# RDF

## (Hasta RDFs)

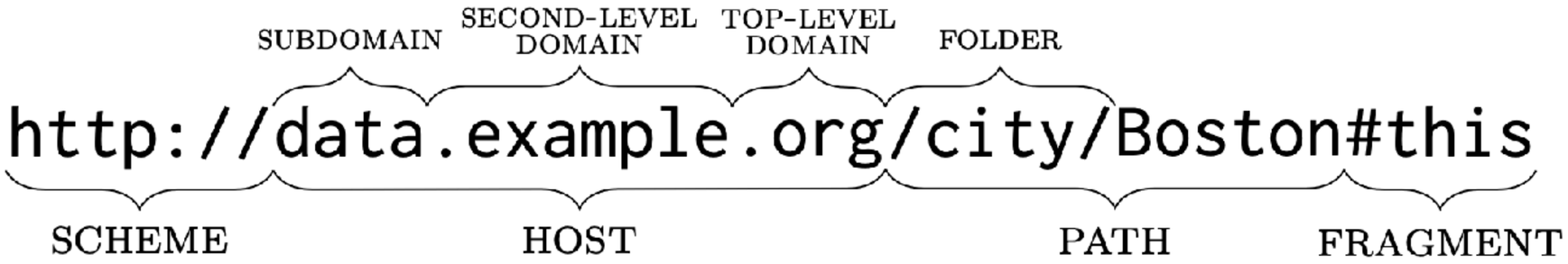
# RDF

# Términos

- En RDF se utilizan diferentes términos referencias a los recursos que se intentan describir.  Hay tres tipos de términos
  - IRIS
  - Literales
  - Blank Nodes

# Internationalized Resource Identifiers (IRIs)

- Identificadores globales en la Web
- URL, URI, e IRIs
- En RDF las IRIs serán los identificadores de los recursos.




**¿Qué sucede si diferentes silos usan diferentes IRIs para identificar el mismo recurso?**

# Literales

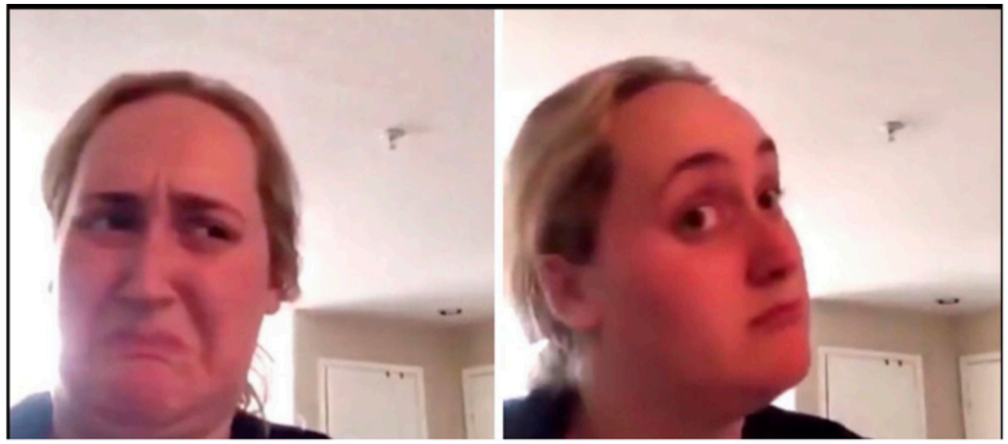
- Agregan información más “comprensible” por personas como textos, números, fechas, etc.
- Formalmente es texto que representa valores numéricos, booleanos, de text, fechas, etc.
- Poseen por lo general dos de las siguientes partes: la forma lexica (Unicode), la IRI del datatype y una etiqueta del lenguaje.
  - “2”^^xsd:decimal
  - “El nombre de la rosa”@es

# Literales

- Boolean
  - xsd:boolean “true”
- Numericos
  - Xsd:integer “-1”
- Temporales
  - Xsd:date "2012-02-29", "2012-12-31+04:00"
- Texto
  - Xsd:string “El lujo es vulgaridad”@es

# Blank Nodes

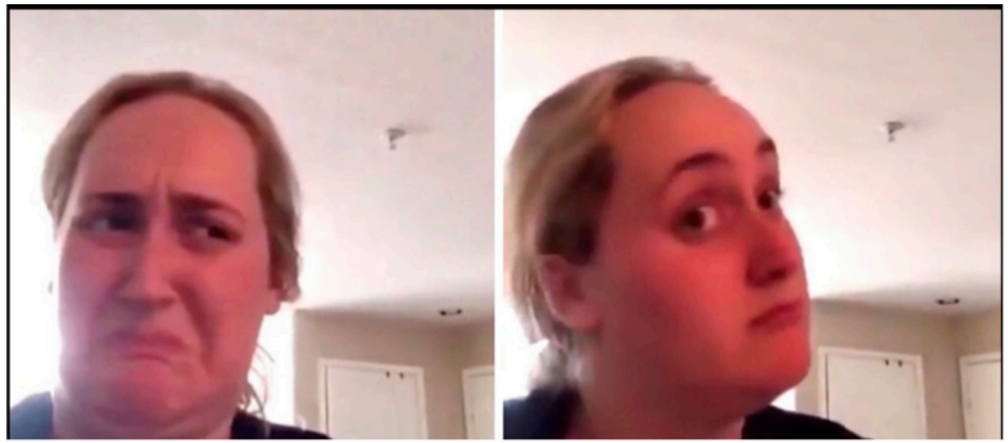
- RDF incluye los *blank nodes* como la forma de representar la existencia de algún recurso en lugar de identificarlo específicamente.



**¿En qué contexto querrían hablar de algo pero sin querer identificarlo?**

# Blank Nodes

- Un libro tiene un autor anónimo.
- El elemento es tan recurrente o pequeño que no tiene sentido identificarlo
  - Un punto en un partido de tenis.



# TRIPLETAS RDF

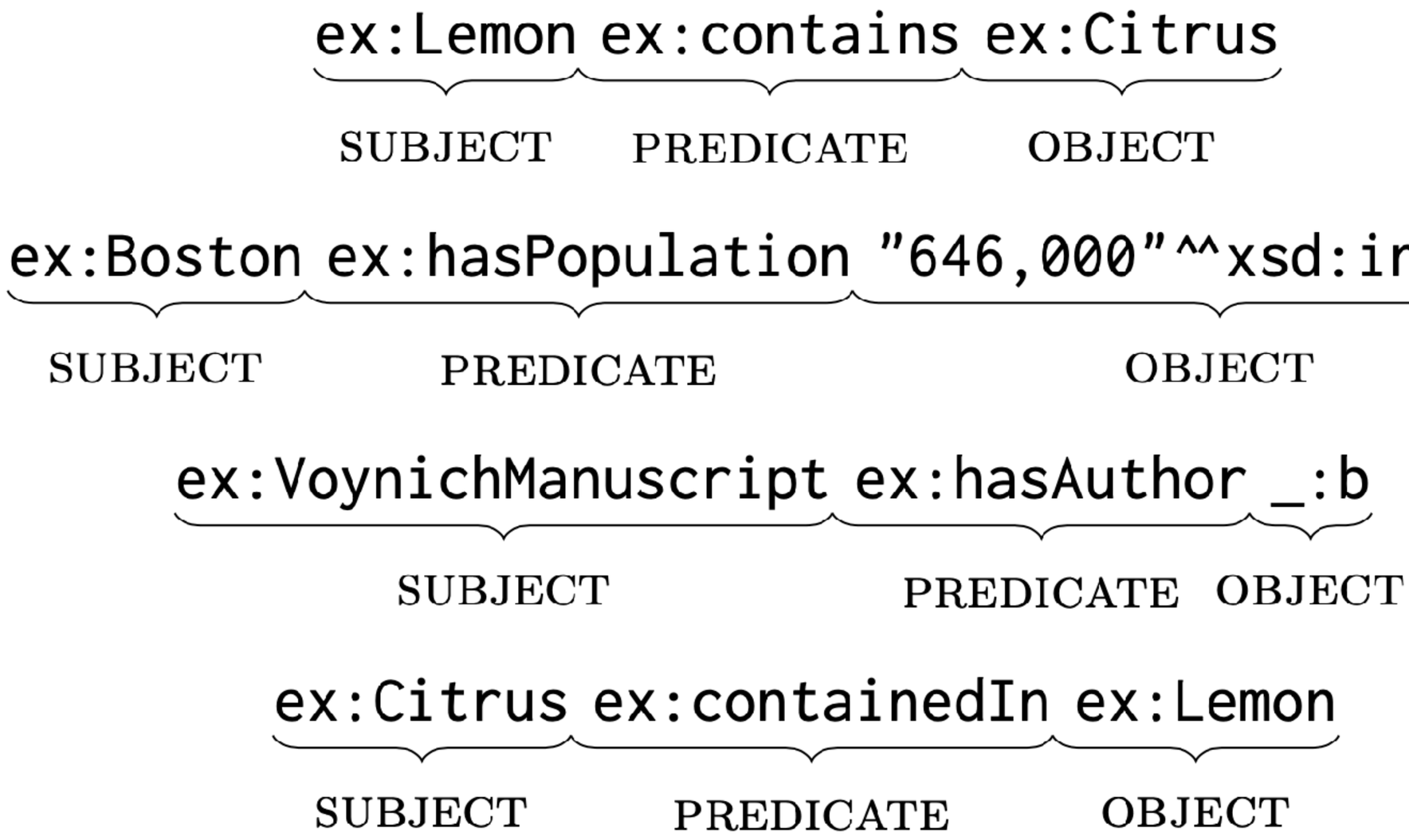
- Tripletas (en inglés triplets)… ternas? tresillo? :P
- Formalizemos lo que hicimos
  - Nosotros “conectamos” los datos …
  - pero una simple conexión no alcanza, los datos deben poseer nombres
  - aquí es donde aparecen las tripletas RDF: una conexión nombrada (etiquetada) entre dos recursos

# TRIPLETAS RDF

- Una tripleta RDF (s,p,o) tiene la siguiente forma:
  - “s” y “p” son URIs, esto es recursos en la Web.
  - “o” es una URI o un literal
    - “s”, “p”, y “o” viene de “sujeto” (*subject*), “propiedad” (*property*), y “objeto” (*object*)

```
(<http://.../Gabriel_García_Márquez>, <rdf:type>, <http://.../ColombianShortStoryWriters>)
```

- *RDF* es un modelo general para estas tripletas en formatos que puedan ser comprensibles por computadoras (RDF/XML, Turtle, N3, RDFa, Json,etc)


ex:Lemon ex:contains ex:Citrus
SUBJECT PREDICATE OBJECT
ex:Boston ex:hasPopulation "646,000"^^xsd:integer
SUBJECT PREDICATE OBJECT
ex:VoynichManuscript ex:hasAuthor _:b
SUBJECT PREDICATE OBJECT
ex:Citrus ex:containedIn ex:Lemon
SUBJECT PREDICATE OBJECT

# TRIPLETAS RDF

- Los recursos pueden utilizar cualquier URI
  - **`http://www.example.org/file.html#home`**
  - **`http://www.example.org/f.xml#xpath(//q[@a=b])`**
  - **`http://www.example.org/form?a=b&c=d`**
- Las sentencias RDF forman un grafo dirigido con etiquetas.
  - esta es la mejor forma de pensar en la organización de estas sentencias

# OTRO EJEMPLO MAS DIVERTIDO EN TURTLE

```
@base <http://example.org/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rel: <http://www.perceive.net/schemas/relationship/> .

<#green-goblin>
    rel:enemyOf <#spiderman> ;
    a foaf:Person ;    # in the context of the Marvel universe
    foaf:name "Green Goblin" .

<#spiderman>
    rel:enemyOf <#green-goblin> ;
    a foaf:Person ;
    foaf:name "Spiderman", "Человек-паук"@ru .
```

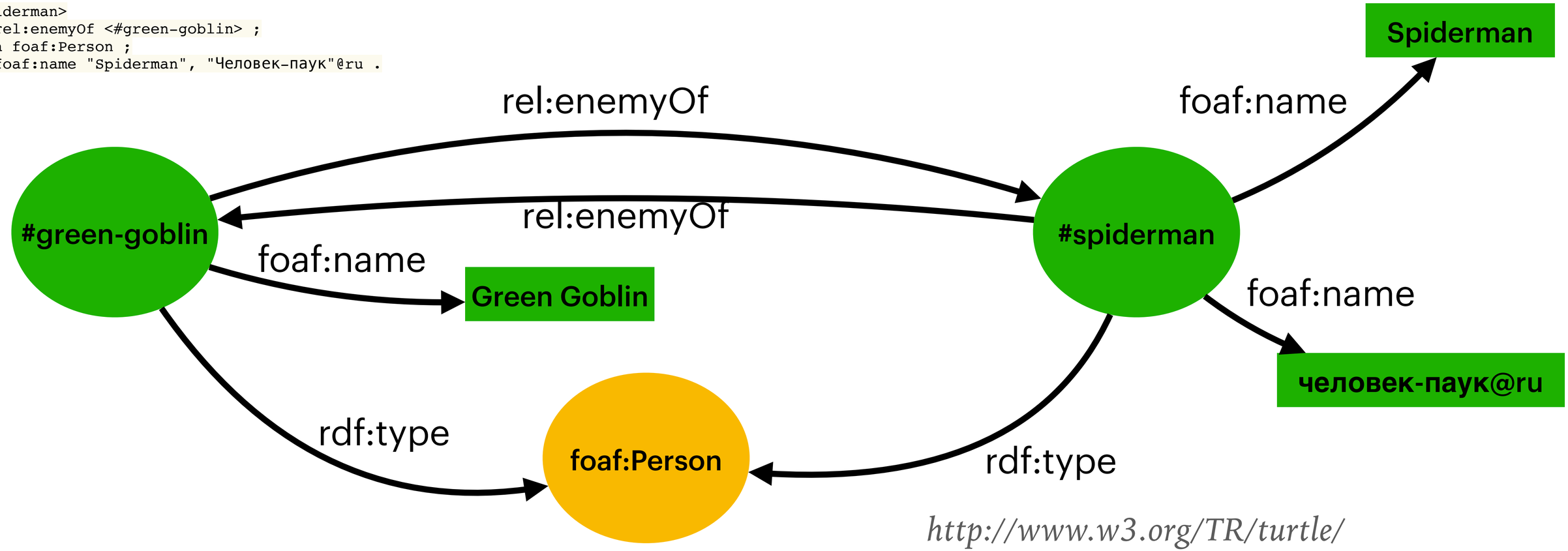
*http://www.w3.org/TR/turtle/*

# OTRO EJEMPLO MAS DIVERTIDO EN TURTLE

```
@base <http://example.org/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rel: <http://www.perceive.net/schemas/relationship/> .

<#green-goblin>
    rel:enemyOf <#spiderman> ;
    a foaf:Person ;    # in the context of the Marvel universe
    foaf:name "Green Goblin" .

<#spiderman>
    rel:enemyOf <#green-goblin> ;
    a foaf:Person ;
    foaf:name "Spiderman", "Человек-паук"@ru .
```




*http://www.w3.org/TR/turtle/*

# RDF/XML ejemplo

```
<?xml version="1.0"?>
<!DOCTYPE img [
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
]>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:ex1="http://ex1.org/#">
  <rdf:Description
        xml:lang="en"
        rdf:about="http://ex1.org/#Jen"
        rdfs:label="Jen">
    <rdf:type rdf:resource="http://ex1.org/#Person" />
    <rdf:type>
      <rdf:Description rdf:about="http://ex1.org/#Female" />
    </rdf:type>
    <ex1:allergy>
      <rdf:Description rdf:about="http://ex1.org/#Citrus" />
    </ex1:allergy>
    <ex1:location rdf:nodeID="loc" />
  </rdf:Description>
  <rdf:Description rdf:nodeID="loc">
    <ex1:long rdf:datatype="&xsd;decimal">-9.0</ex1:long>
    <ex1:lat rdf:datatype="&xsd;decimal">53.3</ex1:lat>
  </rdf:Description>
</rdf:RDF>
```

# RDF N3 ejemplo

```
<http://ex1.org/#Jen> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://ex1.org/#Person> .
<http://ex1.org/#Jen> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://ex1.org/#Female> .
<http://ex1.org/#Jen> <http://www.w3.org/2000/01/rdf-schema#label> "Jen"@en .
<http://ex1.org/#Jen> <http://ex1.org/#allergy> <http://ex1.org/#Citrus> .
<http://ex1.org/#Jen> <http://ex1.org/#location> _:loc .
_:loc <http://ex1.org/#lat> "53.3"^^<http://www.w3.org/2001/XMLSchema#decimal> .
_:loc <http://ex1.org/#long> "-9.0"^^<http://www.w3.org/2001/XMLSchema#decimal> .
```

# RDFa ejemplo

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Recipe for Coffee Parfait</title>
  <base href="http://ex.org/" />
</head>
<body vocab="http://ex.org/#" lang="en"
  prefix="rdfs: http://www.w3.org/2000/01/rdf-schema#">
  <div typeof="Recipe" resource="#CoffeeParfait">
    <h1 property="rdfs:label">Coffee Parfait</h1>

    <p>Time: <span property="minutes" datatype="xsd:integer"
                   content="25">25 mins</span></p>

    <h2>Ingredients:</h2>
    <ul rel="ingredient">
      <li about="#Yolk" property="rdfs:label">Egg Yolk</li>
      <li about="#Sugar" property="rdfs:label">Sugar</li>
      <li about="#Cream" property="rdfs:label">Cream</li>
      <li about="#Coffee" property="rdfs:label">Coffee</li>
    </ul>
  </div>
</body>
</html>
```

# RDF Turtle ejemplo

```
@prefix rdfa: <http://www.w3.org/ns/rdfa#> .
@prefix ex: <http://ex.org/#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://ex.org/> rdfa:usesVocabulary ex: .

ex:CoffeeParfait a ex:Recipe ;
    rdfs:label "Coffee Parfait"@en ; ex:minutes 25 ;
    ex:ingredient ex:Coffee , ex:Cream , ex:Yolk , ex:Sugar .

ex:Coffee rdfs:label "Coffee"@en .
ex:Cream rdfs:label "Cream"@en .
ex:Yolk rdfs:label "Egg Yolk"@en .
ex:Sugar rdfs:label "Sugar"@en .
```

# RDF JSON-LD ejemplo

```
{
  "@context": {
    "xsd": "http://www.w3.org/2001/XMLSchema#",
    "@base": "http://example.com/",
    "@vocab": "http://example.com/#",
    "label": "http://www.w3.org/2000/01/rdf-schema#label",
    "minutes": {
      "@id": "minutes",
      "@type": "xsd:integer"
    },
    "@language": "en"
  },
  "@id": "#CoffeeParfait",
  "@type": "Recipe",
  "label": "Coffee Parfait",
  "minutes": "25",
  "ingredient": [
    { "@id": "#Yolk", "label": "Egg Yolk"},
    { "@id": "#Sugar", "label": "Sugar"},
    { "@id": "#Cream", "label": "Cream"},
    { "@id": "#Coffee", "label": "Coffee"}
  ]
}
```

# Comparación

RDF/XML [41]: an XML-based syntax; the first standard RDF syntax.

- *Pros*: XML-compatible; widely supported.
- *Cons*: unintuitive; diverse XML can represent the same RDF data.

N-Triples [40]: simple line-based syntax for RDF.

- *Pros*: simple to serialise, parse, and process.
- *Cons*: verbose; full IRIs are less human-friendly.

Turtle [39]: human-friendly syntax for RDF.

- *Pros*: most human-friendly to read and write; concise.
- *Cons*: more complex to parse and process.

RDFa [185]: syntax embeddable in HTML webpages.

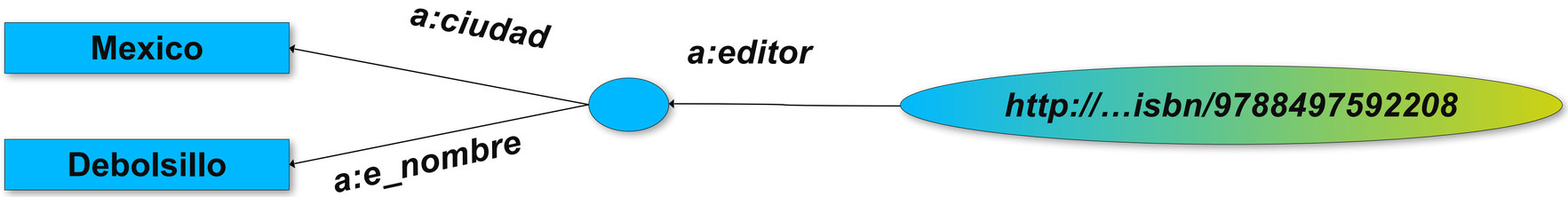
- *Pros*: allows for managing one webpage for humans and machines.
- *Cons*: requires extraction from a webpage; can be unintuitive.

JSON-LD [366]: a JSON-based RDF syntax.

- *Pros*: JSON-compatible; widely used; intuitive; supports datasets.
- *Cons*: incompatibilities with RDF (resolved in JSON-LD 1.1 [229]).

# NODOS INTERNOS (BLANK NODES)

- Veamos la siguiente sentencia:
  - “el editor es una “cosa” (algo) (*thing*) que posee un nombre y una dirección/localidad.
- Hasta ahora los nodos estaban identificados por una URI. Sin embargo….
- ¿Cuál es la URI para una cosa ?



# EXISTEN DIFERENTES SOPORTES EN LENGUAJES DE PROGRAMACIÓN – JENA

- RDF y otros elementos de Web Semántica en Java
- API para manipular un grafo RDF
- Statement representa una tripleta.
  - getSubject() retorna un Resource
  - getObject() retorna un RDFNode
  - getPredicate() retorna una Property
- Permite el manejo de espacio de nombres
- Reglas de inferencia
- Motor de consultas SPARQL (lo veremos más adelante)

# EXISTEN DIFERENTES SOPORTES EN LENGUAJES DE PROGRAMACIÓN – PYTON

- Python y RDFLib
  - existe el objeto “Graph”.
  - Un archivo RDF se parsea en un objeto Graph
  - el Graph ofrece métodos para obtener o agregar:
    - tripletas
    - pares (property,object) para un sujeto particular
    - (subject,property) para un objeto particular
    - etc.