Validator / **Converter**
  - <http://www.rdfabout.com/demo/validator/>
- PREFIX foaf: <http://xmlns.com/foaf/0.1/>  
SELECT ?url  
FROM  
<http://journal.dajobe.org/journal/2003/07/semblogs/  
bloggers.rdf>  
WHERE {  
    ?contributor foaf:name "Tim Berners-Lee" .  
    ?contributor foaf:weblog ?url  
}



# SPARQL - dotaz

- proměnné se uvozují „?“
  - **?variable**
  - (alternativně lze proměnnou uvést „\$“)
- klauzule **FROM** říká nad kterými RDF daty se bude dotaz vyhodnocovat
- klauzuli **WHERE** tvoří soubor RDF trojic (v Turtle syntaxi), který dohromady tvoří tzv. „**graph pattern**“
- vyhodnocování dotazu probíhá tak, že se porovnává **graph pattern** s RDF daty
- nejdříve je do proměnné ?contributor uložen Blank Node identifikátor, jemuž atribut **foaf:name** přiřazuje hodnotu „Tim Berners-Lee“ (TimBL)
- následně je do proměnné ?url uložena hodnota atributu **foaf:weblog**, která získanému Blank Node (označujícímu TimBL) přiřazuje URL jeho weblogu



# Typy SPARQL dotazů

- SPARQL podporuje 4 typy dotazů
  - SELECT
  - ASK
  - DESCRIBE
  - CONSTRUCT
- rozdíl mezi nimi spočívá v tom, jaké vrátí výsledky

# SPARQL (SELECT)

- SELECT zvoleným proměnným přiřadí příslušné hodnoty a vrátí je v tabulce



# SPARQL (SELECT) – RDF data



- @prefix foaf: <http://xmlns.com/foaf/0.1/> .  
\_:a foaf:name "Alice" .  
\_:a foaf:knows \_:b .  
\_:a foaf:knows \_:c .  
\_:b foaf:name "Bob" .  
\_:c foaf:name "Clare" .  
\_:c foaf:nick "CT" .



# SPARQL (SELECT) – dotaz

- PREFIX foaf: <http://xmlns.com/foaf/0.1/>  
SELECT ?nameX ?nameY ?nickY  
WHERE {  
    ?x foaf:knows ?y ;  
        foaf:name ?nameX .  
    ?y foaf:name ?nameY .  
OPTIONAL {  
    ?y foaf:nick ?nickY  
}  
}

# SPARQL (SELECT) – výsledek



- | <b>nameX</b> | <b>nameY</b> | <b>nickY</b> |
|--------------|--------------|--------------|
| "Alice,,     | "Bob,,       |              |
| "Alice,,     | "Clare,,     | "CT"         |

# SPARQL (SELECT) – výsledek (XML)



- ```
<?xml version="1.0"?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
<head>
  <variable name="nameX"/>
  <variable name="nameY"/>
  <variable name="nickY"/>
</head>
<results>
  <result>
    <binding name="nameX">
      <literal>Alice</literal>
    </binding>
    <binding name="nameY">
      <literal>Bob</literal>
    </binding>
  </result>
  <result>
    <binding name="nameX">
      <literal>Alice</literal>
    </binding>
    <binding name="nameY">
      <literal>Clare</literal>
    </binding>
    <binding name="nickY">
      <literal>CT</literal>
    </binding>
  </result>
</results>
</sparql>
```

# SPARQL (ASK)



- vrací boolean
- testuje pouze, zda má **graph pattern** řešení

# SPARQL (ASK) – RDF data



- @prefix foaf: <http://xmlns.com/foaf/0.1/> .  
\_:a foaf:name "Alice" .  
\_:a foaf:homepage  
<http://work.example.org/alice/> .  
\_:b foaf:name "Bob" .  
\_:b foaf:mbox <mailto:bob@work.example> .



# SPARQL (ASK) – dotaz

- PREFIX foaf: <http://xmlns.com/foaf/0.1/>  
ASK {  
    ?x foaf:name „Alice“  
}

# SPARQL (ASK) – výsledek



- yes
  - :-)

# SPARQL (ASK) – výsledek (XML)



- `<?xml version="1.0"?>`  
`<sparql`  
`xmlns="http://www.w3.org/2005/sparql-`  
`results#">`  
`<head>`  
`</head>`  
`<results>`  
`<boolean>>true</boolean>`  
`</results>`  
`</sparql>`



# SPARQL (DESCRIBE)(1)

- vrací graf (který je subgrafem dotazoveného grafu) se všemi informacemi o zdroji, který vyhověl **graph pattern(u)**
- **DESCRIBE** ?person  
**WHERE** {  
    ?person foaf:name "Tim Berners-Lee"  
}
- vrátí graf obsahující trojice, které popisují Tima Bernerse-Leeho



# SPARQL (DESCRIBE)(2)

- popis zdroje lze získat i explicitním zadáním jeho URI
  - **DESCRIBE** <http://example.org/>
- dotaz může obsahovat více URI, Blank Nodes
  - PREFIX foaf: <http://xmlns.com/foaf/0.1/>  
DESCRIBE ?x ?y <http://example.org/>  
WHERE {  
    ?x foaf:knows ?y  
}



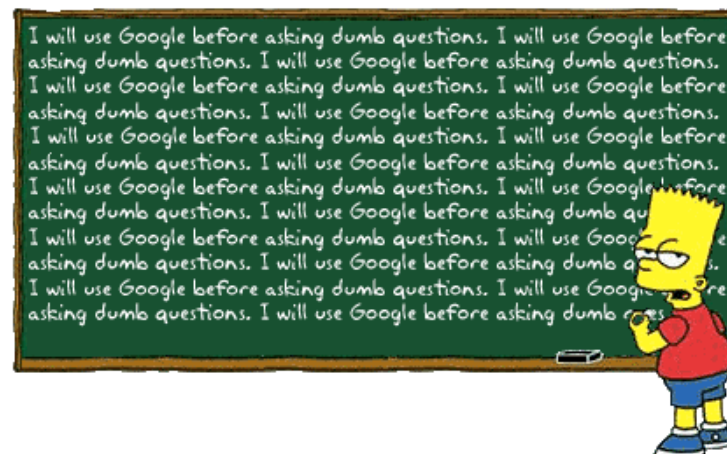
# SPARQL (DESCRIBE)(3)

- Dotaz
  - **DESCRIBE** <[http://dbpedia.org/resource/J.\\_R.\\_R.\\_Tolkien](http://dbpedia.org/resource/J._R._R._Tolkien)>
- Výsledek (v Turtle syntaxi)
  - @prefix rdf: <<http://www.w3.org/1999/02/22-rdf-syntax-ns#>>.  
@prefix xsd: <<http://www.w3.org/2001/XMLSchema#>>.  
@prefix dc: <<http://purl.org/dc/elements/1.1/>>.  
@prefix foaf: <<http://xmlns.com/foaf/0.1/>>.  
@prefix dbpedia: <<http://dbpedia.org/resource/>>.  
@prefix dbpedia-owl: <<http://dbpedia.org/ontology/>>.  
dbpedia:J.\_R.\_R.\_Tolkien  
rdf:type dbpedia-owl:Writer,  
foaf:name "J. R. R. Tolkien"@en;  
foaf:givenName "J. R. R."@en;  
foaf:surname "Tolkien"@en;  
dc:description "British philologist and author"@en;  
dbpedia-owl:birthDate "1892-01-03"^^xsd:date;  
dbpedia-owl:birthPlace dbpedia:Bloemfontein,



# SPARQL a Bart Simpson

- <http://dbpedia.org/snorql/>
- PREFIX dbpedia2: <http://dbpedia.org/property/>  
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>  
SELECT ?episode ?chalkboard\_gag  
WHERE {  
    ?episode skos:subject  
    <http://dbpedia.org/resource/Category:The\_Simpsons\_episodes%2C\_season\_12>.  
    ?episode dbpedia2:blackboard ?chalkboard\_gag  
}



# SPARQL (CONSTRUCT)



- CONSTRUCT umožňuje naplnit hodnotami přiřazenými proměnným z **graph pattern** vlastní „triple **template**“ a tím vytvořit zbrusu nový graf
- CONSTRUCT je pro SPARQL do určité míry tím, čím jsou XSLT transformace pro XML

# SPARQL (CONSTRUCT) – RDF data



- @prefix foaf: <http://xmlns.com/foaf/0.1/> .  
@prefix alice: <http://alice.name/#>  
alice:me foaf:name "Alice" .  
alice:me foaf:mbox <mailto:alice@example.org> .

# SPARQL (CONSTRUCT) – dotaz



- PREFIX foaf: <http://xmlns.com/foaf/0.1/>  
PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>  
PREFIX alice: <http://alice.name/#>  
CONSTRUCT {  
    <http://example.org/person#Alice> vcard:FN ?name  
}  
WHERE {  
    alice:me foaf:name ?name  
}

# SPARQL (CONSTRUCT) – výsledek



- @prefix vcard: <http://www.w3.org/2001/vcard-rdf/3.0#> .  
<http://example.org/person#Alice> vcard:FN "Alice" .

# Komplexnější dotazy – RDF data



- @prefix foaf: <http://xmlns.com/foaf/0.1/> .  
\_ :a foaf:name "Jon Foobar" ;  
foaf:mbox <mailto:jon@foobar.xx> ;  
foaf:depiction <http://foobar.xx/2005/04/jon.jpg> .  
\_ :b foaf:name "A. N. O'Ther" ;  
foaf:mbox <mailto:a.n.other@example.net> ;  
foaf:depiction <http://example.net/photos/an-2005.jpg> .  
\_ :c foaf:name "Liz Somebody" ;  
foaf:mbox\_sha1sum  
"3f01fa9929df769aff173f57dec2fe0c2290aeea" .  
\_ :d foaf:name "M Benn" ;  
foaf:depiction <http://mbe.nn/pics/me.jpeg> .



# SPARQL (OPTIONAL)

- chceme získat jména **všech** osob společně s URL jejich fotografie (pokud je uvedena)
  - nevystačíme s obyčejným **graph pattern**(em) jako dosud
    - do výsledků by totiž nebyla zahrnuta „Liz Somebody“
- pro tyto případy existuje blok **OPTIONAL**
  - pokud je **graph pattern** v něm uvedený v dotazovaném grafu přítomen, bude jeho výsledek uveden v celkovém výsledku

# SPARQL (OPTIONAL) – dotaz



- PREFIX foaf: <http://xmlns.com/foaf/0.1/>  
SELECT ?name ?depiction  
WHERE {  
    ?person foaf:name ?name .  
    OPTIONAL {  
        ?person foaf:depiction ?depiction .  
    }  
}

# SPARQL (OPTIONAL) – výsledek



- **name**
  - "A. N. O'Ther"
  - "Jon Foobar"
  - "Liz Somebody"
  - "M Benn"
- **depiction**
  - <<http://example.net/photos/an-2005.jpg>>
  - <<http://foobar.xx/2005/04/jon.jpg>>
  - <<http://mbe.nn/pics/me.jpeg>>

# SPARQL dotazy s omezováním hodnot proměnných



- PREFIX p: <http://dbpedia.org/property/>  
PREFIX dbpedia: <http://dbpedia.org/resource/>

```
SELECT ?movie
WHERE
{
  ?movie    p:wordnet_type
            <http://www.w3.org/2006/03/wn/wn20/instances/synset-movie-noun-1> .
  ?movie    p:starring      dbpedia:Angelina_Jolie .
  ?movie    p:released      ?released .
  FILTER (?released > "2005-01-01"^^xsd:date)
}
```



# SPARQL endpoints

- v podstatě jde o Web Service
- SPARQL endpointy jsou identifikovány pomocí URL
  - na kterou se posílají SPARQL dotazy
    - ty jsou následně vyhodnoceny a výsledky odeslány zpět uživateli
- SPARQL definuje komunikační protokol pro distribuci dotazů a výsledků
- aktivní endpointy
  - <http://esw.w3.org/topic/SparqlEndpoints>

# Odkazy



- RDF data
  - <http://planetrdf.com/index.rdf>
  - <http://www.deri.ie/fileadmin/scripts/foaf.php?id=287>
  - <http://jspetrak.zapisky.info/profile.rdf>
  - <http://journal.dajobe.org/journal/2003/07/semblogs/bloggers.rdf>
- FOAF ontologie
  - <http://xmlns.com/foaf/spec/index.rdf>
- SPARQL cheatsheet
  - <http://www.dajobe.org/2005/04-sparql/SPARQLreference-1.7.pdf>
- SPARQL execution services
  - <http://www.sparql.org/sparql.html>
  - <http://dbpedia.org/sparql>
  - <http://dbpedia.org/snorql/>
- SPARQL validátor
  - <http://www.w3.org/2005/01/yacker/uploads/SPARQL>
- RDF validátory
  - <http://www.w3.org/RDF/Validator/>
  - <http://www.rdfabout.com/demo/validator/>



# Užitečné odkazy

- <http://dbpedia.org/sparql>
  - Do „Default Graph URI“ vložit adresu RDF souboru
  - V roletovém menu vybrat druhou možnost
    - „Retrieve remote RDF data for all missing source graphs“
- <http://dbpedia.org/snorql/>
  - Pro seznam prefixů