

Linguaggi e tecnologie per il Web – appello del 23/7/2018

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:

- cognome e nome (casella di testo editabile lunga 50 caratteri)
- matricola (casella di testo editabile lunga 12 caratteri)
- anno di corso (da scegliere da un menu con le opzioni “1”, “2”, “3”, “FC”);
- anno fuori corso (casella di testo editabile lunga 1 carattere). Questo campo deve essere attivo (editabile) solo se l’anno di corso selezionato al campo precedente è FC;
- canale di corso (da scegliere da un menu con le opzioni “1”, “2”, “Unico”);
- numero esami sostenuti (da scegliere da un menu a tendina contenente i numeri da 0 a 27)
- media voti esami (casella di testo editabile lunga 6 caratteri)
- dichiarazione di presa visione dell’informativa sulla privacy (casella di spunta)
- foto (permette all’utente di caricare un file con una sua foto)
- bottone di invio

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

- (a1) verifica che la matricola sia un numero e che contenga un minimo di 6 cifre e un massimo di 8 cifre. Questa verifica va fatta ogni volta che l’utente cambia la matricola;
- (a2) verifica che, se il numero di esami sostenuti selezionato è 0, allora la media dei voti degli esami è 0. Questa verifica va fatta ogni volta che cambia il campo numero di esami sostenuti oppure il campo media;
- (a3) verifica che sia stata spuntata la casella di presa visione dell’informativa sulla privacy. Questa verifica va fatta all’atto dell’invio della form;
- (a4) verifica che, se l’anno di corso selezionato è “3”, allora il valore del campo canale di corso è “Unico”, e che, se l’anno di corso selezionato è “1” o “2”, allora il valore del campo canale di corso è “1” se la lettera iniziale del cognome dello studente è compresa tra “A” e “K”, ed è “2” se la lettera iniziale del cognome dello studente è compresa tra “L” e “Z”. Questa verifica va fatta ogni volta che cambia il campo anno di corso oppure il campo canale di corso.

(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 **Carrello** un array di oggetti con 3 campi di tipo stringa, **codiceProdotto**, **numeroPezzi**, **prezzoUnitario**. Scrivere un documento HTML che:

- visualizza in due tabelle le informazioni contenute nell’array **Carrello** del Local Storage. La prima tabella deve presentare le informazioni sui prodotti che hanno un prezzo unitario maggiore di 20 Euro, mentre la seconda tabella deve presentare le informazioni sui restanti prodotti. Nelle due tabelle, il codice di ogni prodotto deve essere un link ipertestuale alla pagina di quel prodotto, che si trova alla URL <http://www.abc.com/<codice prodotto>>. Inoltre, in entrambe le tabelle ogni riga deve contenere un bottone **Cancella** che permette di cancellare quel prodotto dal carrello. La visualizzazione della tabella deve essere sempre aggiornata con il contenuto corrente dell’array **Carrello**;
- permette all’utente di aggiungere un nuovo prodotto nell’array **Carrello** del Local Storage, tramite una form con degli opportuni campi editabili e un bottone;
- deve infine essere presente un bottone **Calcola totale** il cui click fa comparire il costo totale dei prodotti presenti nel carrello, e un bottone **Costo medio unitario** il cui click fa comparire il costo medio unitario dei prodotti presenti nel carrello.

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

```
<html>
  <body>
    <table>
      <tr><td></td><td></td><td></td>
        <td></td></tr>
      <tr><td></td><td></td><td></td>
        <td></td></tr>
      <tr><td></td><td></td><td></td>
        <td></td></tr>
      <tr><td></td><td></td><td></td>
        <td></td></tr>
    </table>
  </body>
</html>
```

```

        <tr><td><img src=""></td><td><img src=""></td><td><img src=""></td><td><img src=""></td></tr>
</table>
</body>
</html>

```

modificare (il meno possibile) il documento HTML in modo che, ogni volta che viene cliccata una immagine nelle prime 4 righe della tabella, venga effettuato il caricamento asincrono di tale immagine e la sua visualizzazione nella casella della stessa colonna alla quinta riga della tabella.

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 r [
  <!ELEMENT r (((b,a)|(f,e)),((d,c)|(h,g)))+>
  <!ELEMENT a (h,g,g*,(e+|f+))>
  <!ELEMENT b (#PCDATA|b|c|d|r)*>
  <!ELEMENT c EMPTY>
  <!ELEMENT d ((d,g)|(f,e,c)|h)>
  <!ELEMENT e (e|b|(g,e))*>
  <!ELEMENT f ((f,g)|(a,b)|(h,e,c))>
  <!ELEMENT g (#PCDATA|a|b)+>
  <!ELEMENT h EMPTY>
  <!ATTLIST b attr CDATA #REQUIRED
            attrb CDATA #REQUIRED>
  <!ATTLIST h attr CDATA #REQUIRED
            attrb 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 HTML tale che: 1) ogni elemento <t> viene copiato in output come elemento `table`, e il valore dell'attributo `b` (se presente) viene copiato in output come valore dell'attributo `border` di `table`. Inoltre, il contenuto di `t` viene ricorsivamente trasformato; 2) ogni elemento <r> che è contenuto (a qualsiasi livello di annidamento) in un elemento `t` viene copiato in output come elemento `tr`, e il suo contenuto viene ricorsivamente trasformato; 3) ogni elemento <com> non viene copiato in output, e il suo contenuto viene ignorato; 4) ogni elemento <cel> che è contenuto (a qualsiasi livello di annidamento) in un elemento `r` viene copiato in output come elemento `td`, e il suo contenuto viene ricorsivamente trasformato; 5) ogni altro elemento di input non viene copiato in output, e il suo contenuto viene ricorsivamente trasformato; 6) ogni nodo di tipo testo che è figlio di un elemento `td` viene copiato in output, mentre tutti gli altri nodi testuali vanno ignorati.

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

```

<a>
  <t b="2">
    testo 0
    <r>
      <com>testo 1</com>
      <cel>testo 2</cel>
      <cel>testo 3</cel>
      testo 4
    </r>
    <cel>testo 5</cel>
    <r>
      testo 6
      <cel>testo 7</cel>
      <cel>testo 8</cel>
    </r>
  </t>
</a>

```

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

```

<html>
  <body>
    <table border="2">
      <tr>
        <td>testo 2</td>
        <td>testo 3</td>
      </tr>
      <tr>
        <td>testo 7</td>
        <td>testo 8</td>
      </tr>
    </table>
  </body>
</html>

```