# SEO

## Search Engine Optimization Using Semantic Data (RDFa)


## The <meta> tag

### Definition and Usage

Metadata is data (information) about data.

The <meta> tag provides metadata about the HTML document. Metadata will not be displayed on the page, but will be machine parsable.

Meta elements are typically used to specify page description, keywords, author of the document, last modified, and other metadata.

The metadata can be used by browsers (how to display content or reload page), search engines (keywords), or other web services.

### Example

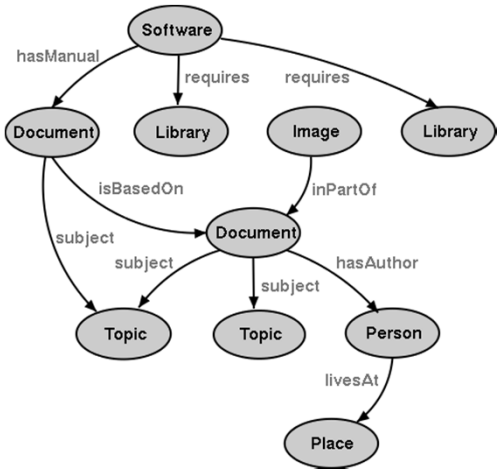
Describe metadata within an HTML document:

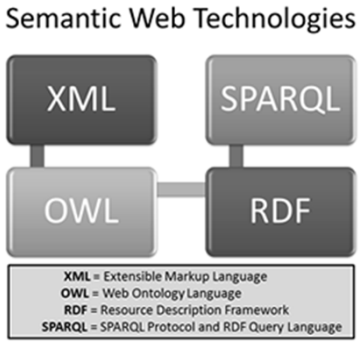
```
<head>
  <meta charset="UTF-8">
  <meta name="description" content="Free Web tutorials">
  <meta name="keywords" content="HTML,CSS,XML,JavaScript">
  <meta name="author" content="John Doe">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
```

BUT, more detailed meta information is needed to more accurately categorize the data on web pages.

## It's all about...


## The Semantic Web




Where URLs/URIs Identify Things

# BUT, what if there is no URL/URI for something?

# Options when NO URL? NO URI?

- Create your own resource reference in your own namespace
  - @prefix : <http://s2.smu.edu/7320/#>.
  - :someguy foaf:name “Rollo” .

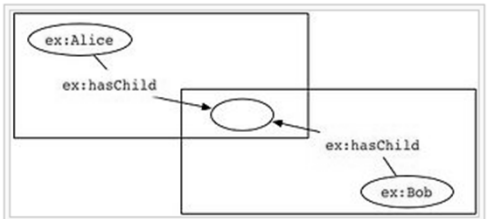
OR

- Define a blank node (some thing that ...)
  - _:someguy foaf:name “Rollo” .

does not require adding your own personal namespace – which others probably will not know about

# Anonymous or Blank Nodes

- In RDF, a **blank node** is a node in an RDF graph representing a resource for which a URI or literal is not given.
- The resource represented by a blank node is also called an **anonymous resource**.
- According to the RDF standard a blank node can only be used as subject or object of an RDF triple.



Example of a blank node in a RDF graph

# blank nodes

A fresh RDF blank node is allocated for each unique blank node label in a document. Repeated use of the same blank node label identifies the same RDF blank node.

EXAMPLE 14

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

_:alice foaf:knows _:bob .
_:bob foaf:knows _:alice .
```

Blank nodes can also be written using [..]

EXAMPLE 15

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

# Someone knows someone else, who has the name "Bob".
[] foaf:knows [ foaf:name "Bob" ] .
```

# more blank nodes...

EXAMPLE 16

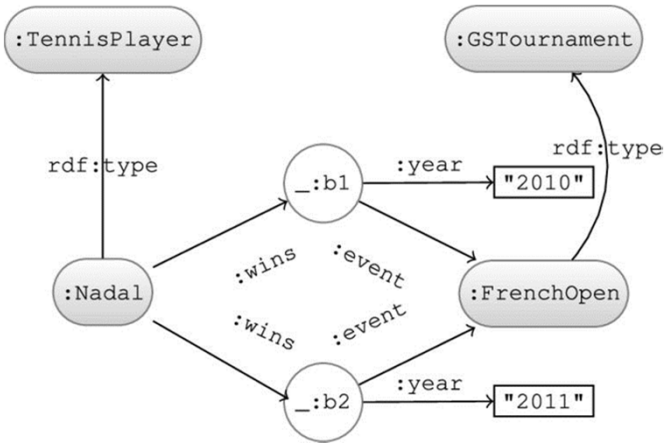
```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

[ foaf:name "Alice" ] foaf:knows [
    foaf:name "Bob" ;
    foaf:knows [
        foaf:name "Eve" ] ;
    foaf:mbox <bob@example.com> ] .
```

Someone named Alice knows someone named Bob who knows someone named Eve.

Corresponding simple triples:

EXAMPLE 17

```
_:a <http://xmlns.com/foaf/0.1/name> "Alice" .
_:a <http://xmlns.com/foaf/0.1/knows> _:b .
_:b <http://xmlns.com/foaf/0.1/name> "Bob" .
_:b <http://xmlns.com/foaf/0.1/knows> _:c .
_:c <http://xmlns.com/foaf/0.1/name> "Eve" .
_:b <http://xmlns.com/foaf/0.1/mbox> <bob@example.com> .
```

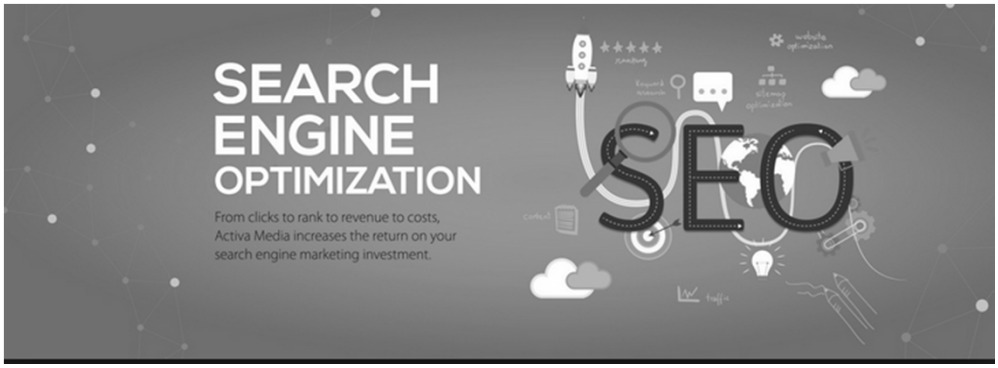


# Turtle

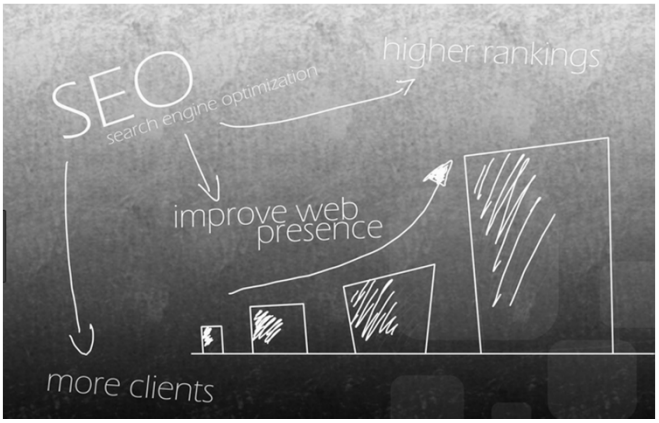
Анонимный узел можно записать также в [...].
Субъект в [...] всегда опущен.

```
@prefix : <http://example.com/>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix schema: <http://schema.org/>.


:vasya rdf:type schema:Person;
       schema:name "Вася Пупкин"@ru,
                   "Vasya Pupkin"@en;
       schema:birthDate "1990-04-30"^^xsd:date;
       schema:email <mailto:vasya.pupkin@gmail.com>,
                    <mailto:vasily@example.com>;
       schema:knows :petya,
                    :masha,
                    [
                    rdf:type schema:Person;
                    schema:name "Марья Ивановна"@ru,
                                "Maria Ivanovna"@en
                    ].
```

What does this mean?



SEO – the practice of optimizing a website to achieve higher rankings on serach engine pages (SERPs) – it means $$$$$$$$


Three (competing) technologies for Adding Semantic Markup to web pages

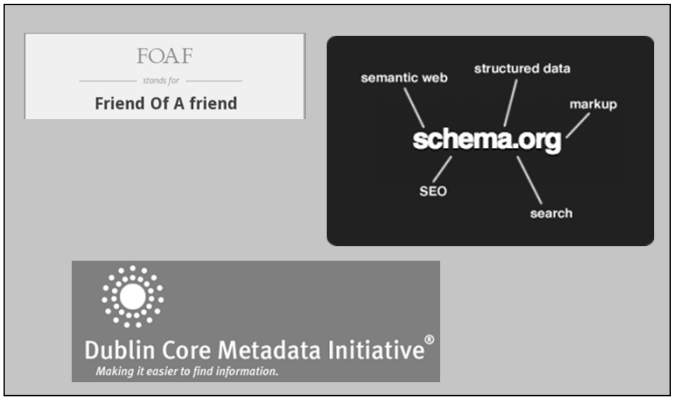
W3C RDFa

SEO MICRODATA

Semantic Web

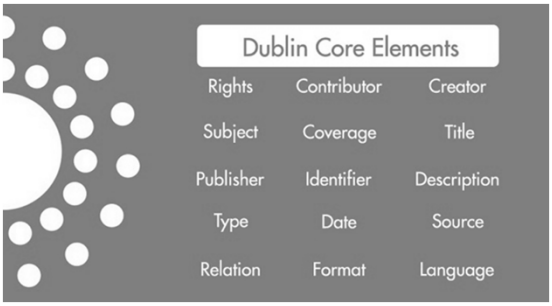
JSON-LD

# Common Ontologies




Here is where you find Classes and Properties of Things


Use dbpedia for URI/URLs of well-known entities: Cities, countries, famous people, events, ...




Dublin Core was one of the first ontologies created for the Semantic Web – used to describe documents. The spec was created in Dublin, Ohio.

```
@prefix :     <http://www.codesupreme.com/#>.
@prefix dc:  <http://purl.org/dc/elements/1.1/>.
@prefix xsd:  <http://www.w3.org/XMLSchema#> .

:book1 dc:title "Homage to Catalonia" .
:book1 dc:creator "George Orwell" .
:book1 dc:type  "text" .
:book1 dc:publisher "Secker and Warburg" .
:book1 dc:date "1938"^^xsd:date .
:book1 dc:language "english" .
```


## Thing

This is the top level term in the schema.org hierarchy: a collection of types (or "classes"), each of which has one or more parent types.

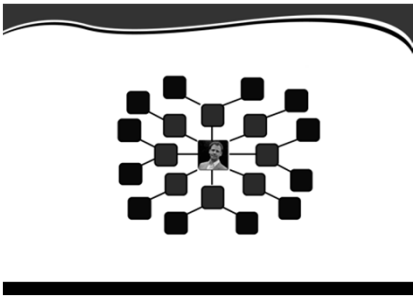
A type may have more than one super-type.

**Subclasses of Thing**

Action
CreativeWork
Event
Intangible
Organization
Person
Place
Product

```
schema:Event  rdfs:subClassOf  schema:Thing .
```

# foaf



Classes: | Agent | Document | Group | Image | LabelProperty | OnlineAccount | OnlineChatAccount | OnlineEcommerceAccount | OnlineGamingAccount | Organization | Person | PersonalProfileDocument | Project |

Properties: | account | accountName | accountServiceHomepage | age | aimChatID | based_near | birthday | currentProject | depiction | depicts | dnaChecksum | familyName | family_name | firstName | focus | fundedBy | geekcode | gender | givenName | givenname | holdsAccount | homepage | icqChatID | img | interest | isPrimaryTopicOf | jabberID | knows | lastName | logo | made | maker | mbox | mbox_sha1sum | member | membershipClass | msnChatID | myersBriggs | name | nick | openid | page | pastProject | phone | plan | primaryTopic | publications | schoolHomepage | sha1 | skypeID | status | surname | theme | thumbnail | tipjar | title | topic | topic_interest | weblog | workInfoHomepage | workplaceHomepage | yahooChatID |

## Semantic Web for SEO

Schema.org and Dublin Core are recognized by Bing, Google and Yahoo. Ontology terms are used to help categorize web pages and improve the accuracy of search engines.

Add semantic markup to your web pages to make them understandable to search engines.

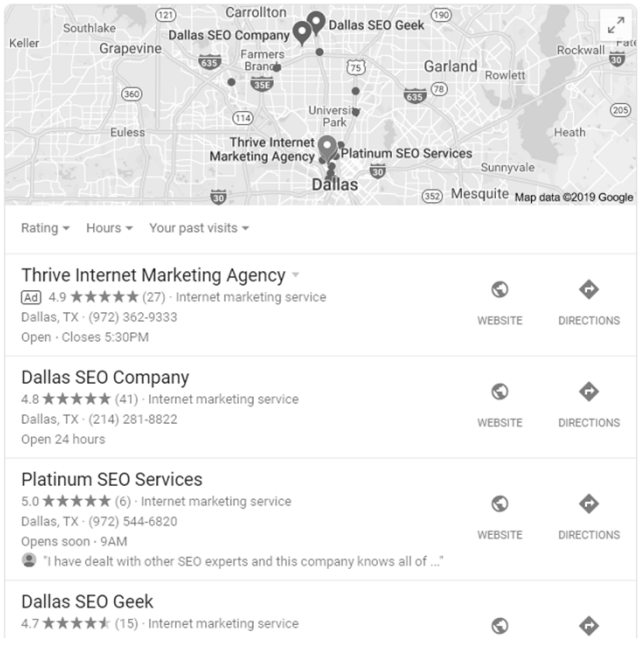
## Three Markup Technologies

W3C RDFa

SEO MICRODATA

Semantic Web

JSON-LD




RDFa is a technology to add semantic meaning to web pages and optimize search engine discovery of what the page is about.

Goal: Increase page hits ... increase $$$$$$

# rdfa-lite examples
# how to markup a web page

https://www.w3.org/TR/rdfa-lite/

# Manu Sporny's Blog Webpage


- http://manu.sporny.org

Manu is a web developer who wants to get the word out that his blog/webpage is about Manu Sporny.

He wants search engines to be able to retrieve information about him

# The  Tag

## Definition and Usage

The  tag defines a division or a section in an HTML document.

The  element is often used as a container for other HTML elements to style them with CSS or to perform certain tasks with JavaScript.

### Example

A section in a document that will have a light blue background color:

```
<div style="background-color:lightblue">
  <h3>This is a heading</h3>
  <p>This is a paragraph.</p>
</div>
```

# The  Tag

## Definition and Usage

The  tag is used to group inline-elements in a document.

The  tag provides no visual change by itself.

The  tag provides a way to add a hook to a part of a text or a part of a document.

### Example

A  element used to color a part of a text:

```
<p>My mother has <span style="color:blue">blue</span> eyes.</p>
```

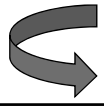
## 2.1 vocab, typeof, and property

RDFa, like Microformats [microformats] and Microdata [microdata], enables us to talk about *things* on the Web such that a machine can understand what we are saying. Typically when we talk about a thing, we use a particular **vocabulary** to talk about it. So, if you wanted to talk about People, the vocabulary that you would use would specify terms like *name* and *telephone number*. When we want to mark up things on the Web, we need to do something very similar, which is specify which vocabulary that we are going to be using. Here is a simple example that specifies a vocabulary that we intend to use to markup things in the paragraph:

EXAMPLE 1 http://manu.sporny.org

```
<p vocab="http://schema.org/">
   My name is Manu Sporny and you can give me a ring via 1-800-555-0199.
</p>
```

In this example we have specified that we are going to be using the **vocabulary** that can be found at `http://schema.org/`. This is a vocabulary that has been released by major search engine companies to talk about common things on the Web that Search Engines care about – things like People, Places, Reviews, Recipes, and Events. Once we have specified the vocabulary, we need to specify the **type of** the thing that we're talking about. In this particular case we are talking about a Person, which can be marked up like so:



http://manu.sporny.org

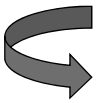
The web page (resource) http://manu.sporny.org is a URL that represents a Person (defined in schema.org)

EXAMPLE 2

```
<p vocab="http://schema.org/" typeof="Person">
   My name is Manu Sporny and you can give me a ring via 1-800-555-0199.
</p>
```

Now all we need to do is specify which **properties** of that person we want to point out to the search engine. In the following example, we mark up the person's name, phone number and web page. Both text and URLs can be marked up with RDFa Lite. In the following example, pay particular attention to the types of data that are being pointed out to the search engine, which are highlighted in blue:

What N3 properties does this Person have ?



EXAMPLE 3

```
<p vocab="http://schema.org/" typeof="Person">
   My name is
   <span property="name">Manu Sporny</span>
   and you can give me a ring via
   <span property="telephone">1-800-555-0199</span>
   or visit
   <a property="url" href="http://manu.sporny.org/">my homepage</a>.
</p>
```

Now, when somebody types in "phone number for Manu Sporny" into a search engine, the search engine can more reliably answer the question directly, or point the person searching to a more relevant Web page.

Since the vocab is schema.org we don't need the prefix

## 2.2 resource

If you want Web authors to be able to talk *about* each thing on your page, you need to create an identifier for each of these things. Just like we create identifiers for parts of a page using the `id` attribute in HTML, you can create identifiers for things described on a page using the `resource` attribute:

EXAMPLE 4

Now Manu's URI is : http://manu.sporny.org#manu

```
<p vocab="http://schema.org/" resource="#manu" typeof="Person">
   My name is
   <span property="name">Manu Sporny</span>
   and you can give me a ring via
   <span property="telephone">1-800-555-0199</span>.
   <img property="image" src="http://manu.sporny.org/images/manu.png" />
</p>
```

Manu's Cat "Fluffy"

we could talk about his cat 'fluffy' = add:

```
<div resource="#fluffy" typeof="Cat">
  My cat is <span property="name">Fluffy</span>
</div>
```

Now Manu's cat's URI is : http://manu.sporny.org#fluffy

## 2.3 prefix

In some cases, a vocabulary may not have all of the terms an author needs when describing their *thing*. The last feature in RDFa 1.1 Lite that some authors might need is the ability to specify more than one vocabulary. For example, if we are describing a Person and we need to specify that they have a favorite animal, we could do something like the following:

EXAMPLE 5

```
<p vocab="http://schema.org/" prefix="ov: http://open.vocab.org/terms/" resource="#manu" typeof="Person">
   My name is
   <span property="name">Manu Sporny</span>
   and you can give me a ring via
   <span property="telephone">1-800-555-0199</span>.
   <img property="image" src="http://manu.sporny.org/images/manu.png" />
   My favorite animal is the <span property="ov:preferredAnimal">Liger</span>.
</p>
```

The example assigns a short-hand prefix to the Open Vocabulary (`ov`) and uses that prefix to specify the `preferredAnimal` vocabulary term. Since schema.org doesn't have a clear way of expressing a favorite animal, the author instead depends on this alternate vocabulary to get the job done.

RDFa Lite also pre-defines a number of useful and popular prefixes,

such as dc, foaf, and schema.

This ensures that even if authors forget to declare the popular prefixes, that their structured data will continue to work.

```
<div>
 <p>Rollo</p>
<div>
```

```
<div>
 <p><span foaf:property="name">Rollo</span></p>
<div>
```

RDFa knows the prefixes:
schema, foaf and dc
No need to define. But you must then use the prefixes.

# RDFa Tutorial examples

EXAMPLE 1

```
<html>
<head>
  ...
</head>
<body>
  ...
  <h2>The Trouble with Bob</h2>
  <p>Date: 2011-09-10</p>
  ...
</body>
```

We have a web page and want to describe what the text in the page refers to.

Marla likes the Dublin Core terms **title** and **created** (for date created)

EXAMPLE 2

```
<html>
<head>
  ...
</head>
<body>
  ...
  <h2 property="http://purl.org/dc/terms/title">The Trouble with Bob</h2>
  <p>Date: <span property="http://purl.org/dc/terms/created">2011-09-10</span></p>
  ...
</body>
```

What is the subject?
With no additional information, the assumption is “This Webpage”

EXAMPLE 2

```
<html>
<head>
  ...
</head>
<body>
  ...
  <h2 property="http://purl.org/dc/terms/title">The Trouble with Bob</h2>
  <p>Date: <span property="http://purl.org/dc/terms/created">2011-09-10</span></p>
  ...
</body>
```

But, annoying to write long URLs for properties

```
<html>
<head>
  ...
</head>
<body vocab="http://purl.org/dc/terms/">
  ...
  <h2 property="title">The Trouble with Bob</h2>
  <p>Date: <span property="created">2011-09-10</span></p>
  ...
</body>
```

Add vocab attribute in higher level tag

Here, we specifically describe the SUBJECT using resource=
In previous examples, the URL of the web page was the SUBJECT

EXAMPLE 11

```
<div resource="/alice/posts/trouble_with_bob">
    <h2 property="title">The trouble with Bob</h2>
    ...
    The trouble with Bob is that he takes much better photos than I do:
    ...
    <div resource="http://example.com/bob/photos/sunset.jpg">
      <img src="http://example.com/bob/photos/sunset.jpg" />
      <span property="title">Beautiful Sunset</span>
      by <span property="creator">Bob</span>.
    </div>
 </div>
```

Assumes there is a vocab element higher up in the document that defines the ontology (e.g. Dublin Core)

EXAMPLE 15

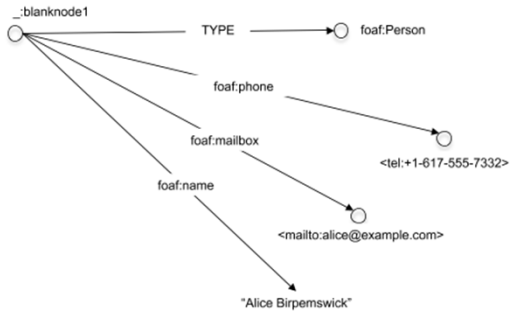
```
<div vocab="http://xmlns.com/foaf/0.1/" typeof="Person"><p>
  <p>
    <span property="name">Alice Birpemswick</span>,
    Email: <a property="mbox" href="mailto:alice@example.com">alice@example.com</a>,
    Phone: <a property="phone" href="tel:+1-617-555-7332">+1 617.555.7332</a>
  </p>
</div>
```

What is the SUBJECT? what Person?

It's a blank node.

Unknown but we know some properties.

Note how Alice did not specify a `resource` like she did when adding blog entry metadata. But, if she is not declaring what she is talking about, how does the RDFa Processor know what she's identifying? In RDFa, in the absence of a `resource` attribute, the `typeof` attribute on the enclosing `div` implicitly sets the subject of the properties marked up within that `div`. That is, the name, email address, and phone number are associated with a new node of type `Person`. This node has no URL to identify it, so it is called a *blank node* as shown on the figure:




NOTE: when the **property** attribute is with an **href** attribute, then the value of the property IS the value of the href.

*2.1.2.2 Describing Social Networks*

Alice continues to mark up her page by adding information about her friends, including at least their names and homepages. She starts with plain HTML:

EXAMPLE 16

```
<div>
   <ul>
      <li>
        <a href="http://example.com/bob/">Bob</a>
      </li>
      <li>
        <a href="http://example.com/eve/">Eve</a>
      </li>
      <li>
        <a href="http://example.com/manu/">Manu</a>
      </li>
   </ul>
</div>
```

How can we say that Bob, Eve and Manu are People (e.g. foaf Person)?

```
<div vocab="http://xmlns.com/foaf/0.1/">
   <ul>
      <li typeof="Person">
        <a href="http://example.com/bob/">Bob</a>
      </li>
      <li typeof="Person">
        <a href="http://example.com/eve/">Eve</a>
      </li>
      <li typeof="Person">
        <a href="http://example.com/manu/">Manu</a>
      </li>
   </ul>
</div>
```

EXAMPLE 16

```
<div>
   <ul>
      <li>
        <a href="http://example.com/bob/">Bob</a>
      </li>
      <li>
        <a href="http://example.com/eve/">Eve</a>
      </li>
      <li>
        <a href="http://example.com/manu/">Manu</a>
      </li>
   </ul>
</div>
```

How can we say that the href links are their homepages?

EXAMPLE 18

```
<div vocab="http://xmlns.com/foaf/0.1/">
   <ul>
      <li typeof="Person">
        <a property="homepage" href="http://example.com/bob/">Bob</a>
      </li>
      <li typeof="Person">
        <a property="homepage" href="http://example.com/eve/">Eve</a>
      </li>
      <li typeof="Person">
        <a property="homepage" href="http://example.com/manu/">Manu</a>
      </li>
   </ul>
</div>
```

EXAMPLE 16

```
<div>
   <ul>
      <li>
        <a href="http://example.com/bob/">Bob</a>
      </li>
      <li>
        <a href="http://example.com/eve/">Eve</a>
      </li>
      <li>
        <a href="http://example.com/manu/">Manu</a>
      </li>
   </ul>
</div>
```

How can we say that the Bob and Eve and Manu are names?

Add a value

Alice would also like to improve the markup by expressing each person's name using RDFa, too. That can be done by adding a separate `span` element and the relevant `property`:

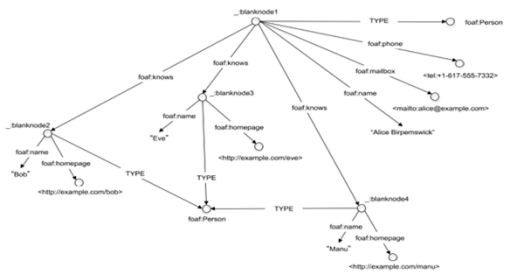
EXAMPLE 19

```
<div vocab="http://xmlns.com/foaf/0.1/">
  <ul>
      <li typeof="Person">
        <a property="homepage" href="http://example.com/bob/"><span property="name">Bob</span></a>
      </li>
      <li typeof="Person">
        <a property="homepage" href="http://example.com/eve/"><span property="name">Eve</span></a>
      </li>
      <li typeof="Person">
        <a property="homepage" href="http://example.com/manu/"><span property="name">Manu</span></a>
      </li>
   </ul>
</div>
```

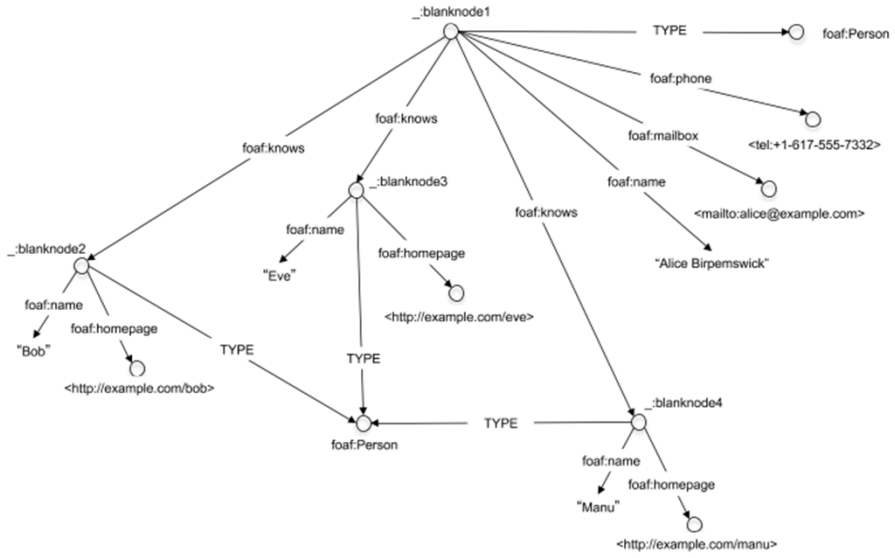
EXAMPLE 20

```
<div vocab="http://xmlns.com/foaf/0.1/" typeof="Person">
   <p>
    <span property="name">Alice Birpemswick</span>,
    Email: <a property="mbox" href="mailto:alice@example.com">alice@example.com</a>,
    Phone: <a property="phone" href="tel:+1-617-555-7332">+1 617.555.7332</a>
  </p>
   <ul>
      <li property="knows" typeof="Person">
        <a property="homepage" href="http://example.com/bob/"><span property="name">Bob</span></a>
      </li>
      <li property="knows" typeof="Person">
        <a property="homepage" href="http://example.com/eve/"><span property="name">Eve</span></a>
      </li>
      <li property="knows" typeof="Person">
        <a property="homepage" href="http://example.com/manu/"><span property="name">Manu</span></a>
      </li>
   </ul>
</div>
```

With this, Alice could describe here social network:



With this, Alice could describe here social network:




## What about working with multiple namespaces?

To alleviate this problem, RDFa offers the possibility of using *prefixed* terms: a special `prefix` attribute can assign prefixes to represent URLs and, using those prefixes, the vocabulary elements themselves can be abbreviated. The `prefix:reference` syntax is used: the URL associated with `prefix` is simply concatenated to `reference` to create a full URL. (Note that we have already used this convention to simplify our figures.) Here is how the HTML of the previous example looks like when prefixes are used:

EXAMPLE 28

```
<html>
 <head>
   ...
 </head>
 <body prefix="dc: http://purl.org/dc/terms/ schema: http://schema.org/">
   <div resource="/alice/posts/trouble_with_bob" typeof="schema:BlogPosting">
      <h2 property="dc:title">The trouble with Bob</h2>
      ...
      <h3 property="dc:creator" resource="#me">Alice</h3>
      <div property="schema:articleBody">
        <p>The trouble with Bob is that he takes much better photos than I do:</p>
      </div>
     ...
   </div>
 </body>
</html>
```

SEE: http://rdfa.info/play/
For interactive RDFa