

Prova Scritta di Tecnologie Web
C.d.S. in Informatica e
C.d.S. in Informatica per il Management
Compito del 13 Luglio 2017

Nome: _____

Cognome: _____

Matricola: _____

C.d.L.: _____

Team: _____

Corso seguito nell'A.A.: _____

Non è il primo scritto. Data indicativa dell'ultimo appello provato: _____

Ho già consegnato il progetto. Data indicativa: _____

Importante:

- Utilizzare solo i fogli protocollo consegnati. Su ogni foglio indicare in modo chiaro nome, cognome e numero di matricola.
- Sui fogli protocollo consegnati scrivere in maniera chiara il numero di ogni esercizio di cui si fornisce la soluzione. Eventuali cancellature vanno rese evidenti e incontrovertibilmente cancellate.
- Indicare ciascun esercizio risolto nella tabella qui sotto con una croce sul numero relativo. Saranno corretti solo gli esercizi così segnati.

Ho svolto l'esercizio?	Esercizio		Punti	Voto
	1	Domande di base	12	
	2	HTML + CSS	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.

N.B.: Scrivere le risposte direttamente su questo foglio.

A. Trovare l'intruso o gli intrusi nella seguente lista:

- URL
- CURIE
- IRL
- IRI
- URI
- URN
- URS

B. Scrivere una regola CSS per colorare in rosso lo sfondo dei collegamenti ipertestuali contenuti in una tabella.

C. Scrivere l'output prodotto dall'esecuzione del seguente script PHP:

```
<?php
$mese = "Luglio";
$a = "Fa troppo caldo a Luglio a Bologna";
echo substr($a, 0, strpos($a, $mese) - 2);
?>
```

D. Spiegare brevemente il significato dell'acronimo REST.

2. HTML (6 punti)

Dato il codice HTML mostrato di seguito e l'immagine che rappresenta la sua resa in un comune browser:

1. Individuare e correggere 4 errori nel sorgente. Gli errori si riferiscono soltanto ad elementi o attributi, e non riguardano il testo della pagina. **Indicare le correzioni direttamente sul foglio, senza ricopiare il sorgente.**
2. Scrivere il codice CSS per ottenere una visualizzazione quanto più possibile simile a quella mostrata in Figura 1. Si tengano in considerazione questi vincoli:
 - Deve essere considerata la struttura HTML corretta dal precedente punto
 - Non si può modificare la struttura HTML (spostare elementi, aggiungere attributi, etc.) in alcun modo
 - i contenuti sono 'statici' (i comportamenti dinamici sono gestiti nell'esercizio Javascript)
 - Le dimensioni esatte di margini, padding, bordi e dimensioni delle immagini non sono rilevanti
 - I colori esatti non sono rilevanti
 - Il bordo esterno della Figura 1 non va specificato
 - È possibile usare solo le seguenti regole CSS: background-color, border, border-radius, color, content, margin-left, margin, padding

Resa della pagina in un browser (Figura 1):

Posizioni di lavoro aperte

In questa pagina è possibile candidarsi per una delle proposte di lavoro che sono aperte. Basta specificare:

- il proprio nome;
- il proprio numero di telefono;
- il proprio indirizzo e-mail.

Inoltre, è necessario selezionare la posizione di lavoro desiderata, indicando se si preferisce un lavoro part-time o full-time.

Nome:

Telefono:

E-mail:

Seleziona posizione aperta: Part-time

Sorgente HTML:

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Candidati per un lavoro</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <link href="style.css" rel="stylesheet"/>
  </head>
  <body>
    <div1>
      <title>Posizioni di lavoro aperte</title>
      <p>In questa pagina è possibile candidarsi per una delle proposte di
        lavoro che sono aperte. Basta specificare:</p>
      <ul>
        <li><p>il proprio nome;</p></li>
        <li><p>il proprio numero di telefono;</p></li>
        <li><p>il proprio indirizzo e-mail.</p></li>
      </ul>
      <p>Inoltre, è necessario selezionare la posizione di lavoro
        desiderata, indicando se si preferisce un lavoro part-time o
        full-time.</p>
    </div1>
    <div2>
      <form encoding="applica.php">
        <p><label for="nome">Nome</label>
          <input id="nome" name="nome" type="text" /></p>
        <p><label for="tel">Telefono</label>
          <input id="nome" name="tel" type="tel" /></p>
        <p><label for="email">E-mail</label>
          <input id="email" name="email" type="email" /></p>
        <p>
          <label for="lavoro">Seleziona posizione aperta</label>
          <select id="lavoro">
            <option for="dasplab">
              Assegnista di ricerca presso il DASPLab</option>
            <option value="disi">
              Tecnico informatico presso il DISI</option>
            <option value="mulino">Curatore presso il Mulino</option>
          </select>
          <input id="partime" name="partime"
            type="checkbox" value="partime" />
          <label for="partime">Part-time</label>
        </p>
        <p><input type="submit" value="Candidati!" /></p>
      </form>
    </div2>
  </body>
</html>
```

3. Javascript (6 punti)

Usando come riferimento il documento HTML dell'esercizio precedente, e basandosi, dove si ritiene, su uno o più framework Javascript a piacere tra quelli illustrati a lezione, si descrivano:

- a) un'API REST che fornisce due servizi:
- *getPositions.php*, che restituisce l'elenco delle posizioni di lavoro aperte in un intervallo di tempo specificato. Non tutte le posizioni ammettono il part-time e questo deve trovare supporto nel formato dati. Il formato dati deve specificare anche la data di apertura e di chiusura delle domande ammissibili.
 - *applyToPosition.php*, che associa i dati specificati nel form ad una posizione di lavoro aperta. l'API richiede anche di specificare la data e l'ora in cui la domanda è stata fatta. Si tratta di due campi diversi, uno con valori del tipo *yyyy-mm-dd* e l'altro con valori del tipo *hh:mm:ss*.

Per ogni servizio si specifichi l'URL di accesso, il metodo HTTP utilizzato, gli eventuali parametri e la sintassi usata e il formato delle strutture dati trasmesse e/o ricevute (anche attraverso un semplice esempio), le tipologie di errore e il messaggio di errore ritornato in ciascun caso. *Il formato dei dati è necessariamente JSON. In nessun caso si potranno passare frammenti, anche parziali, in HTML. Si noti che entrambi i servizi si basano su date e ore in maniera precisa e specifica, e che il form nella pagina web NON HA CAMPI PER QUESTE INFORMAZIONI.*

- b) Uno o più script Javascript che, subito prima di visualizzare la pagina all'utente, invochino il servizio *getPositions.php* (con gli eventuali parametri correttamente inizializzati) e popolino correttamente il menu delle posizioni aperte. Si espliciti tutto il meccanismo di invocazione, di modifica della pagina quando vengono ricevuti i dati, e di gestione degli eventuali errori. Si mostrino in maniera differente le posizioni chiuse, le posizioni con meno di 24 ore alla chiusura, e le posizioni aperte¹. Se la posizione ammette il part-time, il relativo checkbox è attivo, altrimenti deve essere disattivo.
- c) Uno o più script Javascript che vengano attivati cliccando sul pulsante "Candidati", e che invochino il servizio *applyToPosition.php* (con i parametri correttamente inizializzati) *dopo aver controllato che tutti i valori del form siano correttamente riempiti. A conclusione positiva della chiamata, il browser verrà mandato ad una pagina chiamata "positionApplied.html"*. Si espliciti tutto il meccanismo di invocazione, di gestione della risposta sia in caso positivo sia in caso di eventuali errori, in tutti i tipi possibili di errore. Si noti che in sottomissione dei dati è necessario la data e l'ora in cui è stata sottomessa la domanda.

¹ Si noti che nei browser moderni è possibile modificare lo stile di una voce di un menù a tendina solo relativamente a colore del testo e colore dello sfondo.

4. Semantic Web (6 punti)

Considerate il seguente frammento di JSON-LD

```
{
  "@id": "http://www.example.com/john-doe",
  "@type": "http://xmlns.com/foaf/0.1/Person",
  "http://xmlns.com/foaf/0.1/name": "John Doe",
  "http://xmlns.com/foaf/0.1/age": {
    "@value": "42",
    "@kind": "http://www.w3.org/2001/XMLSchema#nonNegativeInteger"
  }
}
"http://xmlns.com/foaf/0.1/knows" = [
  { "@id": "http://www.example.com/charlie-brown" },
  { "@id": "http://www.example.com/jane-doe" }
]
}
```

Rispondere alle seguenti domande:

1. Individuare tutti gli errori presenti nel frammento JSON-LD e proporre le opportune correzioni.
2. Considerando il JSON-LD corretto, quanti sono gli statement che hanno come oggetto delle risorse RDF e quanti sono gli statement che hanno come oggetto dei letterali?
3. Definire un contesto appropriato in modo che il JSON-LD corretto precedentemente possa essere rappresentato come segue:

```
{
  "@context": { ... il contesto da definire ... }
  "@id": "john-doe",
  "@type": "person",
  "name": "John Doe",
  "age": "42",
  "knows": [ "charlie-brown", "jane-doe" ]
}
```

5. Teoria (4 punti)

Rispondere alle seguenti domande relative alla codifica dei caratteri:

- Descrivere le differenze tra *UTF-8* e *ISO-Latin1*
- Descrivere le differenze tra *content-encoding* e *content-escaping*