

Tecnologie Web

C.d.L. in Informatica e Informatica per il Management

Compito del 6 settembre 2013

Nome:

Cognome:

Matricola:

C.d.L.:

Team:

Corso seguito nell'A.A.:

Non è la prima volta. Data indicativa dell'ultimo appello provato: _____

Importante:

- Indicare ciascun esercizio risolto con una croce sul numero relativo. Saranno corretti solo gli esercizi così segnati.
- Se si consegna un foglio protocollo, scrivere in maniera chiara il numero di ogni esercizio presso la sua soluzione e, in intestazione, indicare nome, cognome e numero di matricola.

Esercizio		Punti	Voto
1	Domande di base	12	
2	HTML	6	
3	Javascript	6	
4	Semantic Web	6	
5	Teoria	4	
Totale		34	

1. Domande di base (12 punti)

Rispondere correttamente a tre delle seguenti domande:

A. Scrivere gli URL assoluti di tutte le risorse esterne incluse nella seguente pagina HTML, il cui URL è `http://www.cs.unibo.it/en/courses/index.html`:

```
<html>
  <head>
    <link rel='stylesheet' href="/style/layout.css" type="text/css">
    <link rel='stylesheet' href="../../style/layout.css" type="text/css">
  </head>
  <body>
    <a href='../..//slides/index.html'>Slides</a>
    
  </body>
</html>
```

B. Spiegare brevemente la seguente regola CSS:

```
div.sidebar p{
font-size: 11pt;
}
```

C. Tradurre il seguente frammento XML in JSON:

```
<hotels>
  <hotel>
    <nome>Roma</nome>
    <indirizzo>Via Roma 23</indirizzo>
  </hotel>
  <hotel>
    <nome>Miramare</nome>
    <indirizzo>Lungomare C. Colombo 18</indirizzo>
  </hotel>
</hotels>
```

D. Scrivere l'output del seguente foglio di stile XSLT, se applicato al precedente documento XML:

```
<xsl:template match="hotels">
  <alberghi/>
</xsl:template>
```

2. HTML (6 punti)

Scrivere il codice HTML (e CSS) di un documento che rappresenti il contenuto visualizzato nell'immagine seguente:

<ol style="list-style-type: none">1. Debian Linux2. Debian Red Hat3. Mac OS X4. Windows XP	<p>Name: Mac OS X</p> <p>Software house: Apple</p> <p>Logo:</p>  <p>Comments:</p> <p>Do you use Mac OS X? Yes <input type="radio"/> No <input type="radio"/></p> <p>If yes, what for? <input type="text" value="---"/></p> <p>If no, briefly explain why: <input type="text"/></p> <p><input type="button" value="Submit"/> <input type="button" value="Reset"/></p>
---	--

Nello scrivere il codice si tengano in considerazione questi vincoli:

- gli elementi `table`, `th` e `td` non possono essere usati
- gli elementi presentazionali `b` e `i` non possono essere usati (basta usare gli elementi semantici più adatti e del CSS);
- nessun elemento deve contenere l'attributo `style`;
- la destinazione dei link non è rilevante
- i dati devono essere spediti allo script `/os.php`
- alla domanda "If yes, what for?" è possibile rispondere con i valori "---", "Personal", "Business" e "School"
- Si richiede di scrivere tutto il codice CSS in un file separato, non in elementi `<style>`.

3. Javascript (6 punti)

Riscrivere l'output dell'esercizio 2 (HTML) precedente come segue:

La pagina HTML contiene in realtà solo due elementi, uno a destra e uno a sinistra, vuoti, e del codice javascript che svolge le seguenti attività:

A (2 crediti): nel momento più adatto della fase di accesso - caricamento - visualizzazione della pagina, si acceda al servizio server-side `http://www.osdata.com/oslist.php`. Tale servizio restituisce un JSON come il seguente:

```
{osdata: [
  {
    id: 'debian',
    name: 'Debian Linux',
    swhouse: 'Debian community',
    logo: 'debian.jpg',
  },
  {
    id: 'redhat',
    name: 'Debian Red Hat',
    swhouse: 'Red Hat',
    logo: 'redhat.jpg',
  },
  {
    id: 'mac',
    name: 'Mac OS X',
    swhouse: 'Apple',
    logo: 'apple.jpg',
  },
  {
    id: 'xp',
    name: 'Windows XP',
    swhouse: 'Microsoft',
    logo: 'XP.jpg',
  }
]}
```

che va usato per la creazione della lista a sinistra. Cliccando su una delle voci a sinistra i dati corrispondenti vengono usati per riempire lo schema a destra come da esempio.

B (2 crediti): Per contrastare il crescente trend detto "lurking", secondo il quale molti utenti leggono i commenti altrui ma non forniscono una propria opinione, si è deciso che la lista di commenti venga visualizzata solo dopo che l'utente ha inserito il proprio. A tale fine, il pulsante di submit del form spedisce le informazioni del form in un blocco JSON (attenzione! JSON, non x-form-urlencoded!) nel body di una connessione HTTP al servizio server-side `http://www.osdata.com/getComments.php?os=[id]`, dove id è il campo id di uno dei sistemi operativi ricevuti precedentemente.

C (2 crediti): vd. pagina successiva

C (2 crediti): Il servizio `getComments` precedentemente citato restituisce un JSON come il seguente:

```
{
  os: 'mac',
  comments: [
    {
      id: 'mac2013-0345',
      from: 'John Smith',
      mail: 'smith@yahoo.com',
      inReplyTo: '',
      date: '20130904',
      time: '1205',
      useofOS: false,
      usefor: 'business',
      comment: 'Too expensive?'
    },
    ...,
    {
      id: 'mac2013-0987',
      from: 'Ted Brown',
      mail: 'brown@yahoo.com',
      inReplyTo: 'mac2013-0345',
      date: '20130905',
      time: '1912',
      useofOS: true,
      usefor: 'school',
      comment: 'Too expensive? Possibly, but worth it.'
    },
    ...
  ]
}
```

Alla ricezione di tale risposta, si crei una visualizzazione interessante e utile tale per cui il campo `inReplyTo` (che contiene l'id del messaggio di cui questo è risposta) sia adeguatamente utilizzato (con una organizzazione e uno stile a vostra scelta purché utile e plausibile).

4. Semantic Web (6 punti)

Considerate la seguente descrizione di un vocabolario RDF in linguaggio naturale:

Le entità *swc:Conference* e *foaf:Person* sono due particolari risorse di **tipo** (*rdf:type*) classe (*owl:Class*).

La risorsa *foaf:name* è una **proprietà** (*rdf:Property*) che permette di specificare il nome sia di una persona sia di una conferenza, mentre la risorsa *part:participates* è un'altra proprietà che permette di specificare quali persone (soggetto dello statement RDF) partecipano ad una certa conferenza (oggetto dello statement RDF).

Ci sono, inoltre, cinque risorse di tipo *foaf:Person* - chiamate rispettivamente "Fabio Vitali", "Gioele Barabucci", "Angelo Di Iorio", "Silvio Peroni" e "Francesco Poggi" - e due risorse di tipo *swc:Conference* - chiamate rispettivamente "DocEng 2013" e "ESWC 2013".

Le prime tre persone della lista precedente partecipano alla prima conferenza, mentre le altre due alla seconda.

Rispondere alle seguenti domande:

1. In una sintassi a scelta tra RDF/XML e Turtle, tradurre la precedente descrizione in statement RDF (senza dover specificare il dominio e il codominio delle proprietà e senza definire alcun prefisso).
2. Scrivere una query SPARQL (senza definire alcun prefisso) che **restituisca**, in ordine alfabetico per nome, tutte le conferenze che hanno in comune almeno due partecipanti.
3. Utilizzando la libreria *SPARQL Javascript library*, scrivere un piccolo script che interroghi lo SPARQL end-point all'indirizzo "<http://www.sparql.org/example/query>" e che esegua la query scritta al punto precedente. Non è necessario né impostare i prefissi né l'output, e non è richiesta l'implementazione delle funzioni di callback da chiamare in caso di successo o errore nell'esecuzione della query.

5. Teoria (4 punti)

Spiegare perché sono stati introdotti i 'cookie' e descriverne i tipi principali. Indicare inoltre quali header HTTP usano.