

# Linguaggi e tecnologie per il Web – appello del 10/7/2017

COGNOME: .....
NOME: .....
MATRICOLA: .....

Autorizzo la pubblicazione del mio voto di questo esame sul sito web <a href="http://www.dis.uniroma1.it/~rosati/lw">http://www.dis.uniroma1.it/~rosati/lw</a> , secondo quanto prevede il decreto legislativo 196/2003 (codice in materia di protezione dei dati personali) che dichiaro di conoscere. In fede, .....
---

## Esercizio 1

(a) Scrivere un documento HTML contenente una form contenente i seguenti campi:

- codice fiscale (casella di testo editabile lunga 16 caratteri)
- cognome e nome (casella di testo editabile lunga 60 caratteri)
- data di nascita (casella di tipo date)
- sesso (selezionabile tramite due bottoni radio)
- nazione di residenza (menu a tendina contenente tutte le nazioni)
- indirizzo (area di testo editabile di 6 righe per 80 colonne)
- email (casella di testo editabile lunga 30 caratteri)
- telefono (casella di testo editabile lunga 15 caratteri)
- prodotto richiesto (menu a tendina contenente i valori 1, 2, 3, 4, 5, 6)
- spedizione espresso (casella di checkbox)
- bottone di invio
- bottone di reset

e in cui vengano effettuati tramite funzioni JavaScript i seguenti controlli:

- (a1) verifica che il campo telefono contenga un numero. Questa verifica va fatta ogni volta che l'utente cambia il valore di questo campo;
- (a2) verifica che sia stata selezionata una nazione. Questa verifica va fatta all'atto dell'invio della form;
- (a3) verifica che, se si è selezionata la casella "spedizione espresso", allora il campo "prodotto richiesto" non vuoto e contiene un valore da 4 a 6. Questa verifica va fatta all'invio della form;
- (a4) verifica che, se la nazione di residenza non è l'Italia, non sia selezionata la casella "spedizione espresso", mentre se la nazione di residenza è l'Italia, o l'email o il telefono devono essere non vuoti. Questa verifica va fatta tutte le volte che cambia il valore del campo nazione di residenza;
- (a5) verifica che il campo indirizzo contenga almeno 15 caratteri. Questa verifica va fatta ogni volta che l'utente cambia il valore di questo campo.
- (b) Per ognuno dei controlli specificati al punto (a), dire se è realizzabile in HTML5 senza utilizzare codice JavaScript, e in caso positivo, spiegare come.

## Esercizio 2

Si assuma che il local storage contenga in un campo `URLvisitate` un array di URL (stringhe).

Scrivere un documento HTML che contiene una form che visualizza tale lista di URL e presenta un bottone **Cancella** e un bottone **Visita** a fianco di ogni URL: premendo il bottone **Cancella** relativo ad una URL, questa viene cancellata dall'array `URLvisitate` del local storage, e la form viene aggiornata eliminando tale URL; premendo il bottone **Visita**, viene visualizzata la URL corrispondente in un'altra finestra o scheda del browser.

**Esercizio 3** Data la seguente DTD:

```
<!DOCTYPE r [  
<!ELEMENT r ((b,d)|(c,(d,e)*,f),f)>  
<!ELEMENT b ANY>  
<!ELEMENT c (((b,c)|(d,(e,f)*)),(g,g))>  
<!ELEMENT d (d?,(a|e)*,f+,(b,c)*,g)>  
<!ELEMENT e (#PCDATA,d,e)*>  
<!ELEMENT f EMPTY>  
<!ELEMENT g #PCDATA>  
<!ATTLIST c attr CDATA #REQUIRED  
          attrc CDATA #REQUIRED>  
<!ATTLIST f attrf CDATA #REQUIRED>  
>
```

1) dire se la DTD è corretta ed in caso negativo evidenziare gli errori presenti e correggerli; 2) scrivere un documento XML che sia valido rispetto alla DTD (eventualmente corretta).

**Esercizio 4** Scrivere un foglio di stile XSL che, dato un documento XML, restituisce il documento tale che: 1) tutti gli elementi di input tranne gli elementi `a` e `y` vengono copiati in output, e vengono anche replicati come figli della loro copia in output. Inoltre il loro contenuto viene ricorsivamente trasformato; 2) ogni elemento `<a>` viene copiato in output creando un figlio `aaa` per tale elemento, e il suo contenuto viene ignorato; 3) ogni elemento `<b>` viene trasformato in un elemento

p, e viene creato per tale elemento un sottoelemento che ha per nome `olddb` e che ha un attributo che ha per nome il nome dell'elemento di input corrente e come valore la stringa `elemento b`. Inoltre, il contenuto dell'elemento di input viene ricorsivamente trasformato; 4) per ogni nodo di tipo testo che viene letto, viene generato in output un elemento `nodotesto` che ha un attributo `testo` che ha come valore il contenuto testuale dell'elemento corrente di input.

Ad esempio, se il documento XML di input è il seguente:

```
<z>
  testo 1
  <a><b>testo 2</b></a>
  <c>
    <d><w>testo 3</w></d>
  </c>
  <w>
    <t><b>testo 4<b/></t>
  </w>
</z>
```

il foglio di stile applicato al documento deve restituire il documento seguente:

```
<z>
  <z/>
  <nodotesto testo="testo 1"/>
  <a><aaa/></a>
  <c>
    <c/>
    <d>
      <d/>
      <w>
        <w/>
        <nodotesto testo="testo 3"/>
      </w>
    </d>
  </c>
  <w>
    <w/>
    <t>
      <p>
        <olddb b="elemento b"/>
        <nodotesto testo="testo 4"/>
      </p>
    </t>
  </w>
</z>
```

**Esercizio 5** Dato il seguente documento HTML:

```
<html>
  <body>
    <ul>
      <li>http://www.example.com/index.htm</li>
      <li>http://www.example.com/index2.htm</li>
      <li>http://www.uniroma1.it/esempio/e1.htm</li>
      <li>http://www.uniroma1.it/esempio/e2.htm</li>
      <li>http://www.xyz.abc/a.htm</li>
      <li>http://www.xyz.abc/ab.htm</li>
      <li>http://www.uniroma1.it/esempio/e3.htm</li>
      <li>http://www.uniroma1.it/esempio/e4.htm</li>
      <li>http://www.example.com/index3.htm</li>
      <li>http://www.example.com/index4.htm</li>
      <li>http://www.uniroma1.it/altro/a.htm</li>
      <li>http://www.uniroma1.it/altro/b.htm</li>
      <li>http://www.xyz.abc/abc.htm</li>
      <li>http://www.xyz.abc/abcd.htm</li>
      <li>http://www.uniroma1.it/altro/c.htm</li>
      <li>http://www.uniroma1.it/altro/d.htm</li>
    </ul>
    <div>visualizza qui i documenti che provengono da uniroma1.it</div><hr/>
    <div>visualizza qui i documenti che provengono da example.com</div><hr/>
    <div>visualizza qui i documenti che provengono da xyz.abc</div><br/><hr/>
  </body>
</html>
```

modificare il documento HTML in modo da poter attivare, cliccando sull'elemento `<li>` corrispondente, il caricamento asincrono delle 16 URL elencate nel documento: i documenti provenienti dal dominio `uniroma1.it` vanno visualizzati all'interno del primo elemento `div`, quelli provenienti dal dominio `example.com` vanno visualizzati all'interno del secondo elemento `div`, e quelli provenienti dal dominio `xyz.abc` vanno visualizzati all'interno del terzo elemento `div`.

**Esercizio 6** Scrivere un documento HTML che risolve il precedente esercizio 5 utilizzando JQuery.