

56.1	Un esempio prima di cominciare	597
56.2	Una base di dati fatta di «triple»	598
56.3	Vocabolari	599
56.4	Abbreviazioni	600
56.5	Nodi vuoti	601
56.6	Tipo di nodo o classe	602
56.7	RDFa	602
56.8	N-Triples	605
56.9	Turtle	606
56.10	Riferimenti	608

RDF sta per *Resource description framework* e definisce un «linguaggio» per rappresentare informazioni relative a risorse disponibili nel web, allo scopo di realizzare ciò che è noto come «web semantico». Questo lavoro ha origine nel 1999 e i documenti che da allora si sono succeduti utilizzano una terminologia che cambia nel tempo, mano a mano che i concetti astratti originali trovano attuazioni concrete; pertanto, piuttosto che soffermarsi sulla terminologia, è meglio cercare di capire il senso dei vari componenti.

Anche se in relazione a RDF si parla di «linguaggio», in pratica si tratta di un modo di schematizzare le informazioni; tali schemi vengono poi resi in forma testuale in modi diversi, a seconda della necessità o dell'opportunità.

56.1 Un esempio prima di cominciare

Prima è nato il «web», poi sono nati i motori di ricerca, poi si è posto il problema di dare maggiori informazioni ai motori di ricerca per migliorare la classificazione dei file accessibili attraverso la rete e del loro contenuto. A tale proposito, nelle intestazioni delle pagine HTML sono stati usati gli elementi **'META'**, per esempio così:

```
<head>
  <meta http-equiv="Content-Type"
        content="text/html; charset=UTF-8">
  <meta name="Author" content="Clara Drusilla">
  <meta name="Date" content="2012-12-31">
  <meta name="Description"
        content="Passatempo di una professoressa">
  <meta name="Keywords"
        content="hobby, tempo libero, mare, spiaggia, sole">
  <title>Il mio hobby preferito</title>
</head>
```

Attualmente è preferibile inserire del codice RDF (per cui si parla di HTML+RDF, o di RDFa), trasformando quanto già mostrato nell'esempio seguente:

```
<head>
  <meta http-equiv="Content-Type"
        property="http://purl.org/dc/terms/format"
        content="text/html; charset=UTF-8">
  <meta property="http://purl.org/dc/terms/title"
        content="Il mio hobby preferito">
  <meta property="http://purl.org/dc/terms/creator"
        content="clara.html">
  <meta property="http://purl.org/dc/terms/created"
        content="2012-12-31">
  <meta property="http://purl.org/dc/terms/description"
        content="Passatempo di una professoressa">
  <meta property="http://purl.org/dc/terms/subject"
        content="hobby, tempo libero, mare, spiaggia, sole">
  <title>Il mio hobby preferito</title>
</head>
```

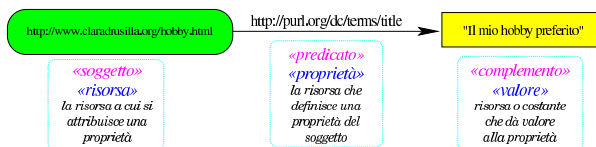
Come si vede, al posto degli attributi **'name'** ci sono attributi

'property', a cui si assegna un URI che definisce la proprietà dell'attributo 'content'. In pratica, il documento rappresentato dalla pagina HTML che contiene questa intestazione ha il titolo «Il mio hobby preferito», in quanto la proprietà che lo definisce è espressa dall'URI <http://purl.org/dc/terms/title>. Si osservi anche che in questo caso, al posto di indicare il nome dell'autrice (Clara Drusilla), si fa riferimento a un'altra pagina (*clara.html*) che presumibilmente contiene delle informazioni personali più articolate.

56.2 Una base di dati fatta di «triple»

« Nel modello delle basi di dati relazionali, i dati sono organizzati in relazioni che si suddividono in tuple; ovvero, si tratta di tabelle suddivise in righe. RDF permette di arrivare alla descrizione di una base di dati che ha la forma di un grafo orientato, attraverso delle **triple**, ovvero terne di valori. Per la precisione, RDF descrive solo delle triple, con cui si costruisce un grafo più o meno articolato.

Figura 56.3. Una tripla che individua il titolo della pagina <http://www.claradrusilla.org/hobby.html>.



I tre componenti della tripla hanno dei nomi convenzionali che però possono creare confusione, perché non esprimono bene il significato che hanno in pratica: **soggetto**, **predicato** e **oggetto**. In alternativa, come si vede nell'esempio della figura, possono essere sostituiti **risorsa**, **proprietà** e **valore**.

1. Il **soggetto**, ovvero il primo elemento di una tripla, può essere solo un URI (*uniform resource identifier*) e lo scopo della tripla è quello di conoscere il valore di una proprietà di questo soggetto. In casi eccezionali, come viene descritto più avanti nel capitolo, il soggetto può essere rappresentato da un «nodo vuoto», privo di URI.
2. Il **predicato**, ovvero il secondo elemento della tripla è un URI che definisce qualcosa che riguarda il soggetto e che ha un valore. Questo URI fa parte di solito di un vocabolario, ovvero una raccolta di proprietà che possa essere consultabile immediatamente. Va poi osservato anche che, nella realizzazione pratica di RDF, si vedono solo indirizzi che fanno capo al protocollo HTTP.
3. L'**oggetto**, ovvero il terzo elemento della tripla è il valore della proprietà. Questo valore può essere costituito da un indirizzo URI, oppure da un dato costante, come una stringa, una data o un numero.

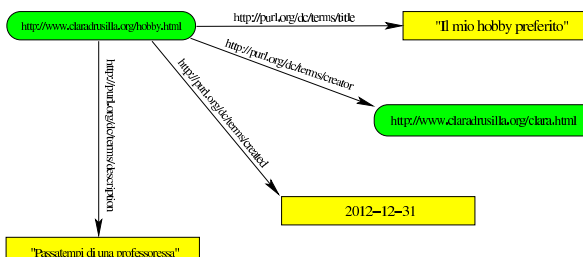
Va osservato che gli indirizzi URI, a cui ci si riferisce in questi casi, non sono necessariamente anche degli indirizzi URL. Pertanto, si tratta di indirizzi univoci, ma a questi indirizzi non fa capo necessariamente una risorsa che possa essere consultabile immediatamente. Va poi osservato anche che, nella realizzazione pratica di RDF, si vedono solo indirizzi che fanno capo al protocollo HTTP.

Un soggetto (il primo elemento di una tripla) può possedere diverse proprietà, ognuna delle quali si esprime con una tripla indipendente, come si vede nella figura successiva.

Figura 56.4. Diverse triple relative a uno stesso soggetto.

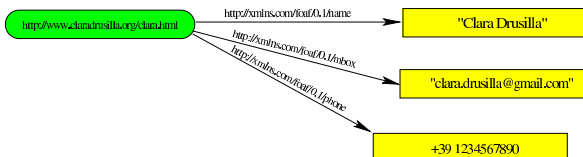


Da questo fatto si può incominciare a intravedere il grafo orientato che si può sviluppare attraverso le triple di RDF:



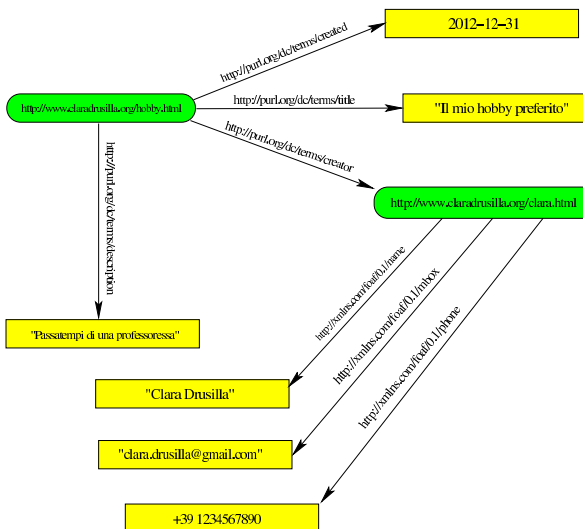
Quando un oggetto (il terzo elemento di una tripla) è costituito da un indirizzo URI, a sua volta questo può essere descritto ulteriormente, diventando soggetto di altre triple.

Figura 56.6. Grafo degli attributi di <http://www.claradrusilla.org/clara.html>.



In presenza di soggetti e oggetti corrispondenti alla stessa risorsa, i grafi possono innestarsi:

Figura 56.7. Innesto dei due grafi.



Rispetto agli esempi mostrati, va aggiunto il fatto che **la proprietà di un soggetto può apparire più volte, con valori diversi**. Per esempio, per indicare più autori di uno stesso soggetto, si può usare la proprietà <http://purl.org/dc/terms/creator> più volte.

56.3 Vocabolari

« Ciò che mette in relazione il soggetto con l'oggetto di una tripla (il primo e l'ultimo elemento), ovvero la proprietà, fa parte di un **vocabolario**. Negli esempi apparsi fino a questo punto, sono stati usati due vocabolari: <http://purl.org/dc/terms/> e <http://xmlns.com/foaf/0.1/>.

Di norma (ma non sempre), se si seguono gli indirizzi URI corrispondenti ai vocabolari rispettivi, attraverso un navigatore ipertestuale, si viene ridiretti alla guida sul loro utilizzo. In tal modo si può conoscere che significato dare a ogni «termine» da usare in qualità di proprietà di una tripla di RDF. Nel caso particolare degli esempi, si può osservare che il primo vocabolario è pensato per attribuire proprietà a ciò che rappresenta un prodotto editoriale (autore, titolo, data,...), mentre il secondo si presta per attribuire proprietà a ciò che rappresenta una persona (nome, indirizzo di posta elettronica, telefono, relazioni con altre persone,...).

Tabella 56.8. Alcuni vocabolari di uso comune.

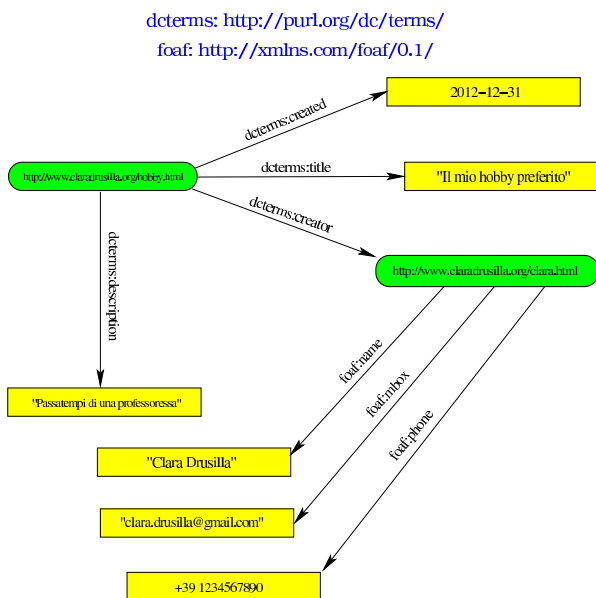
Sigla (common prefix)	URI	Descrizione
rdf	http://www.w3.org/1999/01/22-rdf-syntax-ns#	Vocabolario RDF.
foaf	http://xmlns.com/foaf/0.1/	<i>friend of a friend</i> Proprietà relative a risorse che rappresentano persone. L'acronimo si riferisce al fatto che questo vocabolario sia utile per attribuire proprietà a persone menzionate nei blog o nei <i>social network</i> (nel senso informatico del termine).
dcterms	http://purl.org/dc/terms/	<i>dublin core terms</i> Proprietà relative a risorse che rappresentano un prodotto editoriale.
dc	http://purl.org/dc/elements/1.1/	<i>dublin core elements</i> Proprietà relative a risorse che rappresentano un prodotto editoriale.
geo	http://www.w3.org/2003/01/geo/wgs84_pos#	Proprietà relative a risorse individuabili geograficamente attraverso delle coordinate.
schema	http://schema.org/	«HTML Microdata» trasformato in RDF (http://www.w3.org/TR/microdata-rdf/).

I vocabolari possono essere costruiti utilizzando il linguaggio «RDF Schema» (da non confondere con <http://schema.org/>). Questo aspetto della definizione dei vocabolari viene omissso nel capitolo ed eventualmente va approfondito nella documentazione ufficiale del consorzio W3C: www.w3.org/TR/rdf-schema/.

56.4 Abbreviazioni

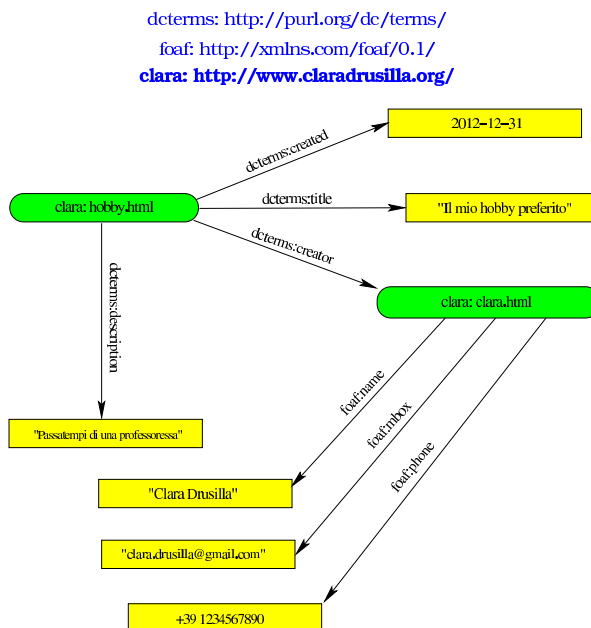
Le risorse che si utilizzano nella costruzione delle triple, possono essere abbreviate. Per fare questo si possono dichiarare delle abbreviazioni come appare nella tabella precedente; per esempio, al posto di scrivere <http://xmlns.com/foaf/0.1/phone> si scriverebbe *foaf:phone*, intendendo la stessa cosa.

Figura 56.9. Rappresentazione di un grafo di triple, abbreviando gli indirizzi delle proprietà.



Anche le risorse che si vanno a descrivere possono essere indicate attraverso delle abbreviazioni, scelte opportunamente in modo da non creare confusione, come si vede nella figura successiva.

Figura 56.10. Abbreviazione anche delle risorse che vengono descritte dalle triple.



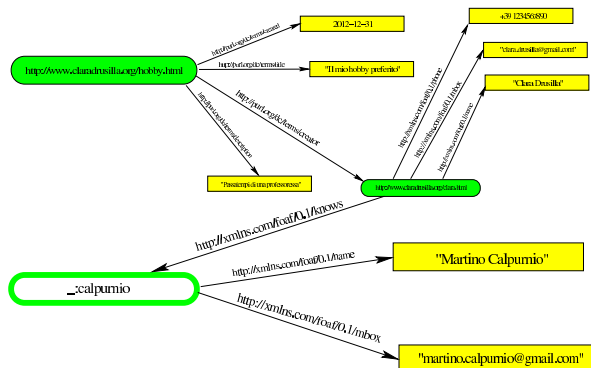
È importante osservare che i percorsi associati ai prefissi, devono essere indicati con attenzione. Per esempio, se il prefisso *foaf* fosse stato indicato come <http://xmlns.com/foaf/0.1>, senza la barra obliqua finale, l'indicazione della risorsa *foaf:name* produrrebbe un risultato errato: <http://xmlns.com/foaf/0.1name> al posto di <http://xmlns.com/foaf/0.1/name>.

Va anche osservato che per poter usare i prefissi, occorre dichiararli espressamente, nella modalità prevista dal contesto in cui si applica il linguaggio RDF. Comunque, in relazione ai vocabolari di proprietà, è sempre bene fare riferimento ai prefissi usati comunemente.

56.5 Nodi vuoti

Ci può essere la necessità o l'opportunità di utilizzare nodi «vuoti», nel senso che non sono identificati da un URI, ma sui quali si vogliono indicare delle proprietà. Per esempio, Clara Drusilla potrebbe voler annotare i dati di alcuni conoscenti nella sua pagina *hobby.html*, come si vede nella figura successiva.

Figura 56.11. Aggiunta di un conoscente con un nodo privo di URI.



Anche se il nodo che si aggiunge non è rappresentato da un URI, occorre dargli un nome provvisorio, in modo da poterlo trattare. Con-

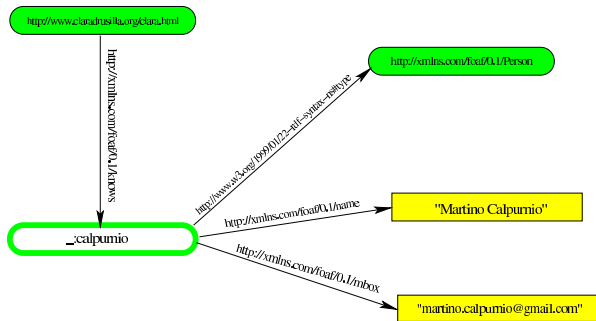
venzionalmente gli si attribuisce il prefisso di abbreviazione ‘_’ (trattino basso), a cui poi si aggiunge un nome qualunque, ma scelto opportunamente, come si vede nella figura. In pratica, in questo modo si definisce un URI inesistente, relativo a un prefisso non dichiarato.

56.6 Tipo di nodo o classe

Quando si dichiarano nodi vuoti, come descritto nella sezione precedente, può essere opportuno specificare che tipo di informazione rappresentano. Il «tipo» di un soggetto si definisce attraverso l'appartenenza a una «classe» secondo RDF. Le classi fanno parte della definizione dei vocabolari di proprietà e, convenzionalmente, i nomi delle classi iniziano con una lettera maiuscola, mentre i nomi delle proprietà hanno una iniziale minuscola.

Per dichiarare espressamente il tipo (la classe) di un soggetto, si usa la proprietà `http://www.w3.org/1999/01/22-rdf-syntax-ns#type`. Nel caso dell'esempio apparso nell'ultima figura, si vede la dichiarazione di un nodo vuoto, relativo a una persona conosciuta da Clara Drusilla: dal momento che questo nodo appare come valore della proprietà `http://xmlns.com/foaf/0.1/knowns` si determina implicitamente che deve trattarsi di una persona, secondo la classe `http://xmlns.com/foaf/0.1/Person`, ma lo si può rendere esplicito come si vede nella figura successiva.

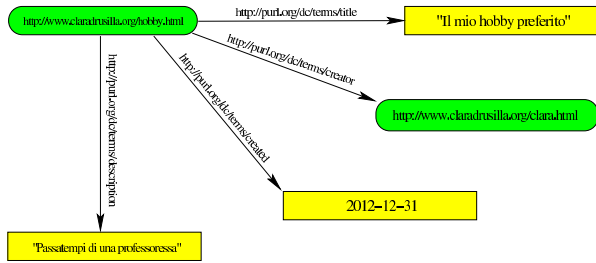
Figura 56.12. Specificazione del tipo di nodo.



56.7 RDFa

La sigla RDFa sta per «RDF in attributes», ovvero una forma di traduzione del linguaggio RDF attraverso attributi SGML/XML. Si tratta in pratica della modalità più semplice di incorporare questo tipo di informazioni in pagine HTML.

Gli attributi del linguaggio RDFa possono essere collocati in qualunque elemento HTML; naturalmente la scelta va fatta in base ai contenuti che si intendono descrivere. All'inizio del capitolo appare un esempio di codice RDFa, con il quale si descrive la pagina `http://www.claradrusilla.org/hobby.html` di Clara Drusilla:



```
<html>
<head>
  <meta http-equiv="Content-Type"
        property="http://purl.org/dc/terms/format"
        content="text/html; charset=UTF-8">
  <meta property="http://purl.org/dc/terms/title"
        content="Il mio hobby preferito">
  <meta property="http://purl.org/dc/terms/creator"
        content="clara.html">
  <meta property="http://purl.org/dc/terms/created"
        content="2012-12-31">
```

```
<meta property="http://purl.org/dc/terms/description"
      content="Passatempo di una professoressa">
<meta property="http://purl.org/dc/terms/subject"
      content="hobby, tempo libero, mare, spiaggia, sole">
<title>Il mio hobby preferito</title>
</head>
```

Dal momento che le informazioni riguardano la pagina nel suo complesso, il codice RDFa viene inserito opportunamente nell'intestazione HTML. Si può vedere l'uso dell'attributo 'property', con il quale si indica la risorsa che identifica la proprietà, e l'uso dell'attributo 'content' con il quale si specifica il valore della proprietà. Si osserva che il «soggetto» non viene dichiarato espressamente, pertanto si riferisce alla pagina stessa, ovvero `http://www.claradrusilla.org/hobby.html`.

L'uso dell'attributo 'content' si è reso necessario perché i valori delle proprietà non potevano essere contenuti diversamente: gli elementi 'META' sono elementi vuoti. Supponendo che la pagina degli hobby di Clara Drusilla mostri parte di queste informazioni al lettore, sarebbe stato possibile organizzare gli attributi RDFa in maniera differente:

```
<html>
<head>
  <meta http-equiv="Content-Type"
        property="http://purl.org/dc/terms/format"
        content="text/html; charset=UTF-8">
  <meta property="http://purl.org/dc/terms/creator"
        content="clara.html">
  <meta property="http://purl.org/dc/terms/subject"
        content="hobby, tempo libero, mare, spiaggia, sole">
  <title property="http://purl.org/dc/terms/title">Il mio
    hobby preferito</title>
</head>
<body>
  <h1 property="http://purl.org/dc/terms/description">Passatempo
    di una professoressa</h1>
  <p property="http://purl.org/dc/terms/created">2012-12-31</p>
  <p><a href="clara.html">Clara Drusilla</p>
```

In tal modo, il titolo principale, la descrizione della pagina e la data di creazione sono stati presi dai contenuti degli elementi, senza bisogno di inserirli in attributi 'content'.

Per dichiarare espressamente il «soggetto», ovvero la risorsa a cui si sta facendo riferimento, si può usare l'attributo 'resource'. In questo caso viene collocata nell'elemento 'HTML', dato che si tratta effettivamente della pagina:

```
<html resource="hobby.html">
```

Supponendo che Clara Drusilla consenta di copiare la sua pagina in altri siti, ma preferisca che la risorsa a cui si fa riferimento sia sempre quella originale, l'indicazione potrebbe avvenire, opportunamente, in modo completo:

```
<html resource="http://www.claradrusilla.org/hobby.html">
```

Logicamente, l'attributo 'resource' diventa indispensabile quando ci si riferisce a qualcosa che è descritto in un file differente da quello in cui appare; per esempio in una bibliografia, dove si indicano dati editoriali riferiti a documenti esterni. Il campo di azione dell'attributo 'resource' riguarda l'elemento in cui appare; pertanto appare frequentemente in elementi 'DIV' e 'SPAN', all'interno dei quali si descrive la risorsa attraverso proprietà e valori.

Per evitare di dover indicare le proprietà attraverso percorsi completi, è possibile specificare il vocabolario RDF attraverso l'attributo 'vocab':

```
<html vocab="http://purl.org/dc/terms/">
<head>
  <meta http-equiv="Content-Type" property="format"
        content="text/html; charset=UTF-8">
  <meta property="creator" content="clara.html">
  <meta property="subject" content="hobby, tempo libero,
    mare, spiaggia, sole">
  <title property="title">Il mio hobby preferito</title>
```

```
</head>
<body>
<h1 property="description">Passatempi di una
professoressa</h1>
<p property="created">2012-12-31</p>
<p><a href="clara.html">Clara Drusilla</p>
```

In questo caso, tutta la pagina utilizza un solo vocabolario, quindi l'attributo **'vocab'** è stato collocato nell'elemento **'HTML'** complessivo. A ogni modo, le eccezioni al vocabolario predefinito potrebbero essere trattate utilizzando dei percorsi completi, come nell'esempio seguente che si riferisce alla pagina *'clara.html'*, nella quale Clara Drusilla mette i propri dati:

```
<html vocab="http://purl.org/dc/terms/">
<head>
<meta http-equiv="Content-Type" property="format"
content="text/html; charset=UTF-8">
<meta property="creator" content="clara.html">
<meta property="created" content="2012-12-31">
<meta property="subject" content="dati personali">
<title property="title">Clara Drusilla</title>
</head>
<body>
<h1 property="description">Scheda personale</h1>
<p>Chi sono: <span
property="http://xmlns.com/foaf/0.1/name">Clara
Drusilla</span></p>
<p>email: <span property="http://xmlns.com/foaf/0.1/mbox">
clara.drusilla@gmail.com</span></p>
<p>tel: <span
property="http://xmlns.com/foaf/0.1/phone">
+39 1234567890</span></p>
```

Quando si usa l'attributo **'vocab'** si crea però ambiguità in merito alla risorsa a cui si sta facendo riferimento; pertanto, anche se si tratta della stessa pagina, è bene specificare anche l'attributo **'resource'**, eventualmente vuoto se si vuole proprio la stessa pagina, senza dover sapere come si chiama o dove si colloca:

```
<html resource="" vocab="http://purl.org/dc/terms/">
```

Con l'accortezza di specificare ogni volta anche la risorsa, è possibile dichiarare un vocabolario diverso nell'area che interessa. In questo caso si utilizza un elemento **'DIV'** per isolare la zona:

```
<html resource="" vocab="http://purl.org/dc/terms/">
<head>
<meta http-equiv="Content-Type" property="format"
content="text/html; charset=UTF-8">
<meta property="creator" content="clara.html">
<meta property="created" content="2012-12-31">
<meta property="subject" content="dati personali">
<title property="title">Clara Drusilla</title>
</head>
<body>
<h1 property="description">Scheda personale</h1>
<div resource="" vocab="http://xmlns.com/foaf/0.1/">
<p>Chi sono: <span property="name">Clara Drusilla</span></p>
<p>email: <span
property="mbox">clara.drusilla@gmail.com</span></p>
<p>tel: <span property="phone">+39 1234567890</span></p>
</div>
```

L'uso dell'attributo **'vocab'** consente di non doversi avvalere di abbreviazioni; tuttavia si può fare anche questo, con l'attributo **'prefix'**:

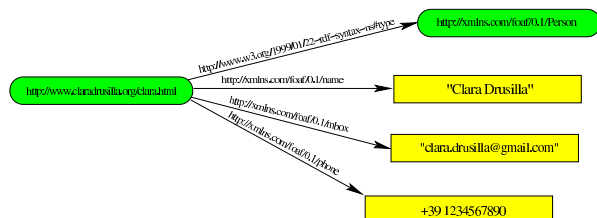
```
<html resource="" prefix="dcterms: http://purl.org/dc/terms/
foaf: http://xmlns.com/foaf/0.1/">
<head>
<meta http-equiv="Content-Type" property="dcterms:format"
content="text/html; charset=UTF-8">
<meta property="dcterms:creator" content="clara.html">
<meta property="dcterms:created" content="2012-12-31">
<meta property="dcterms:subject"
content="dati personali">
<title property="dcterms:title">Clara Drusilla</title>
</head>
<body>
```

```
<h1 property="dcterms:description">Scheda personale</h1>
<p>Chi sono: <span
property="foaf:name">Clara Drusilla</span></p>
<p>email: <span
property="foaf:mbox">clara.drusilla@gmail.com</span></p>
<p>tel: <span
property="foaf:phone">+39 1234567890</span></p>
```

Quando si presenta la necessità o l'opportunità di specificare la classe (il tipo) di appartenenza della risorsa a cui si fa riferimento, si può utilizzare l'attributo **'typeof'**, senza bisogno di indicare la proprietà *http://www.w3.org/1999/01/22-rdf-syntax-ns#type*. Anche in occasione dell'uso dell'attributo **'typeof'** è bene precisare la risorsa a cui ci si riferisce, altrimenti si rischia di far intendere che si tratti di un nodo vuoto:

```
<div resource="" vocab="http://xmlns.com/foaf/0.1/"
type="Person">
<p>Chi sono: <span property="name">Clara Drusilla</span></p>
<p>email: <span
property="mbox">clara.drusilla@gmail.com</span></p>
<p>tel: <span property="phone">+39 1234567890</span></p>
</div>
```

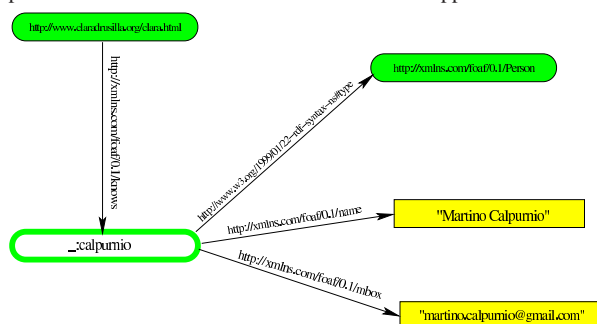
Nell'esempio si vede che nell'area controllata dall'elemento **'DIV'**, le proprietà si riferiscono a una risorsa di tipo *Person*. Per la precisione, la classe che definisce il tipo è *http://xmlns.com/foaf/0.1/Person*, essendo stato dichiarato il vocabolario poco prima:



Nell'esempio successivo si vede la creazione di un nodo vuoto in relazione all'indicazione di un conoscente di Clara Drusilla:

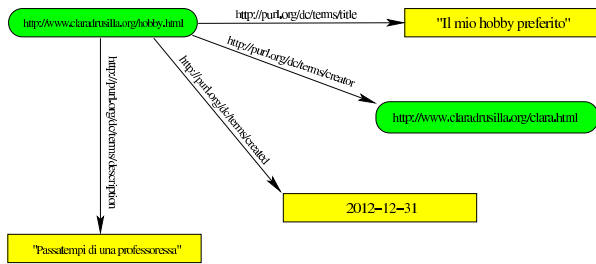
```
<p vocab="http://xmlns.com/foaf/0.1/" property="knows"
type="Person">Il mio migliore amico è <span
property="name">Martino Calpurnio</span> che può essere
contattato all'indirizzo <span
property="mbox">martino.calpurnio@gmail.com</span>.</p>
```

In questo caso, supponendo che prima del codice mostrato si stesse facendo riferimento alla risorsa *http://www.claradrusilla.org/clara.html*, si intende che questa risorsa «conosce» un nodo vuoto, di tipo *http://xmlns.com/foaf/0.1/Person*, il quale ha il nome e l'indirizzo di posta elettronica delimitati con elementi **'SPAN'** appositi:



56.8 N-Triples

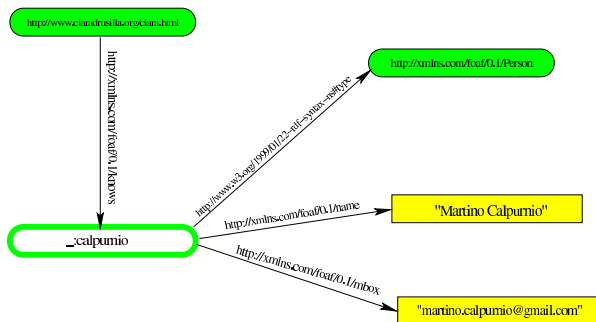
N-Triples è una modalità di rappresentazione molto semplice delle triple RDF; si tratta di righe contenenti i tre elementi consueti: soggetto, predicato e complemento, con la particolarità che si usano le parentesi uncinate, *< e >*, per delimitare soggetto, predicato ed eventualmente anche l'oggetto quanto questo non è espresso in forma costante. L'esempio seguente mostra la trasformazione di un grafo RDF in N-Triples:



```
# questo è un commento
<http://www.claradrusilla.org/hobby.html> <http://purl.org/dc/terms/title>
  "Il mio hobby preferito" .
<http://www.claradrusilla.org/hobby.html> <http://purl.org/dc/terms/creator>
  <http://www.claradrusilla.org/clara.html> .
<http://www.claradrusilla.org/hobby.html> <http://purl.org/dc/terms/created>
  "2012-12-31" .
<http://www.claradrusilla.org/hobby.html> <http://purl.org/dc/terms/description>
  "Passatempi di una professoressa" .
```

Le righe che compongono le triple, possono essere spezzate, proprio come si vede nell'esempio: in ogni caso, il punto fermo finale determina la conclusione della riga.

I nodi vuoti vengono dichiarati espressamente, attraverso una notazione del tipo `'_:nome'`. L'esempio seguente traduce il grafo visibile dalla forma grafica alla forma N-Triples:

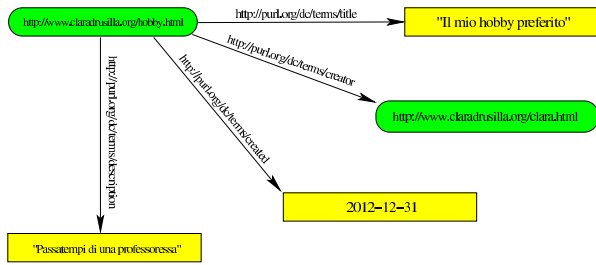


```
# Viene creato un nodo vuoto:
<http://www.claradrusilla.org/clara.html>
  <http://xmlns.com/foaf/0.1/knows> _:calpurnio .
# Vengono definite le proprietà del nodo vuoto:
_:calpurnio <http://www.w3.org/1999/01/22-rdf-syntax-ns#type>
  <http://xmlns.com/foaf/0.1/Person> .
_:calpurnio <http://xmlns.com/foaf/0.1/name>
  "Martino Calpurnio" .
_:calpurnio <http://xmlns.com/foaf/0.1/mbox>
  "martino.calpurnio@gmail.com" .
```

Si osservi che i valori costanti appaiono sempre delimitati come stringhe: per poter specificare il linguaggio o indicare che si tratta invece di valori diversi (per esempio numerici), occorre aggiungere un'estensione. Si veda eventualmente la documentazione elencata alla fine del capitolo.

56.9 Turtle

Turtle sta per *terse RDF triple language* e corrisponde a una modalità di rappresentare il linguaggio RDF in forma testuale, in modo più compatto rispetto a N-Triples. Di fatto, Turtle è un'estensione di N-Triples. Viene ripreso il primo esempio mostrato a proposito di N-Triples, il quale è valido anche per Turtle:



```
# questo è un commento
<http://www.claradrusilla.org/hobby.html> <http://purl.org/dc/terms/title>
  "Il mio hobby preferito" .
<http://www.claradrusilla.org/hobby.html> <http://purl.org/dc/terms/creator>
  <http://www.claradrusilla.org/clara.html> .
<http://www.claradrusilla.org/hobby.html> <http://purl.org/dc/terms/created>
  "2012-12-31" .
<http://www.claradrusilla.org/hobby.html> <http://purl.org/dc/terms/description>
  "Passatempi di una professoressa" .
```

Con Turtle si possono però annotare dei prefissi, così come si può fare con RDFa. L'esempio appena mostrato si potrebbe semplificare nel modo seguente, dopo aver dichiarato i prefissi *clara* e *dcterms*:

```
@prefix clara: http://www.claradrusilla/ .
@prefix dcterms: http://purl.org/dc/terms/ .
clara:hobby.html dcterms:title "Il mio hobby preferito" .
clara:hobby.html dcterms:creator clara:clara.html .
clara:hobby.html dcterms:created "2012-12-31" .
clara:hobby.html dcterms:description
  "Passatempi di una professoressa" .
```

Al posto del prefisso è possibile indicare risorse con percorsi relativi:

```
@base http://www.claradrusilla/ .
@prefix dcterms: http://purl.org/dc/terms/ .
<hobby.html> dcterms:title "Il mio hobby preferito" .
<hobby.html> dcterms:creator clara:clara.html .
<hobby.html> dcterms:created "2012-12-31" .
<hobby.html> dcterms:description
  "Passatempi di una professoressa" .
```

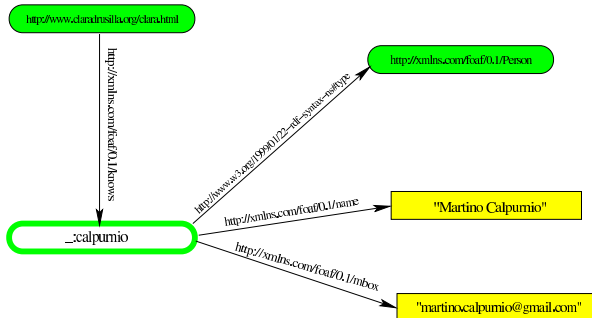
Quando un gruppo di triple condivide lo stesso soggetto, è possibile semplificarle usando il punto e virgola, come si vede nella variante successiva:

```
@base http://www.claradrusilla/ .
@prefix dcterms: http://purl.org/dc/terms/ .
<hobby.html> dcterms:title "Il mio hobby preferito" ;
  dcterms:creator clara:clara.html ;
  dcterms:created "2012-12-31" ;
  dcterms:description
    "Passatempi di una professoressa" .
```

Se un gruppo di triple condivide lo stesso soggetto e lo stesso predicato, è possibile semplificarle usando la virgola, come si vede nella variante successiva, dove si immagina di avere due descrizioni alternative per la pagina degli hobby della professoressa Clara Drusilla:

```
@base http://www.claradrusilla/ .
@prefix dcterms: http://purl.org/dc/terms/ .
<hobby.html> dcterms:title "Il mio hobby preferito" ;
  dcterms:creator clara:clara.html ;
  dcterms:created "2012-12-31" ;
  dcterms:description
    "Passatempi di una professoressa" ,
    "Rilassarsi all'aria aperta" .
```

I nodi vuoti vengono annotati nello stesso modo già visto a proposito di N-Triples, ma con le estensioni di Turtle:



```
@base http://www.claradrusilla/ .
@prefix foaf: http://xmlns.com/foaf/0.1/ .
# Viene creato un nodo vuoto:
<clara.html> foaf:knows _:calpurnio .
# Vengono definite le proprietà del nodo vuoto:
_:calpurnio <http://www.w3.org/1999/01/22-rdf-syntax-ns#type>
  foaf:Person ;
  foaf:name "Martino Calpurnio" ;
  foaf:mbox "martino.calpurnio@gmail.com" .
```

Con Turtle si può rimpiazzare la proprietà <http://www.w3.org/1999/01/22-rdf-syntax-ns#type> con la sigla 'a':

```
@base http://www.claradrusilla/ .
@prefix foaf: http://xmlns.com/foaf/0.1/ .
# Viene creato un nodo vuoto:
<clara.html> foaf:knows _:calpurnio .
# Vengono definite le proprietà del nodo vuoto:
_:calpurnio a foaf:Person ;
             foaf:name "Martino Calpurnio" ;
             foaf:mbox "martino.calpurnio@gmail.com" .
```

56.10 Riferimenti

<<

- Network Working Group, *RFC 3987: Internationalized Resource Identifiers (IRIs)*, 2005, <http://www.ietf.org/rfc/rfc3987.txt>
- W3C, *RDF*, 2004, <http://www.w3.org/TR/rdf-syntax/>, <http://www.w3.org/TR/rdf-concepts/>, <http://www.w3.org/TR/rdf-syntax-grammar/>, <http://www.w3.org/TR/rdf-ml/>, <http://www.w3.org/TR/rdf-schema/>, <http://www.w3.org/TR/rdf-testcases/>
- W3C, *RDFa*, 2012, <http://www.w3.org/TR/rdfa-primer/>, <http://www.w3.org/TR/rdfa-core/>, <http://www.w3.org/TR/rdfa-lite/>, <http://www.w3.org/TR/rdfa-in-html/>, <http://www.w3.org/TR/xhtml-rdfa/>
- W3C, *Turtle*, <http://www.w3.org/TR/turtle/>
- W3C, *RDFa validator*, <http://www.w3.org/2012/pyRdfa/Validator.html>
- RDF:about, *RDF validator and converter*, <http://www.rdfabout.com/demo/validator/>
- W3C, *SPARQL*, <http://www.w3.org/TR/rdf-sparql-query/>, <http://www.w3.org/TR/sparql11-query/>