

# Linguaggi e tecnologie per il Web – appello dell'8/2/2018

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 (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 Lazio, Umbria e Toscana)
- provincia di residenza (menu a tendina contenente le province delle regioni Lazio, Umbria e Toscana)
- 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 di almeno 8 cifre e al più 15 cifre. 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 sia stata selezionata una provincia appartenente alla regione selezionata. Questa verifica va fatta tutte le volte che cambia il valore del campo regione di residenza oppure il valore del campo provincia di residenza (se non è ancora stata selezionata alcuna regione o non è ancora stata selezionata alcuna provincia, allora il controllo deve sempre dare esito positivo);
- (a4) verifica che, se la provincia di residenza non è Roma, Firenze o Perugia, non sia selezionata la casella "spedizione espresso", mentre se la provincia di residenza è Roma, Firenze o Perugia, o l'email o il telefono devono essere non vuoti. Questa verifica va fatta tutte le volte che cambia il valore del campo provincia di residenza.
- (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 `Clients` un array di record, ognuno dei quali contiene quattro campi: `nome` (String), `dataDiNascita` (Date), `email` (String) e `URL` (String).

Scrivere un documento HTML che:

- chiede all'utente (tramite opportuni campi di input) di selezionare un intervallo di età (valore minimo selezionabile: 18 anni, valore massimo selezionabile: 100 anni), e visualizza in una tabella a quattro colonne le informazioni sui clienti contenuti nel Local Storage che hanno età compresa nell'intervallo specificato dall'utente. La visualizzazione deve poi essere sempre aggiornata con il contenuto corrente del Local Storage;
- permette all'utente di aggiungere un nuovo cliente nell'array `Clients` del Local Storage (tramite degli opportuni campi di input e un bottone). Non possono essere presenti due clienti con lo stesso indirizzo di email;
- ogni volta che l'utente clicca su una delle URL presenti nella tabella precedente, viene aggiornato il Local Storage, eliminando il cliente dall'array `Clients`.

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

```
<html>
  <body>
    <div>
      <a href="http://www.abc/index1.htm">link</a> <a href="http://www.xyz/index2.htm">link</a>
      <a href="http://www.abc/index3.htm">link</a> <a href="http://www.xyz/index4.htm">link</a>
      <a href="http://www.abc/index5.htm">link</a> <a href="http://www.xyz/index6.htm">link</a>
      <a href="http://www.abc/index7.htm">link</a> <a href="http://www.xyz/index8.htm">link</a>
    </div>
    <div>
      <a href="http://www.xyz/page1.htm">link</a> <a href="http://www.abc/page2.htm">link</a>
      <a href="http://www.xyz/page3.htm">link</a> <a href="http://www.abc/page4.htm">link</a>
      <a href="http://www.xyz/page5.htm">link</a> <a href="http://www.abc/page6.htm">link</a>
    </div>
  </body>
</html>
```

```

    <a href="http://www.xyz/page7.htm">link</a> <a href="http://www.abc/page8.htm">link</a>
  </div>
</body>
</html>

```

modificare (il meno possibile) il documento HTML in modo da poter attivare, cliccando sull'elemento <a> corrispondente, il caricamento asincrono delle URL elencate nel documento: ogni documento va visualizzato nella posizione in cui compare l'elemento <a> stesso, senza cancellare il link stesso.

**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 a [
<!ELEMENT a (a|(b,c,d)|(e|f)+)>
<!ELEMENT b ((b,g,d)|(d,e,f*)|e+)>
<!ELEMENT d (#PCDATA|d|z)*>
<!ELEMENT e EMPTY>
<!ELEMENT f (#PCDATA|a|(b,d)|e)*>
<!ELEMENT g ((c*,((e,f)|g)*),b)>
<!ATTLIST a attr CDATA #REQUIRED
          attra CDATA #IMPLIED>
<!ATTLIST b attr CDATA #REQUIRED>
<!ATTLIST e attre 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) l'elemento radice viene trasformato nell'elemento <html> di output, viene creato un sottoelemento <head> vuoto e un sottoelemento <body>, il contenuto dell'elemento viene ricorsivamente trasformato e il risultato della trasformazione viene posizionato dentro <body>; 2) ogni elemento <c> di input che è figlio dell'elemento radice viene trasformato in un elemento <h1>, il suo contenuto testuale viene copiato, mentre il restante contenuto viene ignorato; 3) ogni elemento <d> di input che è figlio dell'elemento radice viene trasformato in un elemento <div> con attributo align uguale a center, il suo contenuto testuale viene ignorato, mentre il restante contenuto viene ricorsivamente trasformato; 4) per ogni elemento <e> di input che è figlio di un figlio dell'elemento radice, viene creato un elemento <p> e un sottoelemento <span> che ha l'attributo class e che ha come valore il nome dell'elemento corrente di input. Inoltre, il contenuto dell'elemento di input viene ricorsivamente trasformato; 5) ogni elemento <a> di input che viene letto viene trasformato in un elemento <div>, e il suo contenuto viene ricorsivamente trasformato; 6) ogni elemento <b> di input che viene letto viene trasformato in un elemento <span>, e il suo contenuto viene ricorsivamente trasformato; 7) ogni altro elemento non viene copiato in output, e il suo contenuto viene ignorato; 8) ogni nodo di tipo testo che viene letto viene copiato nella posizione corrente (tranne le parti testuali di cui al precedente punto 3).

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

```

<r>
  testo 1
  <c>testo 2<a>testo 2bis</a></c>
  <d>testo 3<b>testo 3bis</b></d>
  <e>testo 4</e>
  <f><e>testo 5</e></f>
  <c>testo 6</c>
  <a>
    testo 7
    <e>testo 8</e>
  </a>
</r>

```

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

```

<html>
  <head></head>
  <body>
    testo 1
    <h1>testo2</h1>
    <div align="center"><span>testo 3bis</span></div>
    <c>testo 6</c>
    <div>
      testo 7
      <p><span class="e">testo 8</span></p>
    </div>
  </body>
</html>

```