

# Tecnologie Web (6 CFU)

## C.d.L. in Informatica

### Compito del 18 febbraio 2012

**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.

<b>Esercizio</b>		<b>Punti</b>	<b>Voto</b>
1	Domande di base	12	
2	HTML	4	
3	Javascript	8	
4	Semantic Web	6	
5	Teoria	4	
<b>Totale</b>		<b>34</b>	

## **Domande di base (12 punti)**

Rispondere correttamente ad almeno tre delle seguenti domande:

**A.** Si consideri il seguente frammento XML:

```
<sez>
  <p>Un primo paragrafo</p>
  <p>Un secondo paragrafo</p>
</sez>
```

Descrivere cosa restituisce il seguente XPath: //p

**B.** Si consideri il frammento XML della precedente domanda. Cosa fa il seguente blocco CSS:

```
p { text-align:justify ; font-size:12pt; background-color: yellow; }
```

**C.** Considerare la seguente frase: *HTTP è un linguaggio di markup*. Dire se è vero o falso, motivandone la risposta.

**D.** Indicare quali elementi tra i seguenti non appartengono al vocabolario di HTML:

- input
- list
- ul
- table
- heading

## HTML (4 punti)

Scrivere il codice HTML 5 (e CSS) di un documento che rappresenti il contenuto di questa immagine:

# Menù del giorno

Seleziona le pietanze del giorno preferite (primo, secondo con contorno e dessert) ed invia il tuo ordine.

Primo:	Tagliatelle al ragu
Secondo con contorno:	Scaloppina ai funghi
Dessert:	Fiordilatte
Invia il tuo ordine	

Nello scrivere il codice si tengano in considerazione questi vincoli:

- la funzione Javascript da chiamare una volta cliccato sul pulsante "Invia il tuo ordine" è `spedisciMenu` (vedi esercizi di Javascript, punto 3);
- le etichette dei vari campi del form (ovvero "Primo", "Secondo con contorno" e "Dessert") devono essere definite con l'apposito elemento di markup;
- i ":" dopo il nome delle etichette non sono presenti nel corpo della pagina ma sono aggiunti via CSS;
- il bordo delle etichette è nero (dimensione singola), mentre quello del form (dimensione doppia) e dell'ultimo paragrafo del form (dimensione singola) è blu;
- il colore di sfondo delle etichette e dell'ultimo paragrafo del form è giallo;
- i vari campi del form devono contenere i seguenti valori:

### Primo

Tagliatelle al ragù, Pasta al pomodoro

### Secondo con contorno

Scaloppina ai funghi, Stracchino e grissini

### Dessert

Fiordilatte, Kiwi

Tutti i dettagli non specificati (per esempio nomi file e variabili) possono essere liberamente scelti.

Inoltre,

- non è possibile fare uso di tabelle;
- non si possono usare né selettori classe né i selettori di id nel codice CSS;
- nessun elemento deve contenere l'attributo `style`;

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

## JavaScript (8 punti)

Nel sistema informativo di una mensa scolastica si vuole sperimentare un sistema che permetta ai singoli scolari di scegliere il proprio menù del pranzo. Alla mattina lo scolaro o un genitore accede al sistema informativo, inserisce le proprie credenziali e si trova di fronte ad un'interfaccia che gli permette di scegliere tra i vari piatti offerti quel giorno. Ogni menù è composto da primo, secondo con contorno e dessert (frutta o dolce). Il sistema spedisce il menù del giorno con una struttura JSON come segue:

```
{
  giorno: '2012-02-20',
  scuola: 'Elementare Montessori - Bologna',
  consegna: '12:30',
  menu: {
    primi: [{
      cod: 'trag',
      nome: 'Tagliatelle al ragu',
      quantità: '80 g.',
      ingredienti: ['pasta all'uovo', 'manzo', 'salsa pomodoro', 'spezie']
    }, {
      cod: 'ppom',
      nome: 'Pasta al pomodoro',
      quantità: '80 g.',
      ingredienti: ['pasta secca', 'salsa pomodoro', 'spezie']
    }, ... ],
    secondi: [{
      cod: 'sfun',
      nome: 'Scaloppina ai funghi',
      quantità: '50 g.',
      ingredienti: ['manzo', 'funghi secchi', 'besciamella']
    }, {
      cod: 'stra',
      nome: 'Stracchino e grissini',
      quantità: '100 g.',
      ingredienti: ['formaggio fresco', 'grissini']
    }, ... ],
    dessert: [{
      cod: 'fdlt',
      nome: 'fiordilatte',
      quantità: '100 g.',
      ingredienti: ['latte', 'zucchero', 'aromi naturali']
    }, {
      cod: 'kiwi',
      nome: 'Kiwi',
      quantità: '2',
      ingredienti: ['kiwi']
    }, ... ]
  }
}
```

Data la molteplicità di etnie presenti nelle scuole pubbliche italiane e l'incremento di allergie e intolleranze alimentari nei bambini, alcuni cibi debbono essere di volta in volta esclusi dalle opzioni selezionabili perché non adatti alla cultura, alla religione o alla salute del singolo studente. Poiché non si può pensare che sia il bambino stesso ad evitare di scegliere i cibi inadatti (potrebbe ignorare i dettami della propria dieta o volerli sfidare), è necessario che il sistema escluda automaticamente i cibi inadatti.

In caso di successo dell'operazione di login, dunque, il sistema fornisce un oggetto JSON (non modificabile, stabilito in fase di registrazione dello studente) come il seguente:

```
{
  nome: 'Andrea',
  cognome: 'Rossi',
  classe: '3',
}
```

```
    sezione: 'A',  
    restrizioni: ['vegetariano', 'celiaco']  
}
```

Si assuma che esista già fornita una funzione Javascript puòMangiare(R,I), che restituisce Vero o Falso a seconda che una persona con la restrizione R possa consumare o meno l'ingrediente I, e che contiene ogni compatibilità possibile per ogni tipo di restrizione ed ogni ingrediente. Scrivere:

1. La funzione restringiMenu(AR,M), che restituisce in formato JSON il menù risultante dall'eliminazione dei cibi che contengono ingredienti incompatibili con le restrizioni, dove AR è un array di restrizioni e M è un menu in JSON (come il blocco 'menu' nel JSON qui sopra).
2. La funzione mostraScelte(M), che va a sostituire gli elementi più appropriati del form dell'esercizio precedente e visualizzano i dati del menu M come segue: per ogni categoria (primo, secondo, dessert): se la categoria contiene un solo cibo, visualizza il testo corrispondente; se la categoria contiene due o tre cibi, visualizza altrettanti radio button: se contiene quattro o più voci, visualizza un menu a tendina con un elemento per ogni cibo. Di ogni cibo viene mostrato il nome, la quantità tra parentesi e la lista di ingredienti nel testo alternativo.
3. La funzione spedisciMenu, che si collega con collegamento Ajax e metodo POST all'indirizzo <http://www.menuscolastico.it/mandamenu.php> spedendo i valori di tutti i campi del form, e mostrando in caso di successo il messaggio "Indicazioni del menù ricevute correttamente", e in caso di insuccesso il messaggio "C'è stato un errore nella spedizione delle indicazioni di menù. Per cortesia riprova"

## **Semantic Web (6 punti)**

Considerare le due seguenti frasi in linguaggio naturale:

*Alice è moglie di Bob. Bob e Alice sono delle Persone.*

Rispondere ai seguenti quesiti:

1. In un linguaggio a scelta tra Turtle e RDF/XML, descrivere con tre statement RDF le precedenti frasi, considerando che la risorsa che definisce l'insieme delle persone è *foaf:Person* mentre la relazione "è moglie di" è definita dalla proprietà *ov:wifeOf*. Non è necessario specificare gli URL dei prefissi.
2. In un linguaggio a scelta tra Turtle e RDF/XML, applicare il meccanismo di reificazione per descrivere in RDF la seguente frase: *Carl crede che Alice sia moglie di Bob*. La relazione "crede" è definita dalla proprietà *ov:believes*.
3. Scrivere una query SPARQL che restituisca tutte le persone che sono o mariti o mogli.

## **Teoria (4 punti)**

Descrivere che cos'è XSLT ed indicare le differenze che ci sono tra le due filosofie di riscrittura utilizzabili in XSLT, ovvero lo *stile iterativo* e lo *stile ricorsivo*, supportando la discussione con esempi.