

Linguaggi e tecnologie per il Web – appello del 15/1/2018

COGNOME:
NOME:
MATRICOLA:

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

Esercizio 1 (4 punti)

(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)
- regione di residenza (menu a tendina contenente le regioni italiane)
- 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 regione. 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 regione di residenza non è il Lazio, non sia selezionata la casella "spedizione espresso", mentre se la regione di residenza è il Lazio, o l'email o il telefono devono essere non vuoti. Questa verifica va fatta tutte le volte che cambia il valore del campo regione 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 (6 punti) Si assuma che il Local Storage contenga in un campo `UrlDaVisitare` un array di URL (stringhe) e in un campo `UrlVisitate` un altro array di URL (stringhe).

Scrivere un documento HTML che:

- visualizza in una tabella le URL contenute nel campo `UrlDaVisitare` del Local Storage. La visualizzazione deve essere sempre aggiornata con il contenuto corrente di tale array;
- permette all'utente di aggiungere una nuova URL nell'array `UrlDaVisitare` del Local Storage (tramite un campo input text e un bottone);
- ogni volta che l'utente clicca su una delle URL da Visitare, viene aggiornato il Local Storage, spostando la URL dall'array `UrlDaVisitare` all'array `UrlVisitate`.

Esercizio 3 (5 punti) Dato il seguente documento HTML:

```
<html>
  <body><ul>
    <li>http://www.uniroma1.it/index1.htm</li>
    <li>http://www.uniroma1.it/index2.htm</li>
    <li>http://www.uniroma1.it/index3.htm</li>
    <li>http://www.uniroma1.it/index4.htm</li>
    <li>http://www.uniroma1.it/index5.htm</li>
    <li>http://www.uniroma1.it/index6.htm</li>
    <li>http://www.uniroma1.it/index7.htm</li>
    <li>http://www.uniroma1.it/page1.htm</li>
    <li>http://www.uniroma1.it/page2.htm</li>
    <li>http://www.uniroma1.it/page3.htm</li>
    <li>http://www.uniroma1.it/page4.htm</li>
    <li>http://www.uniroma1.it/page5.htm</li>
    <li>http://www.uniroma1.it/page6.htm</li>
    <li>http://www.uniroma1.it/page7.htm</li>
  </ul></body>
</html>
```

modificare il documento HTML in modo da poter attivare, cliccando sull'elemento `` corrispondente, il caricamento asincrono delle URL elencate nel documento: ogni documento va visualizzato nell'elemento `` in cui compare la sua URL, senza però cancellare la URL stessa.

Esercizio 4 (4 punti) Scrivere un documento HTML che risolve il precedente esercizio 3 utilizzando JQuery.

Esercizio 5 (5 punti) Data la seguente DTD:

```
<!DOCTYPE z [  
<!ELEMENT z (a|(c,d)|e)*>  
<!ELEMENT a ((b,c,d)|(d,e,a*)|f+)>  
<!ELEMENT b ANY>  
<!ELEMENT c ((c*,((e,f)|g)*),b)>  
<!ELEMENT d (#PCDATA|d|z)*>  
<!ELEMENT e EMPTY>  
<!ELEMENT f (#PCDATA|a|(c,d)|e)*>  
<!ATTLIST a attr CDATA #REQUIRED  
          attra CDATA #IMPLIED>  
<!ATTLIST b attr 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) e che contenga tutti gli elementi dichiarati nella DTD.

Esercizio 6 (6 punti)

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 `z` e `p` 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 `<z>` viene copiato in output creando un figlio `trasf` per tale elemento, e il suo contenuto viene ignorato; 3) ogni elemento `<y>` viene trasformato in un elemento `nuovo`, e viene creato per tale elemento un sottoelemento che ha per nome `sottoelem` e che ha un attributo che ha per nome il nome dell'elemento di input corrente e come valore la stringa `elemY`. 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 `attr` che ha come valore il contenuto testuale dell'elemento corrente di input e come contenuto la stringa "TESTO TROVATO". Ad esempio, se il documento XML di input è il seguente:

```
<r>  
  testo 1  
  <z><y>testo 2</y></z>  
  <t>  
    <u><v>testo 3</v></u>  

```

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

```
<r>  
  <r/>  
  <nodotesto attr="testo 1">TESTO TROVATO<nodotesto/>  
  <z><trasf/></z>  
  <t>  
    <t/>  
    <u>  
      <u/>  
      <v>  
        <v/>  
        <nodotesto attr="testo 3">TESTO TROVATO<nodotesto/>  
      </v>  
    </u>  
  </t>  
  <v>  
    <v/>  
    <x>  
      <x/>  
      <nuovo>  
        <sottoelem y="elemY"/>  
        <nodotesto attr="testo 4">TESTO TROVATO<nodotesto/>  
      </nuovo>  
    </x>  
  </v>  
</r>
```