Tecnologie Web (6 CFU)

C.d.L. in Informatica

Compito del 20 giugno 2011

Nome:

Cognome:

Matricola:

Team:

□ Non è la prima volta che provo questo esame

Ricapitolo: fare qui sotto una croce sul numero di ciascun esercizio risolto. Se lo si consegna su un foglio protocollo, scrivere in maniera chiara il numero dell'esercizio presso la sua soluzione.

Esercizio		**Punti**	**Voto**
1	Domande di base	12	
2	HTML	6	
3	XSLT	6	
4	Javascript	6	
5	Javascript	4	
6	Teoria	4	
Totale		**38**	

Domande di base (12 punti)

Rispondere correttamente ad almeno tre delle seguenti domande:

1) Spiegare perché la seguente affermazione è falsa: "SGML e XML sono linguaggi di markup."

2) Elencare il nome di almeno tre elementi HTML relativi all'uso di form.

3) Quale tra le seguente regole CSS non è corretta?

- `p p {font-size:10pt;}`
- `p.p {font-size:10pt;}`
- `p & p {font-size:10pt;}`
- `p {font-size:10pt;}`
- `pp {font-size:10pt;}`
- `#p {font-size:10pt;}`

4) Scrivere un'altra espressione XPath che, applicata al seguente frammento XML, produca lo stesso risultato dell'espressione `/book/chapter[@num='1']/title`.

```
<book>
        <chapter num="1">
                <title>Introduzione</title>
                <para>Testo ... testo ... testo.</para>
        </chapter>
        <chapter num="2">
                <title>La prima fase</title>
                <para>Altro testo</para>
        </chapter>
</book>
```

HTML (6 punti)

Prima parte

Scrivere il codice XHTML 1.0 Strict (e CSS) di un documento che rappresenti il contenuto di questa immagine:

Nello scrivere il codice si tengano in considerazione questi vincoli:

- nessun elemento deve contenere l'attributo `class` (di conseguenza non si possono usare usare selettori classe nel codice CSS);
- nessun elemento deve contenere l'attributo `style`;
- gli URI di destinazione dei link presenti nella pagina devono essere inventati e devono essere URI validi (e sensati).

Nota: il DocType di XHTML 1.0 Strict è `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">` e il namespace è `http://www.w3.org/1999/xhtml`.

Si consiglia di scrivere tutto il codice CSS in un "file" separato, non in elementi `<style>`.

Seconda parte

Alcune parti della pagina precedentemente creata potrebbero essere marcate usando alcuni dei nuovi elementi semantici introdotti in HTML 5. Quali parti? Attraverso quali elementi?

XSLT (6 punti)

Scrivere un foglio di stile XSLT che trasforma il documento A nel documento B:

Documento A	Documento B

<grigliata>
  <spiedino tipo="misto">
  <carne>maiale</carne>
  <carne>pollo</carne>
  <verdura>peperone</verdura>
  <carne>manzo</carne>
  <verdura>melanzana</verdura>
  </spiedino>
  <spiedino tipo="carne">
  <carne>wurstel</carne>
  <carne>maiale</carne>
  <carne>pollo</carne>
  </spiedino>
  <spiedino tipo="misto">
  <verdura>peperone</verdura>
  <carne>salsiccia</carne>
  <verdura>zucchina</verdura>
  <carne>wurstel</carne>
  </spiedino>
  <spiedino tipo="verdura">
  <verdura>zucchina</verdura>
  <verdura>peperone</verdura>
  <verdura>melanzana</verdura>
  </spiedino>
  <spiedino tipo="carne">
  <carne>salsiccia</carne>
  <carne>maiale</carne>
  <carne>pollo</carne>
  </spiedino>
</grigliata>
	<grill skewers='5'
	  ingredients-in-italian="maiale, manzo,
	  melanzana, peperone,
	  pollo, salsiccia,
	  wurstel, zucchina">
	  <meat>
	  <skewer>
	  <wurst/>
	  <pork/>
	  <chicken/>
	  </skewer>
	  <skewer>
	  <sausage/>
	  <pork/>
	  <chicken/>
	  </skewer>
	  </meat>
	  <vegetable>
	  <skewer>
	  <courgette/>
	  <pepper/>
	  <eggplant/>
	  </skewer>
	  </vegetable>
	  <mixed>
	  <skewer>
	  <pork/>
	  <chicken/>
	  <pepper/>
	  <beef/>
	  <eggplant/>
	  </skewer>
	  <skewer>
	  <pepper/>
	  <sausage/>
	  <courgette/>
	  <wurst/>
	  </skewer>
	  </mixed>
	  </grill>

NOTA: la traduzione dei termini è riportata sotto:

```
gliglia   = grill
spiedino  = skewer
carne     = meat
verdura   = vegetable
maiale    = pork
manzo     = beef
melanzana = eggplant
peperone  = sausage
pollo     = chicken
salsiccia = sausage
wurstel   = wurst
zucchina  = courgette
```

Javascript (6 punti)

Dato in Javascript un array (privo di ordini particolari) come il seguente:

```
var campionato2011 = [
        { squadra: 'Milan',   punteggio: 82},
        { squadra: 'Bari',    punteggio: 24}
        { squadra: 'Udinese', punteggio: 66},
        { squadra: 'Lazio',   punteggio: 66},
        { squadra: 'Inter',   punteggio: 76},
        { squadra: 'Napoli',  punteggio: 70},
        ...
        ];
```

si scriva una funzione che genera una tabella HTML (come stringa o come DOM) tale per cui:

- Ogni squadra è posta su una riga diversa
- Il nome della squadra è posto nella prima colonna
- Il punteggio della squadra è posto nella seconda colonna
- La tabella è ordinata in ordine alfabetico per squadra
- Ad ogni riga è associato uno stile CSS 'primoQuartile', 'secondoQuartile, 'terzoQuartile', 'quartoQuartile' a seconda che la squadra si sia posizionata nel primo 25% della classifica, oppure rispettivamente nel secondo, nel terzo o nel quarto.

Si noti che le squadre in questione sono in effetti 20, ma non si assuma che l'array debba per forza avere esattamente venti posizioni (né un numero pari di posti).

Funzioni utili a tal proposito sono:

- `a.sort()` ordina un array di elementi atomici (non di array associativi!!!).
- `a.join(sep)` restituisce una stringa in cui sono presenti tutti i valori dell'array a separati dalla stringa `sep`
- `s1.indexOf(s2)` restituisce la posizione in cui si trova la stringa `s2` all'interno della stringa `s1`, o -1 se non compare.
- `Math.floor(n)` restituisce l'intero inferiore ad un numero float
- `Math.ceiling(n)` restituisce l'intero superiore ad un numero float.

Bonus:

- (2 punti) Si usi una versione della funzione a.sort(f) a cui va passato un parametro f, che è una funzione a cui vengono passati a due a due gli elementi dell'array e restituisce un numero minore di zero se il primo è inferiore al secondo, un numero maggiore di zero se il primo è superiore al secondo, e 0 se i due numeri sono uguali.
- (2 punti) La funzione di associazione dei quartili funziona correttamente anche se più squadre hanno lo stesso punteggio finale. Si spieghi l'algoritmo prescelto.

Javascript (4 punti)

Dato in Javascript un array (privo di ordini particolari) come il seguente:

si scriva una funzione che genera una tabella HTML (come stringa o come DOM) tale per cui:

- ogni squadra è posta su una riga diversa,
- ol nome della squadra è posto nella prima colonna,
- ol punteggio della squadra è posto nella seconda colonna,
- la tabella è ordinata in ordine alfabetico per squadra,
- ad ogni riga è associato uno stile CSS 'primoQuartile', 'secondoQuartile, 'terzoQuartile', 'quartoQuartile' a seconda che la squadra si sia posizionata nel primo 25% della classifica, oppure rispettivamente nel secondo, nel terzo o nel quarto.

Si noti che le squadre in questione sono in effetti 20, ma non si assuma che l'array debba per forza avere esattamente venti posizioni (né un numero pari di posti).

Funzioni utili a tal proposito sono:

- `a.sort()` ordina un array `a` di elementi atomici (non di array associativi!).
- `a.join(sep)` restituisce una stringa in cui sono presenti tutti i valori dell'array `a` separati dalla stringa `sep`.
- `s1.indexOf(s2)` restituisce la posizione in cui si trova la stringa `s2` all'interno della stringa `s1`, o -1 se non compare.
- `Math.floor(n)` restituisce l'intero inferiore ad un numero float `n`.
- `Math.ceiling(n)` restituisce l'intero superiore ad un numero float `n`.

Bonus:

(2 punti) Si usi una versione della funzione `a.sortBy(f)` a cui va passato un parametro `f`, che è una funzione a cui vengono passati a due a due gli elementi dell'array e restituisce un numero `< 0` se il primo è inferiore al secondo, un numero `> 0` se il primo è superiore al secondo, e `0` se i due numeri sono uguali.

(2 punti) La funzione di associazione dei quartili funziona correttamente anche se più squadre hanno lo stesso punteggio finale. Si spieghi l'algoritmo prescelto.

Teoria (4 punti)

Spiegare cosa è un 'tesauro' e descrivere le relazioni principali che stabilisce (gerarchica, sinonimica, associativa).

Spiegare inoltre perché SKOS è rilevante in questo contesto.