

# *Representação do Conhecimento*

Standards da Web Semântica

# Standards



- Na posse dos conhecimentos base da web semântica, existem algumas questões que permanecem:
  - Que predicados usar?
  - Que tipos de dados se encontram? (números? texto?)
  - Porque é que uma entidade aparece representada de uma determinada forma e não de outra?
  - Como é que outros poderão saber que representações foram usadas?
  - A melhor forma de guardar e partilhar dados semânticos será na forma de triplos separados por vírgulas?

# Standards



- Estas questões relacionam-se sobretudo com a gestão e a partilha dos grandes conjuntos de dados semânticos, que já existem e que ainda serão criados.
- A resposta encontra-se na adoção de standards que, partilhados por todos, permitem trazer uniformidade ao tratamento da informação na web semântica.

# Standards



- Standards base da Web Semântica:
  - URIs (*Uniform Resource Identifiers*) – identificadores globais únicos que permitem identificar uma entidade (coisa), com grau absoluto de certeza
  - RDF (*Resource Description Framework*) – norma formal para representar e partilhar dados semânticos
  - SPARQL (*SPARQL Protocol and RDF Query Language*) – linguagem standard de pesquisa para dados semânticos

*URI*  
*Uniform Resource Identifier*

Standard base da WS



# URIs



- Num grafo, cada nó (entidade) necessita de um identificador único, de modo a ser referenciado de forma consistente, através de todo o conjunto de triplos que descrevem as relações do grafo.
- Em exemplos anteriores, foi utilizado um conjunto de símbolos, como “ridley\_scott”, para servir de identificador da entidade com nome “Ridley Scott”.
- Contudo, esta forma não garante a identificação unívoca absoluta.

# URIs



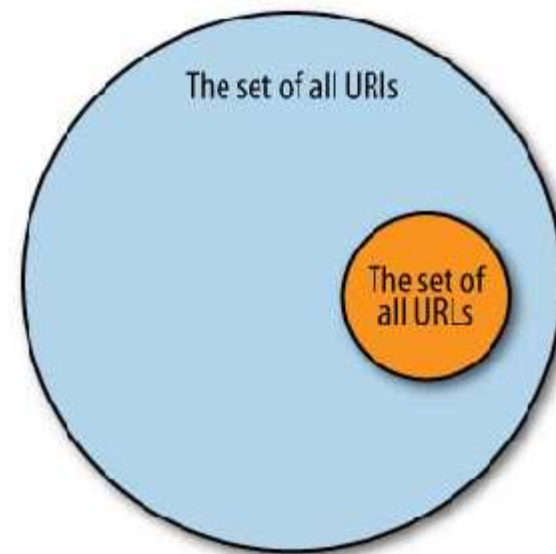
- Para evitar alguma possibilidade de ambiguidade, o standard RDF conceptualiza tudo no universo como um recurso (material ou imaterial).
- A qualquer recurso, sobre o qual seja possível falar, pode ser atribuído um URI.
- Assim, um URI identifica univocamente um recurso, qualquer que ele seja.

# URIs



- URIs e URLs

- URL (*Uniform Resource Locator*) – permite localizar um determinado recurso
  - Muito utilizado para especificar a localização e protocolo de acesso a um recurso na web
- Os URLs são um subconjunto dos URIs.





*RDF*  
*Resource Description Framework*

Standard base da WS



# RDF



- RDF é um modelo padrão para expressar dados semânticos, usando declarações na forma de triplos.
- Estes triplos possuem os elementos: sujeito – predicado – objeto.
  - sujeitos podem ser recursos;
  - objetos podem ser recursos;
  - predicados são sempre recursos.
- O URI de um recurso representado numa declaração RDF é chamado *URI Reference* (URIref).

# RDF



- Representação abreviada de URIs em RDF
  - Exemplo:  
referência do predicado “type”
    - URI total
      - `http://www.w3.org/1999/02/22-rdf-syntax-ns#type`
    - URI base
      - `rdf = http://www.w3.org/1999/02/22-rdf-syntax-ns#`
    - URI abreviada
      - `rdf:type`

# RDF

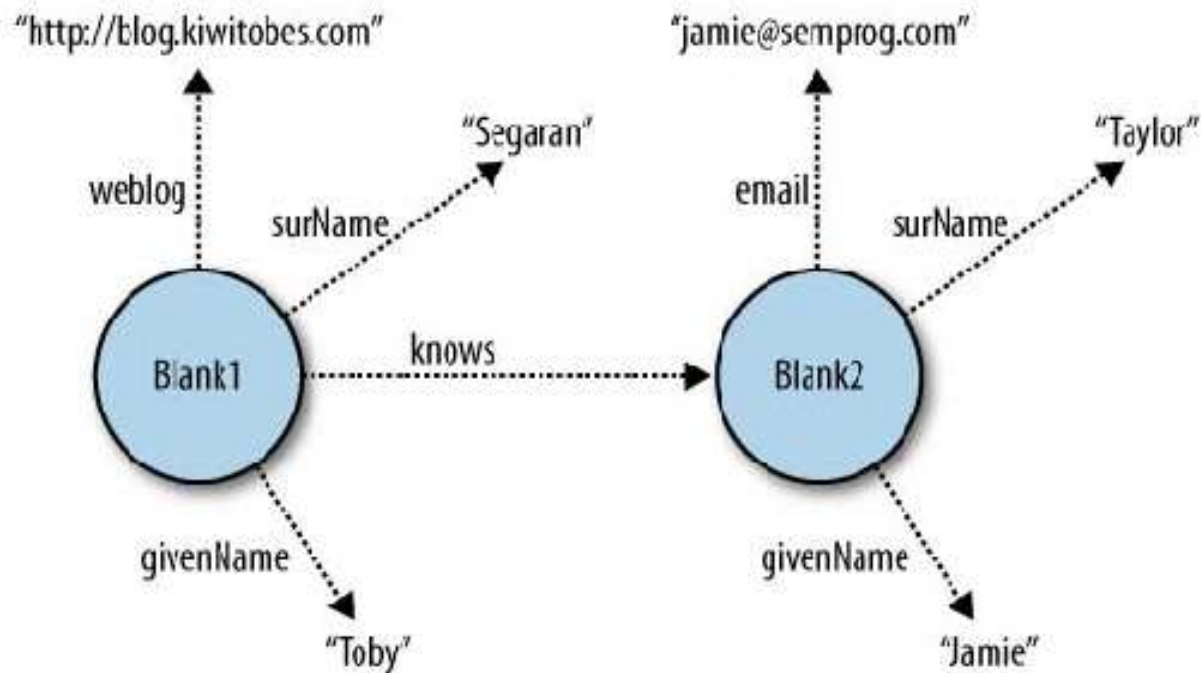


- Nós Anónimos (ou Vazios)
  - Atrás não foi dito que todos os sujeitos são recursos, porque existe a possibilidade de o seu URI ser desconhecido
  - Isto não quer dizer que não seja possível fazer declarações acerca desse sujeito
  - Para estas situações o RDF prevê a representação de nós anónimos ou vazios

# RDF



- Exemplo de um grafo com nós anónimos



# RDF



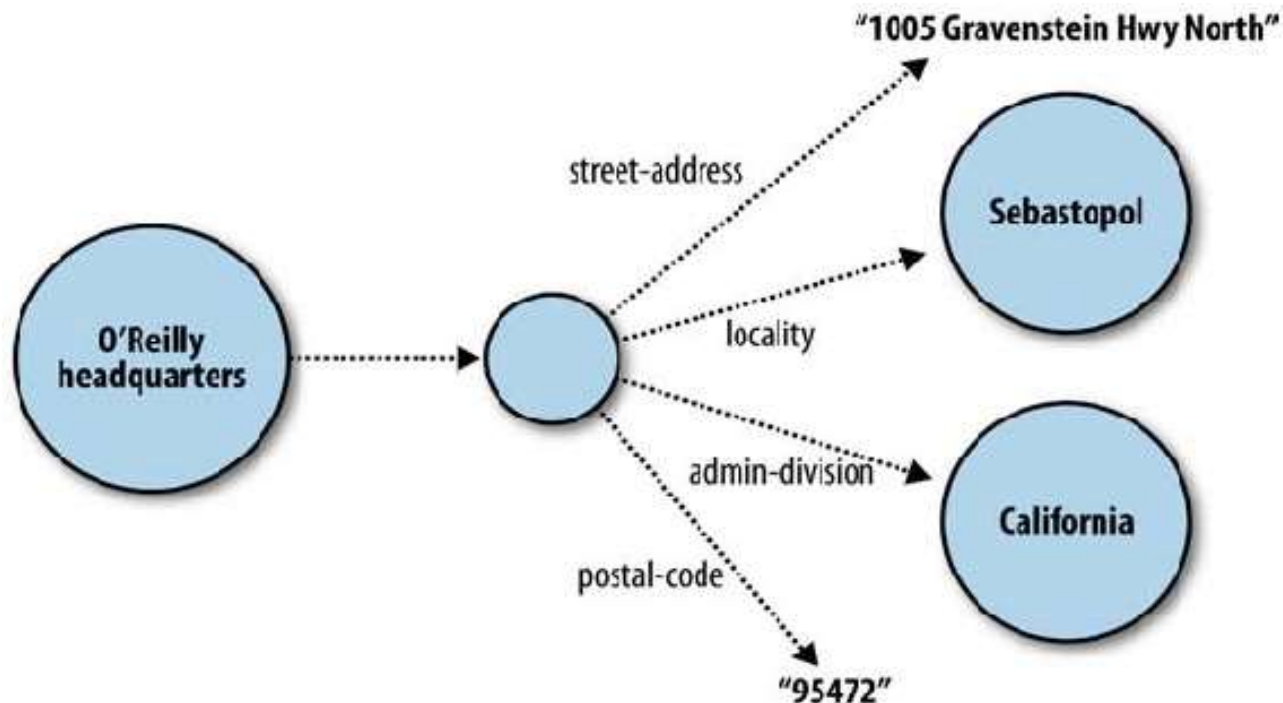
- No exemplo anterior, os nós Blank1 e Blank2 não possuem URIs, mas possuem Ids na forma:
  - “\_:id”
  - “id” é um identificador local
  - a maioria das bibliotecas RDF gera automaticamente um “id” interno que só é válido naquele grafo
- Triplos do exemplo anterior:

```
(_:ax1, "weblog", "http://blog.kiwitobes.com")  
(_:ax1, "surName", "Segaran")  
(_:ax1, "givenName", "Toby")  
(_:ax1, "knows", _:zb7)  
(_:zb7, "surName", "Taylor")  
(_:zb7, "givenName", "Jamie")  
(_:zb7, "email", "jamie@semprog.com")
```

# RDF



- Pode acontecer haver interesse na utilização de nós anónimos
- Exemplo:



# RDF



- Valores Literais

- Em RDF, os objetos também poderão tomar valores literais.
- Opcionalmente, pode associar-se uma língua (português, inglês, etc.) e/ou um tipo (integer, boolean, string, etc) ao valor.
  - para o tipo, usam-se os tipos especificados pelo XML Schema (ex: xsd:integer)
  - para a língua, usam-se os códigos ISO 639 (ex: pt, en)



*RDF*  
*Resource Description Framework*

A Sua Representação



# RDF - Representação



- A representação de RDF é algo para a qual existem atualmente várias bibliotecas *open source* e não oferece por isso muita preocupação.
- Contudo, é importante conhecer os formatos mais utilizados:
  - N-Triples – a mais simples;
  - N3 – uma compactação da N-Triples;
  - RDF/XML – uma das mais utilizadas;
  - RDFa – permite embutir RDF em páginas web.

# RDF - Representação



- Para estudar os formatos de representação, vamos usar um cenário de aplicação:
  - Um grafo de amigos
    - Uma modelação de parte da rede social do individuo chamado Toby
    - Neste grafo é incluída uma série de informação que ajudará a identificar o individuo de forma única.
    - Esta informação é a chamada informação de contexto que ajuda a identificar este Toby de muitos outros Tobys que existirão.
    - Isto levará também a obter um URI para o individuo, que poderá ser usado a partir daí.

# RDF - Exemplo

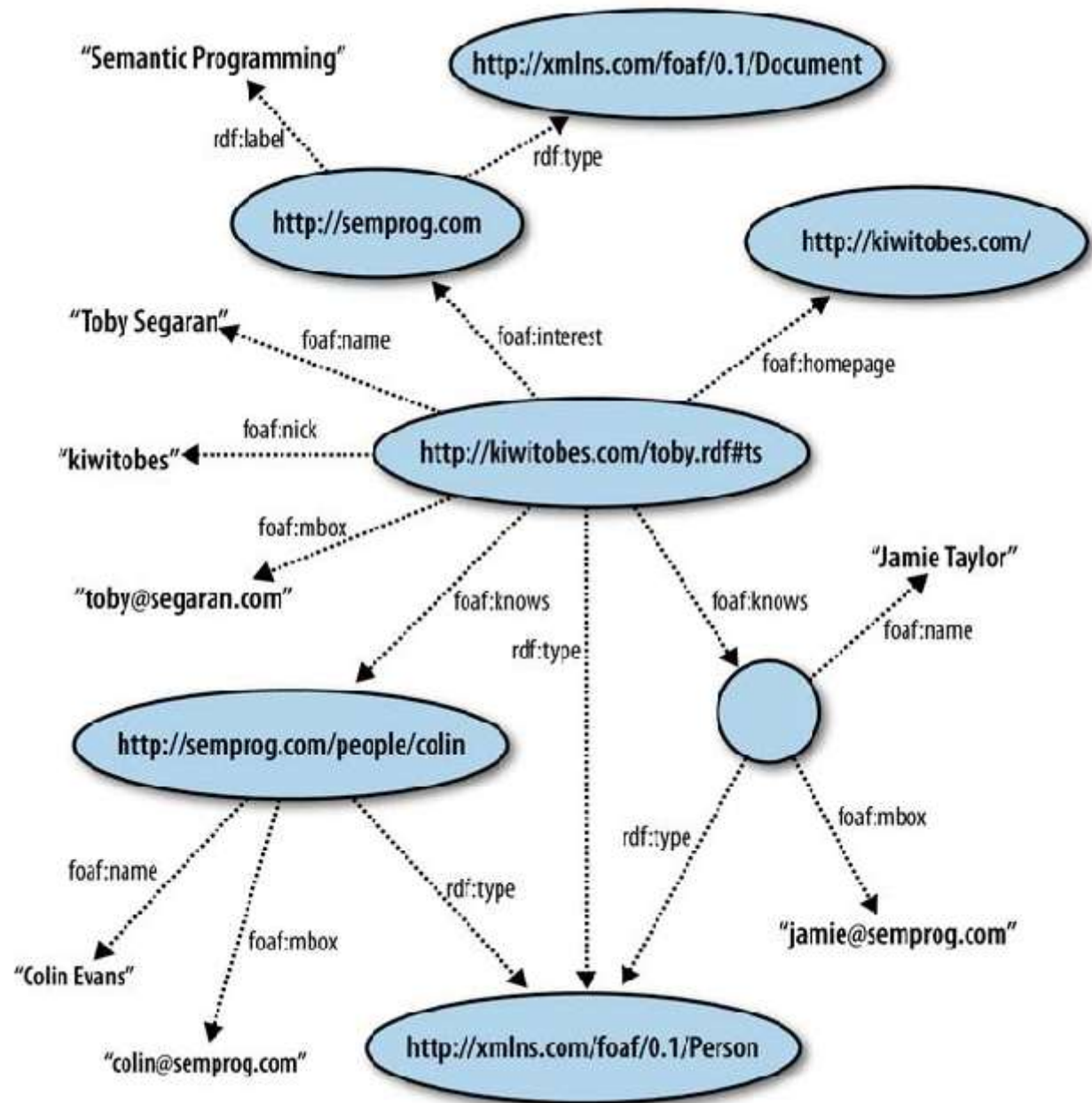


- FOAF (Friend of a Friend)
  - A expressão das redes de relações sociais que as pessoas têm entre si têm levado naturalmente ao surgimento de grafos em RDF.
  - Estes grafos sociais encontram-se entre os maiores conjuntos de dados disponíveis na internet.
  - Ao longo do tempo, as relações expressas neste grafos tornaram-se num conjunto de predicados bem conhecidos.
  - Este conjunto forma um vocabulário de expressão que se tornou conhecido por FOAF.
  - O FOAF tem sido adotado e estendido para descrever um número de coisas comuns, existentes na internet.

# RDF - Exemplo



- Rede social do Toby



*RDF*  
*Resource Description Framework*

Representação em N-Triples



# RDF – N-Triples



- A representação N-Triples é muito simples e aproxima-se muito do formato usado nos ficheiros CSV
- Devido à sua simplicidade, este formato é mais utilizado em situações de teste e depuração de conjuntos de dados em que há necessidade de intervenção manual.
- Cada linha neste formato, representa uma única declaração (triplo), contendo sempre:
  - sujeito, predicado e objeto
  - finalizada por ponto final

# RDF – N-Triples



- Os sujeitos, predicados e objetos são sempre expressos com URIs absolutos entre parênteses angulares “< >”
  - Com exceção para os nós anónimos, que aparecem na forma `_:name`, sendo name uma palavra alfanumérica começada por uma letra.
- Os objetos literais aparecem como strings entre aspas – ex: “Manuel Maria”
  - Opcionalmente, podem ter associado:
    - uma língua – ex: “Manuel Maria”@pt
    - ou um tipo – ex: “27”^^xsd:integer



# RDF – N-Triples



## • Exemplo

```
<http://kiwitobes.com/toby.rdf#ts> <http://xmlns.com/foaf/0.1/homepage>
<http://kiwitobes.com/>.
<http://kiwitobes.com/toby.rdf#ts> <http://xmlns.com/foaf/0.1/nick>
    "kiwitobes".
<http://kiwitobes.com/toby.rdf#ts> <http://xmlns.com/foaf/0.1/name>
    "Toby Segaran".
<http://kiwitobes.com/toby.rdf#ts> <http://xmlns.com/foaf/0.1/mbox>
<mailto:toby@segaran.com>.
<http://kiwitobes.com/toby.rdf#ts> <http://xmlns.com/foaf/0.1/interest>
<http://semprog.com>.
<http://kiwitobes.com/toby.rdf#ts> <http://www.w3.org/1999/02/22-rdf-syntax-
    ns#type>
<http://xmlns.com/foaf/0.1/Person>.
<http://kiwitobes.com/toby.rdf#ts> <http://xmlns.com/foaf/0.1/knows> _:jamie .
<http://kiwitobes.com/toby.rdf#ts> <http://xmlns.com/foaf/0.1/knows>
<http://semprog.com/people/colin>.
...
```

*RDF*  
*Resource Description Framework*

Representação em N3



# RDF – N3



- A representação N-Triples é muito simples mas também muito custosa:
  - Muita repetição ==> o que leva a maior tempo de transmissão e de processamento.
- A representação N3 consegue condensar muita da repetição imposta pela N-Triples.
- Neste formato é feito:
  - o uso de URIs prefixo para encurtar o URI de cada nó
  - um reaproveitamento dos nós que participam em múltiplos triplos.

# RDF – N3



- URIs prefixo – exemplo
  - usando no início:  
@prefix semp: <http://semprog.com/people/>.
  - pode-se substituir:  
<http://semprog.com/people/colin>
  - por:  
semp:colin

# RDF – N3



- Reaproveitamento de nós

- Sujeito

- Novos triplos, podem ser representados, adicionando duplos (predicado, objeto) separados por ponto e vírgula (;)
    - Exemplo:  
semp:colin foaf:knows <http://kiwitobes.com/toby.rdf#ts>;  
foaf:mbox "colin@semprog.com".

- Predicado

- Novos triplos, pode ser representados, adicionando apenas objetos, separados por vírgula (,)
    - Exemplo:  
semp:colin rdf:type foaf:person,  
dc:author.

# RDF – N3



- Abreviação
  - Entidades anónimas
    - Exemplo:  
<http://kiwitobes.com/toby.rdf#ts> foaf:knows [  
    <http://www.w3.org/2006/vcard/ns#street-address>  
        "1005 Gravenstein Hwy North" ;  
    <http://www.w3.org/2006/vcard/ns#locality>  
        "Sebastopol, California" ].
  - Predicados
    - Alguns predicados mais usados podem ser abreviados por
      - a   <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
      - =   <http://www.w3.org/2002/07/owl#sameAs>
      - => <http://www.w3.org/2000/10/swap/log#implies>
      - <= <http://www.w3.org/2000/10/swap/log#implies> (inverso)

# RDF – N3



## • Exemplo

```
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix semp: <http://semprog.com/people/>.
@prefix tobex: <http://kiwitobes.com/toby.rdf#>.

tobex:ts a foaf:Person;
    foaf:homepage <http://kiwitobes.com/>;
    foaf:interest <http://semprog.com>;
    foaf:knows semp:colin,
        [a foaf:Person;
            foaf:mbox <mailto:jamie@semprog.com>;
            foaf:name "Jamie Taylor"];
    foaf:mbox <mailto:toby@segaran.com>;
    foaf:name "Toby Segaran";
    foaf:nick "kiwitobes".

<http://semprog.com> a foaf:Document;
    rdfs:label "Semantic Programming".

semp:colin a foaf:Person;
    foaf:mbox <mailto:colin@semprog.com>;
    foaf:name "Colin Evans".
```

*RDF*  
*Resource Description Framework*

Representação em XML





# RDF – XML

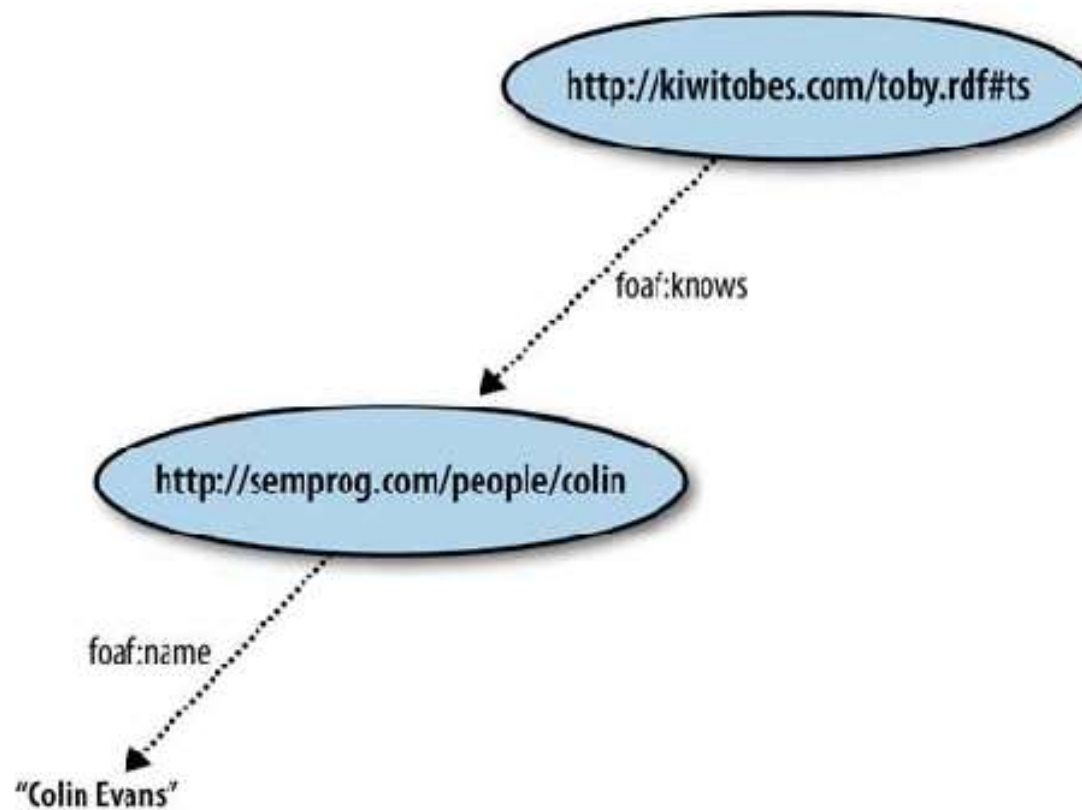


- A representação de RDF sobre XML foi originalmente usada pelo W3C na recomendação do RDF e é uma das mais usadas ainda atualmente.
- Isto, ao ponto de muitas pessoas confundirem RDF com RDF/XML.
- Contudo, é uma representação muito criticada devido à dificuldade de leitura que apresenta para o ser humano.
- Conceptualmente, a RDF/XML é feita de pequenas descrições, traçando cada uma delas um caminho através do grafo RDF.
- Quando existe mais que um caminho, o elemento raiz deve iniciar obrigatoriamente por:
  - `<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" />`

# RDF – XML



- Exemplo de um caminho



# RDF – XML



- Cada caminho inicia-se sempre com o elemento `<rdf:Description>`
- A referência URI é indicada no atributo “rdf:about”
- Para nós anónimos, sem URI, o ID é indicado no atributo “rdf:NodeID”
- Os predicados são especificados como elementos filhos do elemento `<rdf:Description>`

# RDF – XML



- O caminho do exemplo anterior pode ser dado como:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/">
  <rdf:Description rdf:About="http://kiwitobes.com/toby.rdf#ts">
    <foaf:knows>
      <rdf:Description rdf:About="http://semprog.com/people/colin">
        <foaf:name>Colin Evans</foaf:name>
      </rdf:Description>
    </foaf:knows>
  </rdf:Description>
</rdf:RDF>
```

# RDF – XML



- Os objetos literais podem ser representados como texto de um elemento ou como atributo do elemento `<rdf:Description>`

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/">
  <rdf:Description rdf:About="http://kiwitobes.com/toby.rdf#ts">
    <foaf:knows>
      <rdf:Description rdf:About="http://semprog.com/people/colin">
        <foaf:name>Colin Evans</foaf:name>
        <foaf:mbox>colin@semprog.com</foaf:mbox>
      </rdf:Description>
    </foaf:knows>
    <foaf:knows>
      <rdf:Description foaf:mbox="jamie@semprog.com"/>
    </foaf:knows>
  </rdf:Description>
</rdf:RDF>
```

# RDF – XML



- À imagem de outras representações, em XML também é possível condensar a escrita
- O predicado `rdf:type`, por exemplo, pode ser escrito de duas formas
  - `<rdf:Description rdf:about="http://www.kiwitobes.com/toby.rdf#ts">  
<rdf:type> <foaf:Person>`
  - Ou diretamente
  - `<foaf:Person  
rdf:about="http://kiwitobes.com/toby.rdf#ts">`

# RDF – XML



- Representação total do grafo

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
          xmlns:foaf="http://xmlns.com/foaf/0.1/"
          xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
```

```
  <foaf:Person rdf:about="http://kiwitobes.com/toby.rdf#ts">
    <foaf:name>Toby Segaran</foaf:name>
    <foaf:homepage rdf:resource="http://kiwitobes.com/" />
    <foaf:nick>kiwitobes</foaf:nick>
    <foaf:mbox rdf:resource="mailto:toby@segaran.com" />
```

```

    <foaf:interest>
      <foaf:Document rdf:about="http://semprog.com">
        <rdfs:label>Semantic Programming</rdfs:label>
      </foaf:Document>
    </foaf:interest>
```

...

# RDF – XML



- Representação total do grafo (cont.)

...

```
<foaf:knows>
  <foaf:Person rdf:about="http://semprog.com/people/colin">
    <foaf:name>Colin Evans</foaf:name>
    <foaf:mbox rdf:resource="mailto:colin@semprog.com"/>
  </foaf:Person>
</foaf:knows>
```

```
<foaf:knows>
  <foaf:Person>
    <foaf:name>Jamie Taylor</foaf:name>
    <foaf:mbox rdf:resource="mailto:jamie@semprog.com"/>
  </foaf:Person>
</foaf:knows>
```

```
</foaf:Person>
</rdf:RDF>
```