

# Tecnologie Web (6 CFU)

## C.d.L. in Informatica

### Compito del 14 luglio 2010

**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	4	
4	Codifica	4	
5	Teoria	6	
<b>Totale</b>		<b>32</b>	

## Domande di base (12 punti)

1) Scrivere un frammento XML validato dal DTD A, ma non dal DTD B.

### DTD A:

```
<!ENTITY % content "(#PCDATA | b)*">
<!ELEMENT b %content;>

<!ELEMENT paper (title, subtitle?)>
<!ELEMENT subtitle %content;>
<!ELEMENT title %content;>
```

### DTD B:

```
<!ENTITY % content "(#PCDATA)">

<!ELEMENT paper (title, subtitle)>
<!ELEMENT subtitle %content;>
<!ELEMENT title %content;>
```

2) Scrivere le regole CSS per formattare in grassetto tutti i paragrafi P di un documento HTML, tranne quelli con @class='content' la cui formattazione è normale.

3) Il seguente documento HTML è memorizzato su un server web. Quante richieste HTTP deve eseguire un browser per caricarlo?

```
<html>
  <head>
    <link rel='stylesheet' href="layout.css" type="text/css">
    <link rel='stylesheet' href="skin.css" type="text/css">
  </head>
  <body>
    
  </body>
</html>
```

## HTML (6 punti)

Scrivere il codice XHTML 1.0 Strict (e CSS) di un documento con queste caratteristiche:

- titolo "Musica al parco" centrato orizzontalmente nella pagina, in grassetto e maiscoletto;
- una lista di tre link esterni (URI assoluti, contenuto e destinazioni non rilevanti). Il testo dei link deve essere non sottolineato e di colore rgb(200,15,0);
- un form per l'invio di messaggi.

Il form per le ordinazioni deve avere le seguenti caratteristiche:

- spedisce i dati in POST all'URI `http://example.org/tw/musica`
- chiede all'utente i seguenti campi (etichetta, nome campo, tipo di dato):
  - Nome o soprannome, nome, testo libero
  - Messaggio per il DJ, `messaggio_dj`, testo libero
  - Canzone preferita, canzone, scelta opzionale di al massimo una di queste possibilità (etichetta, valore) da questi due gruppi diversi:
    - in italiano
      - Notti magiche, `notti_magiche`
      - Gelato al cioccolato, `gelato`
      - La notte vola, `vola`
    - in altre lingue
      - Waka Waka, `waka`
      - Rock Me Amadeus, `rock`
      - Vamos a la playa, `playa`
  - Spedisci messaggio, `spedisci`, pulsante d'invio modulo
- ha tutte le etichette dei campi di colore giallo e in grassetto su sfondo nero
- ha un collegamento all'indirizzo email `info@example.org` di colore rosso.

Note

- 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`.
- Il codice XHTML non deve contenere attributi `style`.
- Scrivere tutto il codice CSS in un "file" separato.

## XSLT (4 punti)

Scrivere un foglio di stile XSLT **progettato secondo lo stile PULL (iterativo)** che trasforma il seguente frammento XML nella visualizzazione HTML mostrata di seguito.

Si noti inoltre che:

- il foglio XSLT deve produrre codice HTML che **NON** usa tabelle.
- in output gli attori sono in ordine alfabetico per cognome
- le misure esatte di margini e padding non sono rilevanti, purché siano consistenti

```
<filmografia>
  <film code="f1">
    <titolo>The A-Team</titolo>
    <attori>
      <attore nome="Liam" cognome="Neeson"
        personaggio="Col. John 'Hannibal' Smith" type="main-character"/>
      <attore nome="Bradley" cognome="Cooper"
        personaggio="Lt. Templeton 'Faceman' Peck" type="main-character"/>
      <attore nome="Jessica" cognome="Biel"
        personaggio="Charisa Sosa" type="main-character"/>
      <attore nome="Quinton" cognome="Jackson"
        personaggio="Cpl. Bosco 'B.A.' Baracus" type="main-character"/>
    </attori>
  </film>
  <film code="f2">
    <titolo>Toy Story 3</titolo>
    <attori>
      <attore nome="Tom" cognome="Hanks"
        personaggio="Woody" type="main-character"/>
      <attore nome="Tim" cognome="Allen"
        personaggio="Buzz Lightyear"/>
      <attore nome="Joan" cognome="Cusack"
        personaggio="Jessie"/>
    </attori>
  </film>
  <film code="f3">
    <titolo>The Last Airbender</titolo>
    <attori>
      <attore nome="Noah" cognome="Ringer"
        personaggio="Aang" type="main-character"/>
      <attore nome="Dev" cognome="Patel"
        personaggio="Prince Zuko" type="main-character"/>
      <attore nome="Nicola" cognome="Peltz" personaggio="Katara"/>
    </attori>
  </film>
</filmografia>
```

## Actors:

### The A-Team

- *Biel, Jessica (Charisa Sosa)*
- *Cooper, Bradley (Lt. Templeton 'Faceman' Peck)*
- *Jackson, Quinton (Cpl. Bosco 'B.A.' Baracus)*
- *Neeson, Liam (Col. John 'Hannibal' Smith)*
- ...

### Toy Story 3

- *Allen, Tim (Buzz Lightyear)*
- *Cusack, Joan (Jessie)*
- *Hanks, Tom (Woody)*
- ...

### The Last Airbender

- *Patel, Dev (Prince Zuko)*
- *Peltz, Nicola (Katara)*
- *Ringer, Noah (Aang)*
- ...

## **Codifica (4 punti)**

Che differenza c'è tra ASCII e ISO 8859-1 (Latin 1)?

Di fronte a quale problema ci troviamo se un browser visualizza questa scritta: "MontrÃ©al"?

Su quale header HTTP bisogna agire per risolvere i problemi simili a quello appena illustrato? Come va modificato questo header?

Quali sono i vantaggi dell'uso di UTF-8 rispetto UTF-16? Quali sono gli svantaggi?

### ***Teoria (6 punti)***

1. Descrivere le caratteristiche principali e pregi/difetti del modello REST.
2. Soffermarsi sui ruoli dei diversi metodi HTTP in questo contesto.
3. Discutere la contrapposizione tra il modello REST e il modello basato su Web-Services.