

Linguaggi e tecnologie per il Web – appello del 5/7/2022

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

- cognome e nome (casella di testo editabile lunga 40 caratteri)
- matricola (casella di testo editabile lunga 5 caratteri)
- codice fiscale (casella di testo editabile lunga 16 caratteri)
- corso di laurea (da scegliere da un menu che riporta alcuni corsi di laurea, incluso IngInfAut)
- tipo corso (selezionabile tramite un menu a due opzioni, T e M)
- email (casella di testo editabile lunga 30 caratteri)
- anno di corso (casella di testo editabile lunga 2 caratteri)
- bottone di invio e bottone di reset

in modo tale che vengano effettuati i seguenti controlli lato client:

- (a1) verifica che il campo matricola sia un numero compreso tra 10000 e 99999. Questa verifica va fatta quando l'utente invia la form;
- (a2) verifica che il codice fiscale sia effettivamente un codice fiscale (6 lettere maiuscole + 2 cifre + 1 lettera maiuscola + 2 cifre + 1 lettera maiuscola + 3 cifre + 1 lettera maiuscola). Questa verifica va fatta ogni volta che l'utente modifica questo campo;
- (a3) verifica che, se il tipo corso di laurea selezionato è T, allora l'anno di corso sia un numero compreso tra 1 e 3 oppure sia la stringa "FC", e se il tipo corso di laurea selezionato è M, allora l'anno di corso sia un numero compreso tra 1 e 2 oppure sia la stringa "FC". Questa verifica va fatta ogni volta che l'utente cambia o il valore del campo anno di corso o il valore del campo tipo corso.
- (a4) verifica che, se il corso di laurea selezionato è IngInfAut, allora o il codice fiscale o la matricola o il cognome e nome sono non vuoti. Questa verifica va fatta quando l'utente invia la form.

Esercizio 2 (6 punti) Scrivere una pagina web che permette di memorizzare nel Local Storage informazioni relative a film visualizzati dall'utente. Le informazioni da memorizzare sono: il titolo del film visualizzato, l'anno di produzione del film, il nome del regista del film, il genere del film, l'elenco degli attori e attrici del film. La pagina web deve:

1. permettere all'utente di aggiungere nel Local Storage le informazioni relative ad un nuovo film;
2. permettere all'utente di selezionare un genere, e quindi visualizzare nella pagina stessa le informazioni di tutti i film di quel genere;
3. permettere all'utente di visualizzare nella pagina stessa l'elenco di tutti gli attori e attrici che hanno partecipato ad almeno un film;
4. permettere all'utente di inserire il nome di un attore/attrice, e quindi visualizzare nella pagina stessa le informazioni di tutti i film a cui tale attore/attrice ha partecipato;
5. permettere all'utente di inserire il nome di un regista, e quindi cancellare dal Local Storage le informazioni di tutti i film di tale regista.

Esercizio 3 (5 punti) Data una pagina HTML contenente una serie di elementi `<a>` e tre elementi `<p>`, modificare (il meno possibile) la pagina *senza introdurre attributi negli elementi `<a>`* in modo che, cliccando su ogni elemento `<a>`, venga effettuato il caricamento asincrono della URL corrispondente al valore dell'attributo `href` di tale elemento `<a>`. Ogni risorsa la cui URL ha per suffisso `.htm` va visualizzata nel primo elemento `<p>` della pagina, ogni risorsa la cui URL ha per suffisso `.jpg` va visualizzata nel secondo elemento `<p>` della pagina, e ogni altra risorsa va visualizzata nel terzo elemento `<p>` della pagina.

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

Esercizio 5 (5 punti)

(a) Dato il seguente oggetto JSON:

```
{ "n1": [100, "ciao", "ciao", [ "ciao", 9 ], 1 ], "n2": { "p1": 12, "p2": ["x","y"], "p3": "hello" }, "n3": [4] }
```

rappresentare tale oggetto in XML.

(b) Dato il seguente documento XML:

```

<r>
  <x>
    <studente>
      <cognomeNome>Mario Rossi</cognomeNome>
      <annoNascita>2001</annoNascita>
      <matricola>1234</matricola>
    </studente>
  </x>
  <y>
    <z>
      <studente>
        <cognomeNome>Anna Bianchi</cognomeNome>
        <matricola>5678</matricola>
        <indirizzo>Via Verdi 18</indirizzo>
      </studente>
    </z>
  </y>
</r>

```

rappresentare tale documento come oggetto JSON.

Esercizio 6 (6 punti) Scrivere un foglio di stile XSL che, dato un documento XML, restituisce il documento HTML tale che: 1) ogni elemento **impiegato** (a qualsiasi livello di annidamento) viene copiato in output come elemento **p**, e viene generato un sottoelemento **div** per contenere le informazioni sull'impiegato (presenti nei sottoelementi di **impiegato**). Inoltre il contenuto di **impiegato** viene ricorsivamente trasformato; 2) ogni elemento **cognomeNome** viene copiato in output come elemento **span** con valore dell'attributo **name** pari a **cn**, e il suo contenuto viene ricorsivamente trasformato; 3) ogni elemento **codFiscale** viene copiato in output come elemento **span** con valore dell'attributo **name** pari a **cf**, e il suo contenuto viene ricorsivamente trasformato; 4) ogni elemento **citta** viene copiato in output come elemento **div** con valore dell'attributo **name** pari a **ci**, e il suo contenuto viene ignorato; 5) ogni altro elemento di input non viene copiato in output, e il suo contenuto viene ricorsivamente trasformato; 6) solo i nodi di tipo testo contenuti negli elementi **cognomeNome**, **codFiscale** e **citta** vengono copiati in output: tutti gli altri nodi testuali non vengono copiati in output. Ad esempio, se il documento XML di input è il seguente:

```

<r>
  <impiegato>
    <cognomeNome>Carlo Verdi</cognomeNome>
    <codFiscale>ABCDEF89A26Z999Z</codFiscale>
  </impiegato>
  <x>
    <impiegato>
      <cognomeNome>Anna Bianchi</cognomeNome>
      <citta>Roma</citta>
      <codFiscale>GHILM098C12Z999Z</codFiscale>
      <indirizzo>Via Verdi 18</indirizzo>
    </impiegato>
  </x>
  <y>
    <z>
      <impiegato>
        <cognomeNome>Mario Rossi</cognomeNome>
        <citta>Napoli</citta>
        <indirizzo>via Roma 123</indirizzo>
      </impiegato>
    </z>
  </y>
</r>

```

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

```

<html>
  <body>
    <p>
      <div><span name="cn">Carlo Verdi</span><span name="cf">ABCDEF89A26Z999Z</span></div>
    </p>
    <p>
      <div><span name="cn">Anna Bianchi</span><div name="ci">Roma</div><span name="cf">GHILM098C12Z999Z</span></div>
    </p>
    <p>
      <div><span name="cn">Mario Rossi</span><div name="ci">Napoli</div></div>
    </p>
  </body>
</html>

```