Esercizio 1

Scrivere in JSON-LD i seguenti statement RDF senza utilizzare alcun contesto ("@context" è vietato):

```
<http://www.example.com/me>
      <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
            <http://xmlns.com/foaf/0.1/Person> .

<http://www.example.com/me>
      <http://xmlns.com/foaf/0.1/givenName>
            "John" .

<http://www.example.com/me>
      <http://xmlns.com/foaf/0.1/familyName>
            "Doe" .

<http://www.example.com/me>
      <http://xmlns.com/foaf/0.1/mbox>
            <mailto:john.doe@example.com> .

<http://www.example.com/me>
      <http://xmlns.com/foaf/0.1/knows>
            <http://www.example.com/you> .

<http://www.example.com/me>
      <http://schema.org/birthDate>
            "1981-01-15"^^<http://www.w3.org/2001/XMLSchema#date> .
```

Esercizio 2

Scrivere in JSON-LD gli statement precedenti usando **un unico contesto** definito internamente (ovvero: non si può fare riferimento a un file esterno), in modo da:

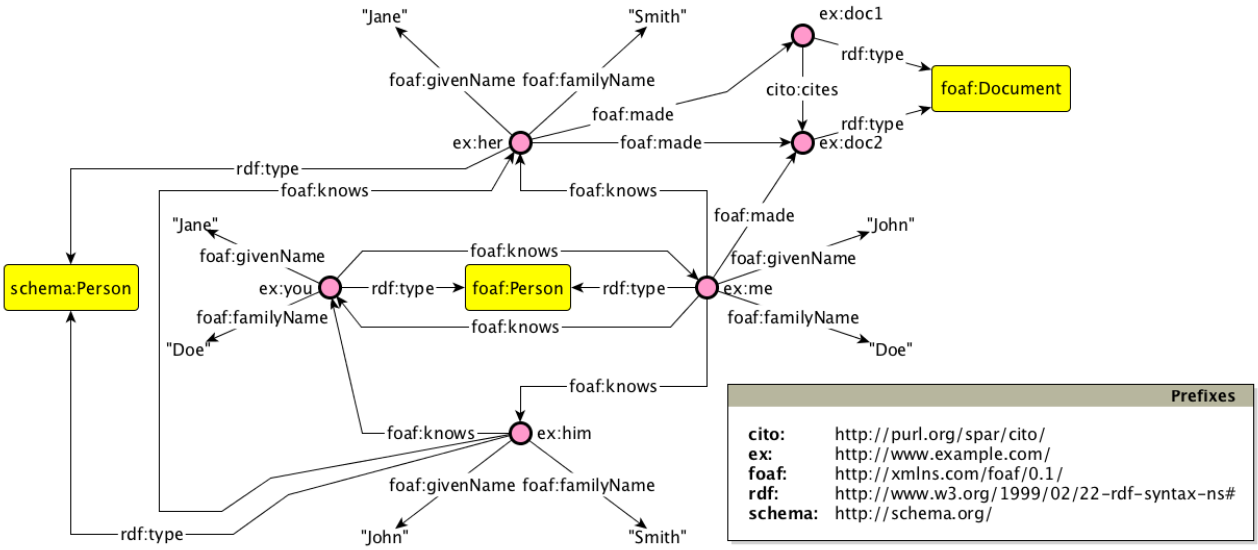
- l'IRI “http://www.example.com/” deve essere abbreviato con il prefisso “ex”;
- utilizzare le seguenti chiavi: nome, cognome, email, amico_di, data_di_nascita;
- per gli statement che prevedono un valore tipato come oggetto, evitare l'uso di “@value” e “@type” (sono ammessi solo nel contesto);
- per gli statement “email” evitare l'uso “@id” (è ammesso solo nel contesto);
- nascondere l'IRI del tipo associato in modo da usare soltanto la stringa “persona”.

Inoltre, aggiungere le seguenti informazioni relative a “http://www.example.com/you”:

- Nome: Jane
- Cognome: Doe
- Tipo: persona
- Email: jane.doe@example.com
- Amico di: http://www.example.com/me
- Data di nascita: 29 Marzo 1982

Esercizio 3

Dato il diagramma qui sotto, scrivere un array JSON-LD contenente tanti oggetti JSON quante sono le risorse da descrivere (i pallini viola nel diagramma).

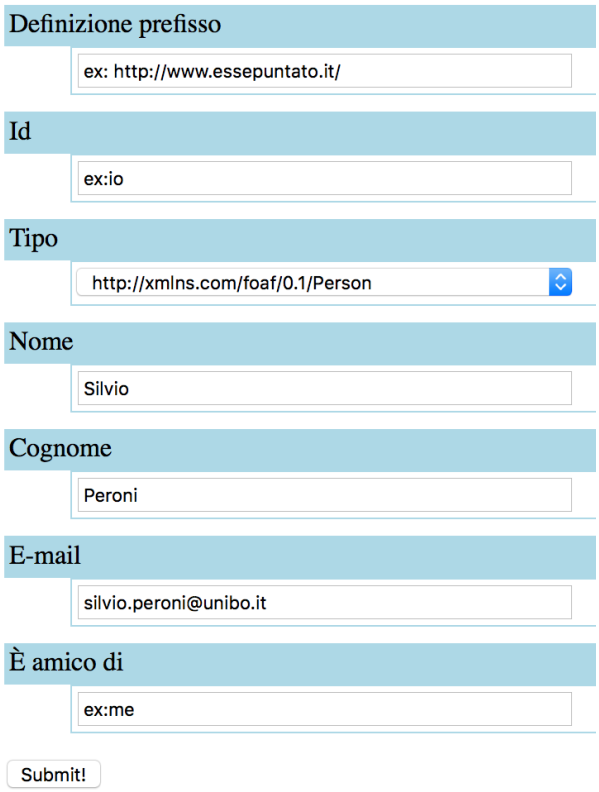


Le varie coppie chiave-valore devono essere conformi ai vari prefissi definiti nel diagramma. Ad esempio, non potete scrivere `'@id': 'http://www.example.com/me'` o `'@type': 'persona'`, ma dovete scrivere `'@id': 'ex:me'` e `'@type': 'foaf:Person'`.

In più tutti i contesti devono fare riferimento a un file esterno “context.json” che dovete creare appositamente.

Esercizio 4

Creare un form HTML come quello mostrato nella figura sottostante, che al seguito di un click sul bottone “Submit!” chiami una funzione Javascript “go()” che prende tutti i valori dei vari elementi del form e chiama la funzione “alert” con mostrando il JSON-LD relativo prodotto analizzando i vari campi.



Un solo prefisso può essere definito a piacere e deve essere usato per semplificare la scrittura dell'identificativo e della risorsa che viene riferita come amico. Gli IRI di tutte le altre proprietà sono gli stessi utilizzati negli esercizi precedenti.

L'identificativo e il tipo della risorsa devono essere specificati obbligatoriamente – in caso contrario, un errore esplicativo deve essere restituito. Tutti gli altri campi sono facoltativi, e se non sono specificati non vanno inclusi nel JSON finale. Inoltre non si deve fare uso di alcuna libreria Javascript/CSS aggiuntiva (JQuery, Bootstrap, etc.).

Esercizio 5

Utilizzando qualsiasi framework Javascript/CSS (ad esempio Jquery e Bootstrap), mettere a disposizione un'interfaccia Web che permetta di creare gli statement RDF indicati nell'esercizio precedente.

Una volta cliccato sul bottone “Submit!”:

1. viene chiamata la funzione “go()” che manda in POST (AJAX, asincrono) allo script “server.py” il JSON-LD ottenuto analizzando i dati del form;
2. lo script “server.py” salva il JSON-LD su un file locale, senza sovrascriverne uno esistente;
3. lo script “server.py” restituisce un frammento HTML descrittivo dei dati che sono stati salvati;
4. il frammento HTML viene ricevuto lato client e appeso come primo elemento dentro un contenitore `div` con *id* “risorse”;
5. il form viene svuotato, in modo che si possa inserire una nuova risorsa.