

Lehrveranstaltung

# Semantic Web Technologien

SS 2010

HTWG Konstanz



# RDF

Resource Description Framework

# Sprachen des Semantic Web – RDF

- XML verwendet Baumstrukturen zur Darstellung von Informationen
- Hervorragend geeignet zur Organisation hierarchischer Daten (Wie beispielsweise in Dokumenten)
- Baumstrukturen können gezielt durchsucht und verarbeitet werden

# Sprachen des Semantic Web – RDF

- Die Baumstruktur von XML hat aber auch Nachteile
  - Modellierung von Sachverhalten oft unklar / uneinheitlich
  - Beispiel:
    - Das Auto-Modell “Golf” wird von Volkswagen hergestellt:

```
<stelltHer>  
  <Hersteller>Volkswagen</Hersteller>  
  <Modell>Golf</Modell>  
</stelltHer>
```

```
<Hersteller name="Volkswagen">  
  <stelltHer Modell="Golf />  
</Hersteller>
```

```
<Modell>  
  <Name>Golf</Name>  
  <Hersteller>Volkswagen</Hersteller>  
</Modell>
```

# Sprachen des Semantic Web – RDF

- Daraus resultieren diverse Probleme:
  - Die Beschreibung der Daten ist **nicht intuitiv**
  - **Probleme beim Zusammenführen** von Daten aus mehreren Quellen (Datenintegration)
  - Die Informationen im WWW sind **dezentral organisiert** und folgen nicht unbedingt einer hierarchischen Struktur
- => Baumstruktur ist für das Semantic Web als Datenmodell nicht geeignet

# Sprachen des Semantic Web – RDF

- Das Semantic Web verwendet **Graphen** an Stelle von Bäumen als Datenmodell
  - Menge von Knoten, verbunden durch gerichtete Kanten/Pfeile
  - Sowohl Knoten als auch Kanten haben eindeutige Bezeichner

- Beispiel:



- Dieses Datenmodell wird als RDF (Resource Description Framework) bezeichnet



# Sprachen des Semantic Web – RDF

- RDF
  - Offizieller Standard des W3C  
Alle wichtigen Dokumente dazu unter:  
<http://www.w3.org/RDF/>
  - Zunächst nur zur Beschreibung von Metadaten über Webseiten gedacht
  - Codierung von strukturierten Informationen
  - Universell, maschinenlesbar, zum Austausch geeignet
  - Kann in diverse Formate “serialisiert” werden  
(zum Beispiel, aber nicht nur, XML – Dazu später mehr)

# Sprachen des Semantic Web – RDF

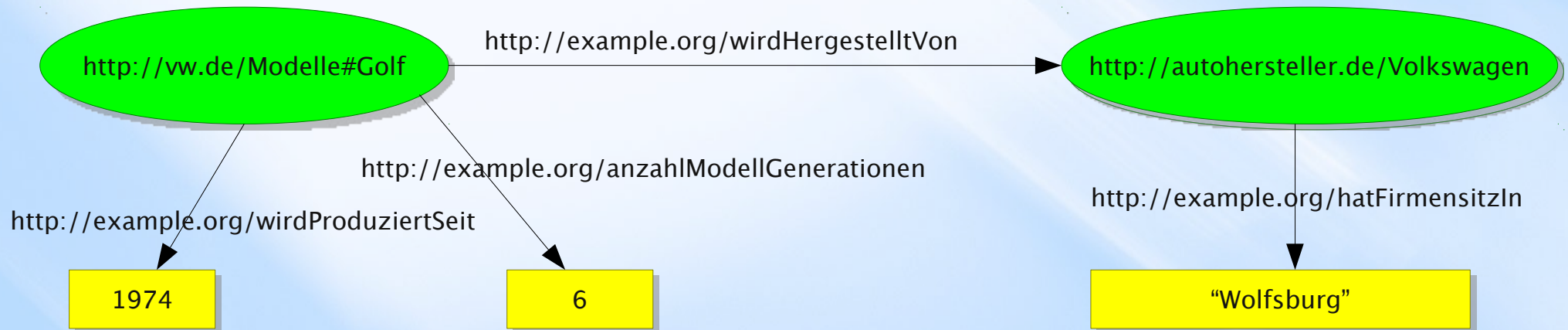
- Bestandteile von RDF
  - Ressourcen / Resources
    - Etwas, das beschrieben werden soll → Die Knoten
  - Literale / Literals
    - Datenwerte ohne separate Existenz
  - Eigenschaften / Properties
    - Die Beziehungen zwischen Ressourcen und anderen Ressourcen oder Literalen → gerichtete Kanten
  - URIs (Uniform Resource Identifiers)
    - Eindeutige Identifizierung von Ressourcen und Properties
  - Leere Knoten / Blank Nodes
    - Ressourcen, denen (u.U. noch) keine URI zugewiesen ist
    - Erlauben Aussagen über Dinge, die wir nicht genau identifizieren können oder wollen
    - Müssen bei der Verarbeitung besonders behandelt werden



# Sprachen des Semantic Web – RDF

- Literale

- Repräsentation von Datenwerten
- Darstellung von Zeichenketten
- Können einen Datentyp besitzen (siehe XSD)
- Wenn kein Datentyp angegeben wird, behandelt man Literale meist als Zeichenketten (ist aber nicht genauer definiert)



- Es können **keine Aussagen** über Literale getroffen werden!

# Sprachen des Semantic Web – RDF

- Probleme:
  - Darstellung als Grafik nur sinnvoll, solange der Graph recht klein ist
    - Praktisch bestehen Datensätze aber aus Tausenden oder Millionen von Kanten und Knoten
  - Zur Verarbeitung in Maschinen denkbar ungeeignet
  - Wie speichert man einen Graphen?
- Lösung
  - Zerlegung des Graphen in kleine Bestandteile
  - Überführung der Bestandteile in Zeichenketten
  - Nacheinander abspeichern

# Sprachen des Semantic Web – RDF

- Wie unterteilt man Graphen?
  - Grundsätzlich mehrere Ansätze möglich
  - Im Falle des Semantic Web wird ein Graph in Aussagen unterteilt
- → Bestandteile von RDF Graphen sind Aussagen/Statements in Form von Tripeln

# Sprachen des Semantic Web – RDF

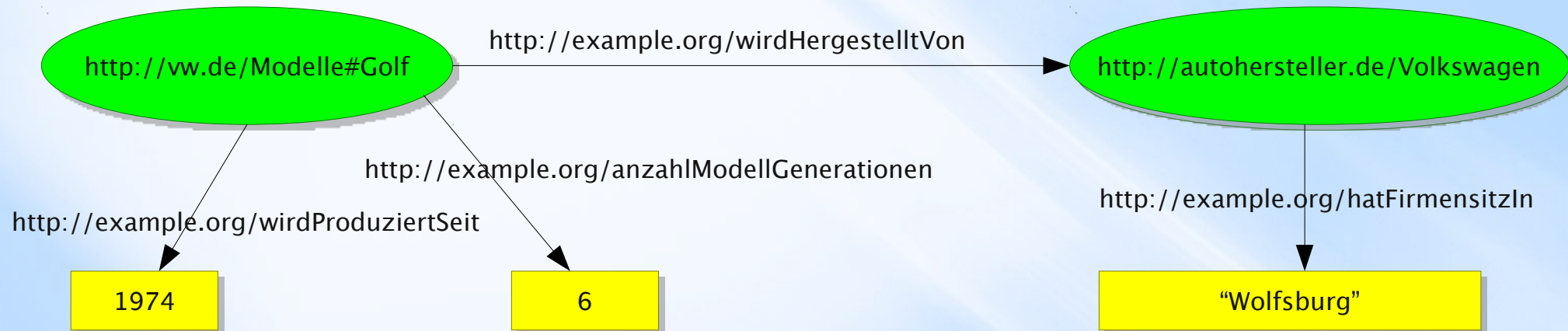
- RDF–Triple



- Ähnlich wie in der Linguistik
- Erlaubte Belegung:
  - Subjekt: URI oder leerer Knoten
  - Prädikat: URI
  - Objekt: URI, leerer Knoten oder Literal
- Durch eindeutige Bezeichnung von Knoten und Kanten kann ursprünglicher Graph wieder aus Liste von Tripeln rekonstruiert werden

# Sprachen des Semantic Web – RDF

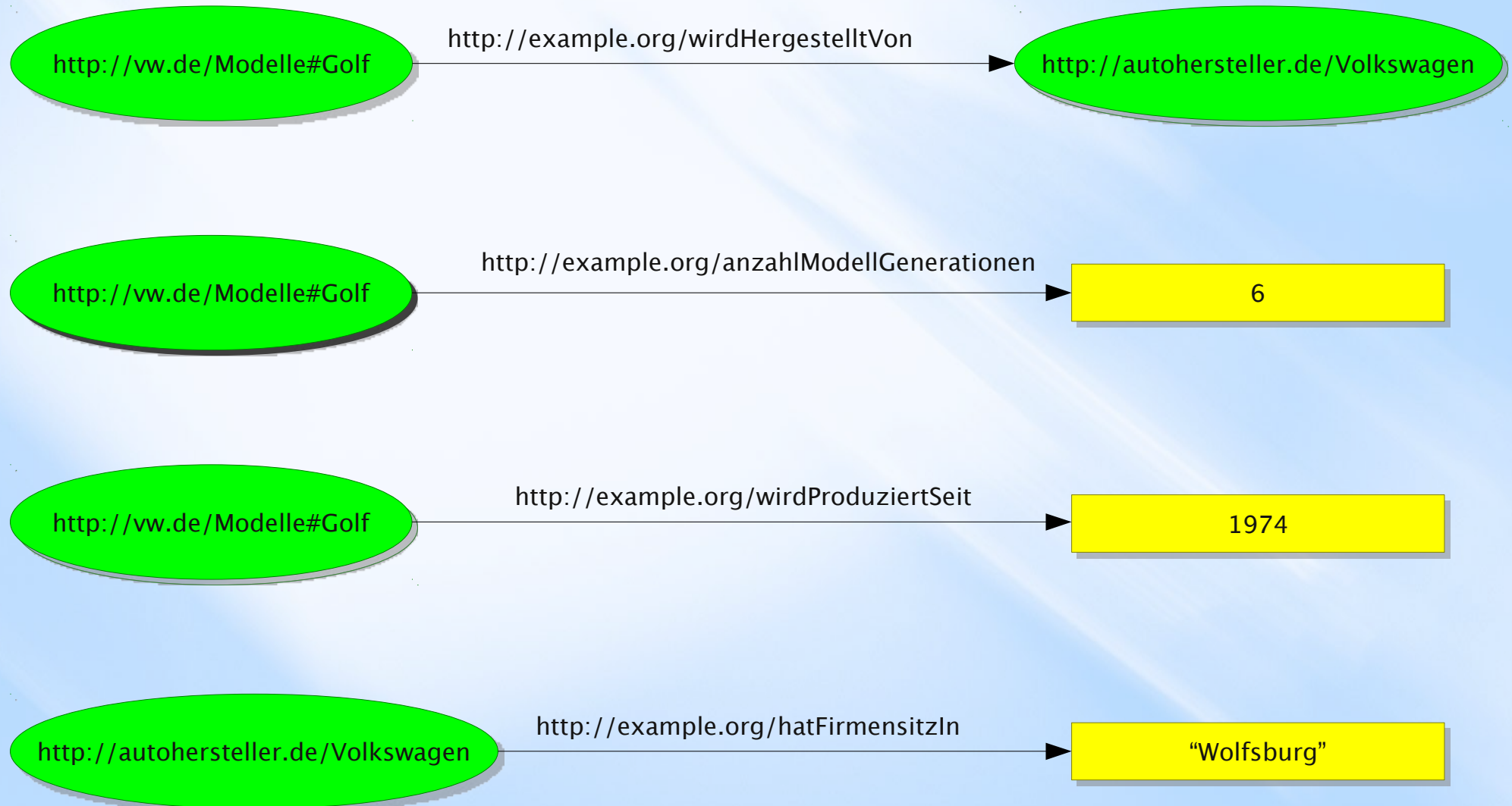
- Beispiel:





# Sprachen des Semantic Web – RDF

- Beispiel:



# Sprachen des Semantic Web – RDF

- Serialisierung von Tripeln
  - RDF beschränkt sich nicht auf ein einzelnes Format zur Serialisierung
  - N3 / N-Triples / Turtle
    - Simple Auflistung von Tripeln
  - RDF/XML
    - Serialisierung der Triple nach XML
  - RDFa (RDF in attributes)
    - Einbettung von Tripeln in (X)HTML
  - Und viele andere:
    - TriX
    - TriG
    - RDF/JSON
    - RDF/YAML ;-)
    - ...

# Sprachen des Semantic Web – RDF

- N3 / N-Triples / Turtle
  - Darstellung von RDF-Graphen als Ansammlung von Tripeln
  - N3 (Notation 3)
    - 1998 von Tim Berners-Lee vorgeschlagen
    - Beinhaltet auch komplexe Ausdrücke wie Pfade und Regeln
  - N-Triples
    - 2004 als offizielle Recommendation des W3C verabschiedet
    - Teilmenge von N3 mit weniger Komplexität
  - Turtle
    - N-Triples hatte zu wenig Funktionalität
      - Turtle erweitert N-Triples
        - z.B. Um eine Kurzschreibweise
  - N-Triples und Turtle sind im Wesentlichen Teile von N3
  - Sie beschränken sich allerdings auf zulässige RDF-Graphen

# Sprachen des Semantic Web – RDF

- Turtle-Syntax:
  - Triple werden nacheinander niedergeschrieben
  - URIs werden in spitze Klammern eingefasst
  - Literale werden in Anführungszeichen gestellt
  - Jede Aussage endet mit einem Punkt

- Beispiel:

```
<http://vw.de/Modelle#Golf> <http://example.org/wirdHergestelltVon>  
  <http://autohersteller.de/Volkswagen> .  
<http://vw.de/Modelle#Golf> <http://example.org/anzahlModellGenerationen> "6" .  
<http://vw.de/Modelle#Golf> <http://example.org/wirdProduziertSeit> "1974" .  
<http://autohersteller.de/Volkswagen> <http://example.org/hatFirmensitzIn>  
  "Wolfsburg" .
```

# Sprachen des Semantic Web – RDF

- Abkürzung von URIs durch Namespaces
  - Zur Vermeidung von Verwechslungen dürfen abgekürzte URIs nicht mehr in spitzen Klammern notiert werden
  - Es empfiehlt sich, verständliche Abkürzungen zu wählen
  - Ein Bezeichner “Präfix:Name” wird auch QName genannt
- Beispiel:

```
@prefix ex:          <http://example.org/>.
@prefix vwmodell:    <http://vw.de/Modelle#> .
@prefix hersteller: <http://autohersteller.de/> .
```

```
vwmodell:Golf          ex:wirdHergestelltVon      hersteller:Volkswagen .
vwmodell:Golf          ex:anzahlModellGenerationen  "6" .
vwmodell:Golf          ex:wirdProduziertSeit      "1974" .
hersteller:Volkswagen ex:hatFirmensitzIn      "Wolfsburg" .
```



# Sprachen des Semantic Web – RDF

- Zusammenfassung von Tripeln
  - Triple mit dem selben Subjekt können zusammengefasst werden
    - Subjekt muss dann nur einmal notiert werden
    - Aussagen werden durch Semikolon getrennt
  - Gibt es zu einem Subjekt und einem Prädikat mehrere Objekte, können diese durch Kommata getrennt notiert werden
- Beispiel:

```
@prefix ex:          <http://example.org/>.
@prefix vwmodell:    <http://vw.de/Modelle#> .
@prefix hersteller: <http://autohersteller.de/> .

vwmodell:Golf          ex:wirdHergestelltVon      hersteller:Volkswagen ;
                      ex:anzahlModellGenerationen  "6" ;
                      ex:wirdProduziertSeit        "1974" .
hersteller:Volkswagen ex:hatFirmensitzIn          "Wolfsburg" ;
                      ex:hatProduktionsstandort    "Wolfsburg",
                                                    "Dresden",
                                                    "Bratislava",
                                                    "Pamplona" .
```

# Sprachen des Semantic Web – RDF

- RDF/XML

- Turtle ist für Menschen und Maschinen leicht lesbar
- RDF wird trotzdem am häufigsten nach XML serialisiert
- Warum?
  - Fehlende Programmbibliotheken für Turtle
  - Jede Programmiersprache unterstützt XML
  - Oft auch: XML ist man bereits gewohnt!
- Gibt es keine Probleme mit den unterschiedlichen Datenmodellen (XML-Bäume vs. RDF-Graphen)?
  - NEIN: XML gibt lediglich syntaktische Struktur des Dokuments vor
  - ABER: XML ist hierarchisch aufgebaut
    - Kodierung von RDF-Triplen muss ebenfalls hierarchisch erfolgen

- RDF/XML Syntax:

- Wurzelement aller RDF/XML Dokument ist **rdf:RDF**  
( rdf = “http://www.w3.org/1999/02/22-rdf-syntax-ns#” )
- Ressourcen werden durch Elemente mit dem XML-Name **rdf:Description** beschrieben
  - Attribut “about” gibt die URI an
  - Mit Attribut “ID” kann einer Ressource innerhalb eines Dokuments eine feste ID zugewiesen werden, die von anderen Tripeln referenziert werden kann.
    - Eine ID darf nur einmal innerhalb eines Dokuments vorkommen
    - Der Wert von “ID” entspricht dem Fragment-Teil einer URI
    - “about” und “ID” dürfen nicht gleichzeitig bei einem Description-Element vorkommen
  - Empfehlung: Auf ID weitestgehend verzichten (Außer bei der Erstellung von Vokabularen)

# Sprachen des Semantic Web – RDF

- RDF/XML Beispiel:

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://example.org/"
  xmlns:vwmodell="http://vw.de/Modelle#"
  xmlns:hersteller="http://autohersteller.de/" >
  <rdf:Description rdf:about="http://vw.de/Modelle#Golf">
    <ex:wirdHergestelltVon>
      <rdf:Description rdf:about="http://autohersteller.de/Volkswagen" />
    </ex:wirdHergestelltVon>
  </rdf:Description>
  <rdf:Description rdf:about="http://vw.de/Modelle#Golf">
    <ex:anzahlModellGenerationen>6</ex:anzahlModellGeneration>
  </rdf:Description>
  <rdf:Description rdf:about="http://vw.de/Modelle#Golf">
    <ex:wirdProduziertSeit>1974</ex:wirdProduziertSeit>
  </rdf:Description>
  <rdf:Description rdf:about="http://autohersteller.de/Volkswagen">
    <ex:hatFirmensitzIn>Wolfsburg</ex:hatFirmensitzIn>
  </rdf:Description>
</rdf:RDF>
```



- RDF/XML Beispiel – Zusammenfassung:

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://example.org/"
  xmlns:vwmodell="http://vw.de/Modelle#"
  xmlns:hersteller="http://autohersteller.de/" >
  <rdf:Description rdf:about="http://vw.de/Modelle#Golf">
    <ex:wirdHergestelltVon>
      <rdf:Description rdf:about="http://autohersteller.de/Volkswagen" >
        <ex:hatFirmensitzIn>Wolfsburg</ex:hatFirmensitzIn>
      </rdf:Description>
    </ex:wirdHergestelltVon>
    <ex:anzahlModellGenerationen>6</ex:anzahlModellGeneration>
    <ex:wirdProduziertSeit>1974</ex:wirdProduziertSeit>
  </rdf:Description>
</rdf:RDF>
```



- RDF/XML Beispiel – Weitere Verkürzungen
  - Literale können als Attribute dargestellt werden
  - `rdf:resource` zur Referenzierung von Ressourcen

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://example.org/"
  xmlns:vwmodell="http://vw.de/Modelle#"
  xmlns:hersteller="http://autohersteller.de/" >
  <rdf:Description rdf:about="http://vw.de/Modelle#Golf"
    ex:anzahlModellGenerationen="6"
    ex:wirdProduziertSeit="1974" >
    <ex:wirdHergestelltVon rdf:resource="http://www.autohersteller.de/Volkswagen" />
  </rdf:Description>
  <rdf:Description rdf:about="http://www.autohersteller.de/Volkswagen"
    ex:hatFirmensitzIn="Wolfsburg" />
</rdf:RDF>
```

# Sprachen des Semantic Web – RDF

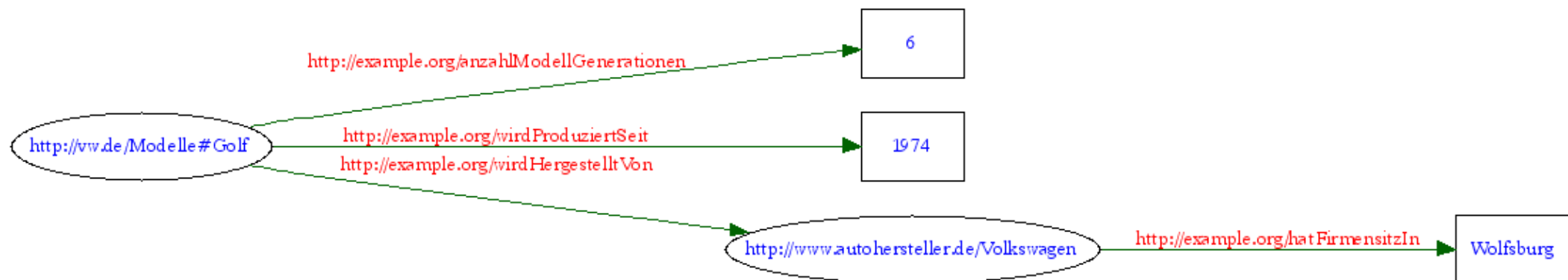
- ENTITYs zur Abkürzung von URIs

- Durch Definition von ENTITYs (siehe XML-Vorlesung) können lange Textstrings abgekürzt werden
- Dies können wir in Attributwerten verwenden

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE rdf:RDF [
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <!ENTITY vwmodell "http://vw.de/Modelle#">
  <!ENTITY hersteller "http://autohersteller.de/"> ]>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://example.org/"
  xmlns:vwmodell="http://vw.de/Modelle#"
  xmlns:hersteller="http://autohersteller.de/" >
  <rdf:Description rdf:about="&vwmodell;Golf">
    <ex:wirdHergestelltVon>
      <rdf:Description rdf:about="&hersteller;Volkswagen" >
        <ex:hatFirmensitzIn>Wolfsburg</ex:hatFirmensitzIn>
      </rdf:Description>
    </ex:wirdHergestelltVon>
  </rdf:Description>
</rdf:RDF>
```

# Sprachen des Semantic Web – RDF

- Darstellung von RDF in XML ist sehr vielfältig
  - RDF/XML Beispiele beschreiben sicher nicht den gleichen XML-Baum
  - Aber den gleichen RDF-Graphen!!!
- W3C-Validator für RDF-XML:
  - <http://www.w3.org/RDF/Validator>
  - Kontrolliert RDF/XML Dokumente auf Gültigkeit
  - Gibt in XML-Dokument enthaltene Triple aus
  - Kann eine grafische Darstellung des Graphs erstellen:



# Sprachen des Semantic Web – RDF

- Probleme mit RDF/XML

- URIs enthalten mindestens einen Doppelpunkt (Nach dem Schema)
  - Unzulässig in XML-Namen
    - Namensräume sind elementarer Bestandteil der Codierung
- Namensräume können nur für Element- und Attributnamen verwendet werden – nicht in Attributwerten
  - Selbe URI muss an verschiedenen Stellen des XML-Dokuments unterschiedlich notiert werden
- Diverse weitere Probleme
  - Direkt nach ':' (z.B. Bei Element-Namen) darf in XML kein '-' auftauchen
    - Obwohl in URIs ohne weiteres erlaubt → Definition von Hilfsnamensräumen
  - Unzulässige Sonderzeichen in URIs werden häufig mit % kodiert
    - %20 steht für Leerzeichen usw.
    - In Attributwerten unproblematisch – In Element-Namen ist % verboten

- Die meisten Probleme werden von RDF-Bibliotheken abgefangen



- XHTML+RDFa Beispiel

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:ex="http://example.org/"
      xmlns:vwmodell="http://vw.de/Modelle#"
      xmlns:hersteller="http://autohersteller.de/" >
<head>
  <title>XHTML+RDFa Beispiel</title>
</head>
<body>
  <div about="http://vw.de/Modelle#Golf">
    <h2 property="ex:ModellName">Golf</h2>
    <label>Aktuelle Modellgeneration:</label>
    <span property="ex:anzahlModellGenerationen">6</span>
    <label>Produktionsbeginn:</label>
    <span property="ex:wirdProduziertSeit">1974</span>
    <label>Hersteller:</label>
    <div rel="ex:wirdHergestelltVon"
          resource="http://autohersteller.de/Volkswagen" >
      <h3 property="ex:herstellerName">Volkswagen</h3>
      <label>Firmensitz:</label>
      <span property="ex:hatFirmensitzIn">Wolfsburg</span>
    </div>
  </div>
</body>
</html>
```



# Sprachen des Semantic Web – RDF

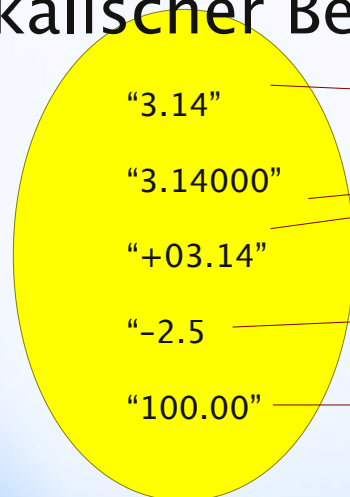
- TriX Beispiel:

```
<?xml-stylesheet type="text/xml" href="http://www.w3.org/2004/03/trix/all.xsl" ?>
<TriX xmlns="http://www.w3.org/2004/03/trix/trix-1/"
  xmlns:ex="http://example.org/"
  xmlns:vwmodell="http://vw.de/Modelle#"
  xmlns:hersteller="http://autohersteller.de/" >
  <graph>
    <uri>http://example.org/graph1</uri>
    <triple>
      <qname>vwmodell:Golf</qname>
      <qname>ex:wirdHergestelltVon</qname>
      <qname>hersteller:Volkswagen</qname>
    </triple>
    <triple>
      <qname>vwmodell:Golf</qname>
      <qname>ex:anzahlModellGenerationen</qname>
      <integer>6</integer>
    </triple>
    <triple>
      <qname>vwmodell:Golf</qname>
      <qname>ex:wirdProduziertSeit</qname>
      <integer>1974</integer>
    </triple>
    <triple>
      <qname>hersteller:Volkswagen</qname>
      <qname>ex:hatFirmensitzIn</qname>
      <plainLiteral>Wolfsburg</plainLiteral>
    </triple>
  </graph>
</TriX>
```

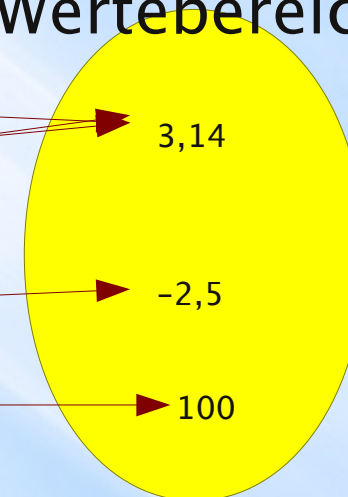
- Datentypen für Literale

- Definieren, wie Literale interpretiert werden sollen
- Beispiel: xsd:decimal

Lexikalischer Bereich



Wertebereich

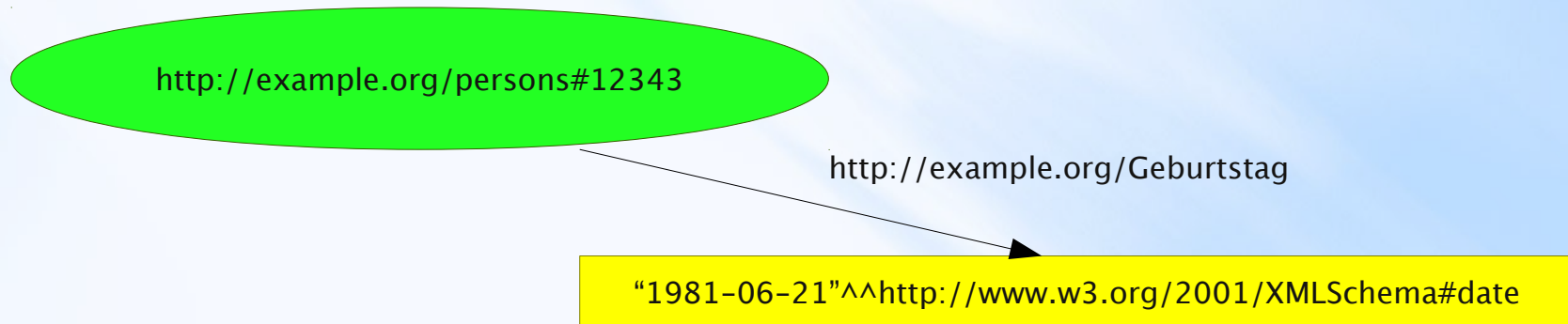


- Bei xsd:decimal gilt "3.14" = "+03.14"  
Bei xsd:String gilt das nicht!

- **Datentypen in RDF**
  - Bisher haben wir “ungetypte” Literale verwendet
    - Verwendung wie Zeichenketten
  - Typisierung erlaubt semantischen Umgang mit Werten
  - Datentypen werden identifiziert durch URIs
  - Datentypen sind frei wählbar
  - Datentypen können selbst bestimmt werden
  - Meistens verwendet man die xsd-Datentypen
- **Schreibweise von typisierten Attributwerten:**
  - “Datenwert”^^Datentyp-URI

# Sprachen des Semantic Web – RDF

- Datentypen-Beispiel:



- Turtle:

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .  
<http://example.org/persons#12343>  
  <http://example.org/Geburtstag>  
    "1981-06-21"^^xsd:date .
```

- XML

```
<rdf:Description rdf:about="http://example.org/persons#12343">  
  <ex:Geburtstag rdf:datatype="http://www.w3.org/2001/XMLSchema#date">  
    1981-06-21  
  </ex:Geburtstag>  
</rdf:Description>
```

# Sprachen des Semantic Web – RDF

- `rdf:XMLLiteral` wird von RDF vordefiniert
  - Bezeichnet (balancierte) XML-Fragmente
  - Hat spezielle Syntax in RDF/XML:

```
<rdf:Description rdf:about="&vwmodell;Golf">  
  <ex:Herstellername rdf:parseType="Literal">  
    <b>VW</b> – Volkswagen  
  </ex:Herstellername>  
</rdf:Description>
```



- Sprachangaben bei Datentypen

- Können nur bei ungetypten Literalen verwendet werden  
→ Entweder Datentyp oder Sprachangabe

- Beispiel:

```
<rdf:Description rdf:about="&vwmodell;Golf">  
  <ex:ModellName xml:lang="de">Golf</ex:ModellName>  
  <ex:ModellName xml:lang="en">Rabbit</ex:ModellName>  
</rdf:Description>
```

- Mehrwertige Beziehungen

- Kochen mit RDF:

“Für die Zubereitung von Chutney benötigt man 450g grüne Mango, einen Teelöffel Cayennepfeffer, ...”

- Erster Modellierungsversuch:

```
@prefix ex: <http://example.org> .  
ex:Chutney ex:hatZutat "450g grüne Mango",  
              "1TL Cayennepfeffer" .
```

- Nicht zufriedenstellend, da Zutaten und Menge in einer Zeichenkette vermischt sind

→ Suche nach Rezepten mit grüner Mango nicht möglich

- Mehrwertige Beziehungen

- Kochen mit RDF:

“Für die Zubereitung von Chutney benötigt man 450g grüne Mango, einen Teelöffel Cayennepfeffer, ...”

- Zweiter Modellierungsversuch:

```
@prefix ex: <http://example.org> .  
ex:Chutney ex:hatZutat ex:grueneMango;  
           ex:Menge    "450g" ;  
           ex:hatZutat ex:Cayennepfeffer;  
           ex:Menge    "1TL" .
```

- Noch schlimmer, da keine Zuordnung mehr zwischen Menge und konkreter Zutat mehr möglich ist

# Sprachen des Semantic Web – RDF

- **Problem:**

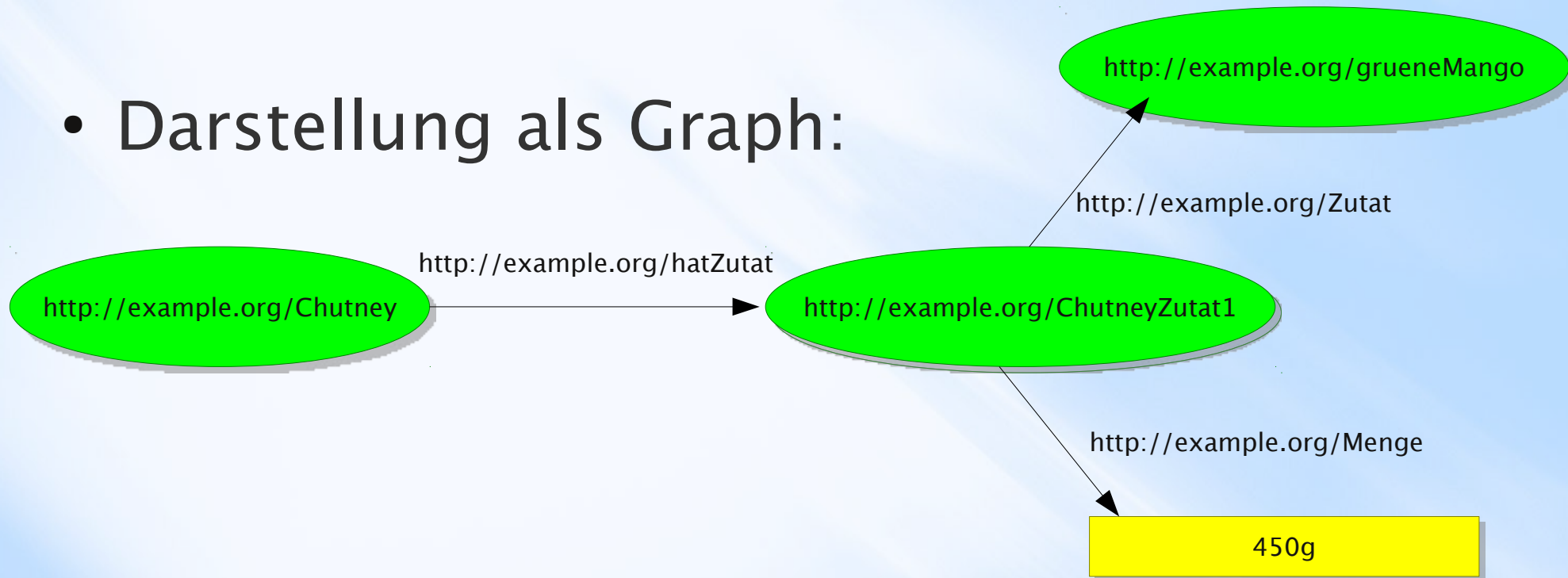
- Es handelt sich um eine echte dreiwertige (auch: ternäre) Beziehung:

Gericht	Zutat	Menge
Chutney	Grüne Mango	450g
Chutney	Cayennepfeffer	1TL

- Direkte Darstellung innerhalb von RDF nicht möglich
- Lösung: Einführung von Hilfsknoten

# Sprachen des Semantic Web – RDF

- Darstellung als Graph:



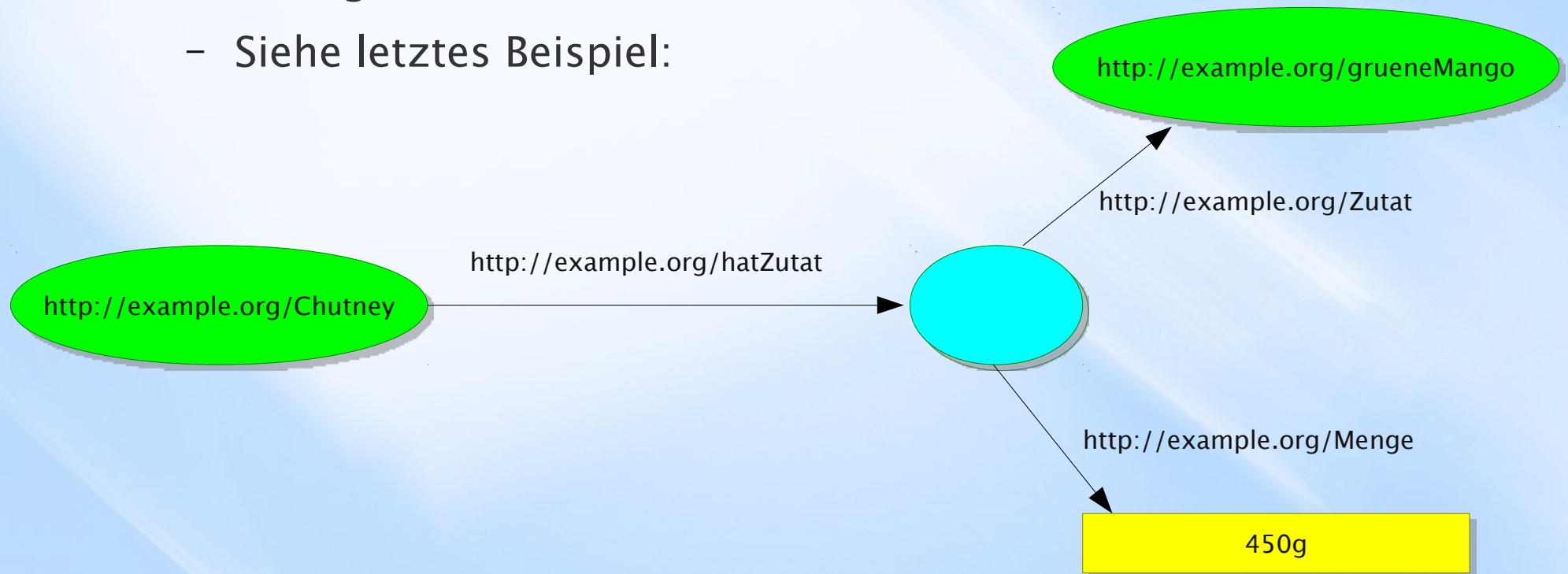
- Turtle Syntax (verwendet `rdf:value`):

```
@prefix ex:    <http://example.org> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
ex:Chutney    ex:hatZutat    ex:ChutneyZutat1 .
ex:ChutneyZutat1  rdf:value  ex:grueneMango ;
                  ex:Menge   "450g" .
```



- **Leere Knoten (Blank Nodes, Bnodes)**

- Verwendung für Ressourcen, die eigentlich keine eigene Identifizierung benötigen
- Häufig verwendet bei Hilfsknoten
- Siehe letztes Beispiel:



# Sprachen des Semantic Web – RDF

- Leere Knoten

- RDF/XML-Syntax

```
<rdf:Description rdf:about="&ex;Chutney">  
  <ex:hatZutat rdf:nodeID="id1" />  
</rdf:Description>  
<rdf:Description rdf:nodeID="id1">  
  <ex:Zutat rdf:resource="&ex;grueneMango" />  
  <ex:Menge>450g</ex:Menge>  
</rdf:Description>
```

- Verkürzt

```
<rdf:Description rdf:about="&ex;Chutney">  
  <ex:hatZutat rdf:nodeID="id1" >  
    <ex:Zutat rdf:resource="&ex;grueneMango" />  
    <ex:Menge>450g</ex:Menge>  
  </ex:hatZutat>  
</rdf:Description>
```

# Sprachen des Semantic Web – RDF

- Leere Knoten

- Turtle-Syntax

```
@prefix ex: <http://example.org> .  
ex:Chutney ex:hatZutat _:idl .  
_:idl      ex:Zutat      ex:grueneMango ;  
           ex:Menge      "450g" .
```

- Verkürzt

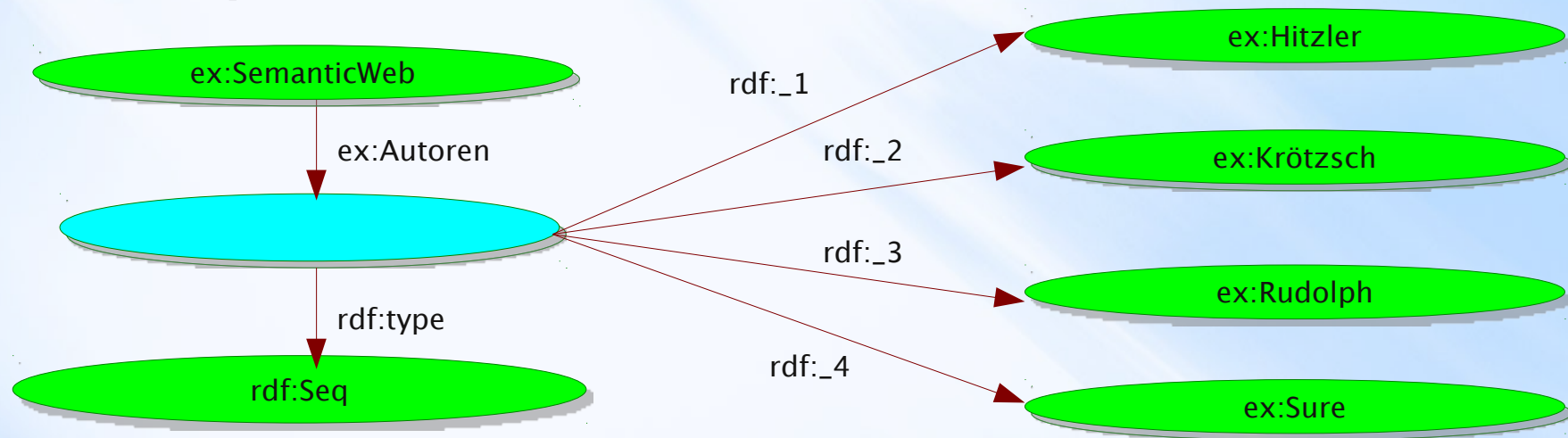
```
@prefix ex: <http://example.org> .  
ex:Chutney ex:hatZutat [ ex:Zutat      ex:grueneMango ;  
                        ex:Menge      "450g" ] .
```

- Listen in RDF

- Datenstrukturen zur Aufzählung von Ressourcen mit relevanter Reihenfolge (z.B. Autoren eines Buches)
- Unterscheidung zwischen
  - Offene Listen (Container)  
Können jederzeit erweitert werden
  - Geschlossene Listen (Collections)  
Neue Einträge können nicht hinzugefügt werden
- Besitzen keine zusätzliche Ausdrucksstärke, da auch mit bereits behandelten Mitteln modellierbar

# Sprachen des Semantic Web – RDF

- Graph



- In RDF/XML (verkürzt)

```
<rdf:Description rdf:about="&ex;SemanticWeb">
  <ex:Autoren>
    <rdf:Seq>
      <rdf:li rdf:resource="&ex;Hitzler" />
      <rdf:li rdf:resource="&ex;Kröttsch" />
      <rdf:li rdf:resource="&ex;Rudolph" />
      <rdf:li rdf:resource="&ex;Sure" />
    </rdf:Seq>
  </ex:Autoren>
</rdf:Description>
```

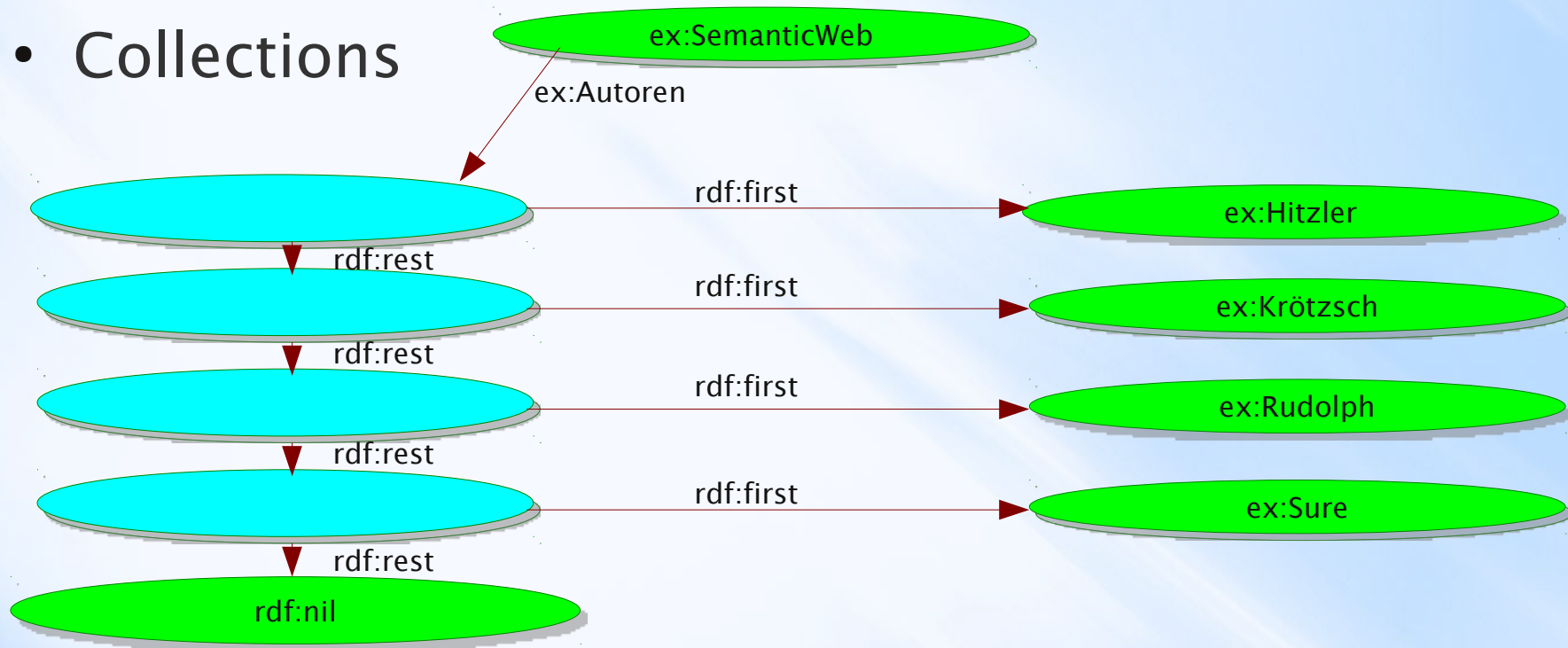


# Sprachen des Semantic Web – RDF

- Der Listentyp wird dem Listenknoten mit dem Prädikat **rdf:type** zugewiesen
- Folgende Typen sind im RDF definiert:
  - rdf:Seq  
geordnete Liste (Sequenz)
  - rdf:Bag  
ungeordnete Menge, reihenfolge nicht von Belang
  - rdf:Alt  
Menge alternativer Möglichkeiten aus denen immer nur eine Relevanz besitzt

# Sprachen des Semantic Web – RDF

- Collections



- Zerlegung der Liste in Kopfelemente und Restliste
- Wenn Restliste = `rdf:nil`, ist das Ende erreicht

# Sprachen des Semantic Web – RDF

- Collections

- RDF/XML-Syntax

```
<rdf:Description rdf:about="&ex;SemanticWeb">  
  <ex:Autoren rdf:parseType="Collection" >  
    <rdf:Description rdf:about="&ex;Hitzler">  
    <rdf:Description rdf:about="&ex;Kröttsch">  
    <rdf:Description rdf:about="&ex;Rudolph">  
    <rdf:Description rdf:about="&ex;Sure">  
  </ex:Autoren>  
</rdf:Description>
```

- Turtle

```
@prefix ex: <http://example.org> .  
ex:SemanticWeb      ex:Autoren      ( ex:Hitzler      ex:Kröttsch  
                                   ex:Rudolph      ex:Sure ) .
```

- Verbreitung von RDF
  - Große Anzahl an Tools und Bibliotheken für RDF
    - Die meisten davon frei (Open Source) verfügbar
  - Es existieren diverse Systeme zum effizienten Umgang mit riesigen RDF-codierten-Datenmengen
    - Sogenannte Triple-Stores
  - Auch kommerzielle Anbieter wie Oracle implementieren zunehmend RDF-Unterstützung
  - Diverse Datenformate basieren auf RDF:
    - RSS 1.0
    - XMP (Adobe)
    - SVG

- **Bewertung**

- Etablierter und gut unterstützter Standard zur Speicherung und zum Austausch von Daten
- Weitgehend syntaxunabhängige Darstellung durch abstraktes, graphenbasiertes Datenmodell
- Gut geeignet zur Vereinigung von Daten aus verschiedenen Quellen

- **Kritik**

- Reines RDF codiert eigentlich nur Informationen über Individuen / Ressourcen
- Es besteht praktisch keine Möglichkeit zur Codierung von Schemawissen → Siehe RDF Schema



Noch Fragen ?

# Sprachen des Semantic Web – XML

- Literatur:

- Buch “Semantic Web Grundlagen”, Springer Verlag 2008  
Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph, York Sure  
ISBN: 978-3-540-33993-9
- RDF Webseite  
<http://www.w3.org/RDF/>
- Resource Description Framework Primer  
F. Manola, E. Miller  
W3C Recommendation, 10<sup>th</sup> February 2004  
<http://www.w3.org/TR/rdf-primer>
- RDF Planet – Blogs zum Thema RDF / Semantic Web  
<http://www.planetrdf.com/>