

**Tecnologie Web**  
**C.d.L. in Informatica e**  
**Informatica per il Management**  
**Compito del 8 Febbraio 2016**

**Nome:**

**Cognome:**

**Matricola:**

**C.d.L.:**

**Team:**

**Corso seguito nell'A.A.:**

☐ Non è la prima volta. Data indicativa dell'ultimo appello provato: \_\_\_\_\_

☐ Ho già consegnato il progetto. Data indicativa: \_\_\_\_\_

**Importante:**

- Indicare ciascun esercizio risolto con una croce sul numero relativo. Saranno corretti solo gli esercizi così segnati.
- Se si consegna un foglio protocollo, scrivere in maniera chiara il numero di ogni esercizio presso la sua soluzione.
- Sul foglio protocollo, indicare inoltre il modo chiaro: nome, cognome e numero di matricola.

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

## 1. Domande di base (12 punti)

Rispondere correttamente a tre delle seguenti domande:

**A.** Elencare due header HTTP.

**B.** Quante righe e quante colonne ha la tabella generata dal seguente codice HTML?

```
<table>
  <tr>
    <td colspan="2">A</td>
    <td>B</td>
    <td rowspan="3">B</td>
  </tr>
  <tr>
    <td>C</td>
    <td colspan="2">D</td>
  </tr>
  <tr>
    <td>F</td>
    <td>G</td>
    <td>H</td>
  </tr>
</table>
```

**C.** Descrivere brevemente il seguente template XSLT:

```
<xsl:template select="treno">
  <train/>
</xsl:template>
```

**D.** Scrivere una query SPARQL a piacere.

## 2. HTML (6 punti)

Dato il codice HTML mostrato di seguito:

1. Scrivere il codice CSS per ottenere la visualizzazione mostrata in Figura 1. Si tenga presente che:
  - L'etichetta Mese rientra di 20 pixel, e le etichette Giorno e Orario rientrano di 50 pixel;
  - Il testo dell'intestazione della tabella è bianco e maiuscolo, e il colore di sfondo ha codice RGB 468284;
  - Il colore di sfondo delle righe della tabella relative ai campi 1 e 3 ha codice RGB 90C4DE, mentre il codice RGB delle altre è E6E6FA;
2. Scrivere il codice HTML e le regole CSS per ottenere la maschera di prenotazione mostrata in Figura 2.

Si tengano in considerazione questi vincoli:

- nessun elemento deve contenere l'attributo id tranne la tabella del testo dell'esercizio (di conseguenza non si può usare il selettore di id nel codice CSS);
- non è possibile aggiungere né fare riferimento ad altri attributi class tranne quelli presenti nel testo dell'esercizio (classi content, info e reservation);
- nessun elemento deve contenere l'attributo style.
- i contenuti sono 'statici' (i comportamenti dinamici sono gestiti nell'esercizio Javascript)
- dove non specificato, le dimensioni esatte di margini, padding e bordi non sono rilevanti

Sorgente HTML:

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <h1>Campi calcetto</h1>
    <div class="content">
      <div class="info"><h3>Scegli data e orario e verifica la disponibilità del campo</h3>
        <p>
          <label>Mese:</label><select name="mese">
            <option value="giugno">giugno</option>
            <option value="luglio">luglio</option>
            <option value="agosto">agosto</option>
          </select>
          <label>Giorno:</label><select name="giorno">
            <option value="24">24</option>
            <option value="25">25</option>
            <option value="26">26</option>
          </select>
          <label>Orario:</label><select name="giorno">
            <option value="20">20-21</option>
            <option value="21">21-22</option>
            <option value="22">22-23</option>
          </select>
        </p>
        <table id="timetable">
          <thead>
            <tr>
              <th>nome</th>
              <th>giocatori</th>
              <th>descrizione</th>
              <th>prezzo</th>
              <th>libero</th>
            </tr>
          </thead>
          <tbody>
            <tr>
              <td>campo 1</td>
              <td>8 giocatori</td>
              <td>all'aperto, illuminazione</td>
              <td>40 euro</td>
              <td></td>
            </tr>
            <tr>
              <td>campo 2</td>
              <td>10 giocatori</td>
              <td>coperto, aria condizionata, illuminazione, spogliatoi</td>
```

```

        <td>60 euro</td>
        <td></td>
    </tr>
    <tr>
        <td>campo 3</td>
        <td>10 giocatori</td>
        <td>coperto, aria condizionata, illuminazione, spogliatoi</td>
        <td>60 euro</td>
        <td></td>
    </tr>
    <tr>
        <td>campo 4</td>
        <td>14 giocatori</td>
        <td>all'aperto, illuminazione</td>
        <td>70 euro</td>
        <td></td>
    </tr>
</tbody>
</table>
</div>
<div class="reservation"></div>
</div>
</body>
</html>

```

Figura 1 - Resa della pagina in un browser:

## Campi calcetto

Scegli data e orario e verifica la disponibilità del campo

Mese: 
 Giorno: 
 Orario:

NOME	GIOCATORI	DESCRIZIONE	PREZZO	LIBERO
campo 1	8 giocatori	all'aperto, illuminazione	40 euro	✓
campo 2	10 giocatori	coperto, aria condizionata, illuminazione, spogliatoi	60 euro	✗
campo 3	10 giocatori	coperto, aria condizionata, illuminazione, spogliatoi	60 euro	✓
campo 4	14 giocatori	all'aperto, illuminazione	70 euro	✓

Figura 2 - Resa del form di prenotazione in un browser:

### Effettua una prenotazione

Scegli il campo:

- ☒ campo 1
- ☐ campo 3
- ☐ campo 4

Nome e cognome

Numero di telefono

commenti o richieste di informazioni

### 3. Javascript (6 punti)

Facendo riferimento al documento HTML dell'esercizio precedente, e utilizzando un framework a piacere tra JQuery, AngularJs o ExtJs, si eseguano i seguenti esercizi:

1. Ad ogni modifica di uno dei menù a tendina della pagina nella prima figura (evento *change*), viene invocato il servizio <http://www.polisportivaquestoequello.it/searchAvailability.py>, con parametri *month*, *day*, e *time* opportunamente popolati via form. Questo servizio ritorna un JSON della forma:

```
<tt>
{
    "query": {
        "month": 6,
        "day": 28,
        "time": 20
    },
    "fields": [
        {
            "name": "campo 1",
            "maxPlayers": 8,
            "open": true,
            "lights": true,
            "lockerRoom": false,
            "price": 40,
            "available": true
        }, ecc. ...
    ]
}
</tt>
```

In caso di risposta corretta, creare la tabella nella prima figura con i dati specificati nella risposta. Non si utilizzino stringhe di tag HTML nel javascript. E' possibile usare template HTML.

2. Facendo click sulla riga della tabella si renda visibile (ad esempio, dentro ad una finestra modale) il form della seconda figura opportunamente popolato: i campi non disponibili vengono nascosti (ma esistono nel DOM) e il campo su cui si è fatto click è selezionato automaticamente (ma è possibile selezionarne uno diverso). Cliccando sul pulsante "prenota" viene invocato il servizio <http://www.polisportivaquestoequello.it/bookField.py> con parametri opportuni, ma solo dopo aver verificato che i campi nome e cognome e numero di telefono sono popolati in maniera non banale e che il campo commenti non supera i 2000 caratteri. In caso contrario si generi un errore sensato e NON si attivi il servizio.
3. I framework citati non permettono di creare callback ad elementi del DOM non ancora esistenti (la riga della tabella di figura 1 non esiste ancora quando viene eseguita il `document.ready`). Discutere come sia possibile risolvere il problema e associare callback ad eventi su elementi del DOM che non esistono ancora.

#### 4. Semantic Web (6 punti)

Considerate la seguente descrizione in linguaggio naturale:

*Esistono due classi: una descrive i libri, mentre l'altra descrive le pagine. Inoltre, esistono due proprietà: una permette di associare un titolo ad un libro, mentre l'altra permette di dire che una pagina è inclusa in un libro.*

Rispondere alle seguenti domande:

- Utilizzando la sintassi Turtle, creare tutte le classi e proprietà discusse nella precedente descrizione.
- Scrivere una query SPARQL tale che, se una pagina è inclusa in due libri differenti, venga **costruita** una nuova tripla usando *owl:sameAs* come predicato per esprimere che i due libri sono, in realtà, lo stesso libro.
- Utilizzando la libreria Python RDFLIB e considerando un dataset RDF salvato nella variabile "my\_dataset" e conforme all'ontologia sviluppata, estrarre tutti i titoli di tutti i libri nel dataset e stamparli a video.

## **5. Teoria (4 punti)**

Rispondere alle seguenti domande:

- spiegare cosa è DOM, fornendo esempi di metodi
- discutere le differenze tra parser SAX e parser DOM