

Tecnologie Web

C.d.L. in Informatica e Informatica per il Management

Compito del 24 luglio 2013

Nome:

Cognome:

Matricola:

C.d.L.:

Team:

Corso seguito nell'A.A.:

Non è la prima volta. Data indicativa dell'ultimo appello provato: _____

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 e, in intestazione, indicare 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	

Domande di base (12 punti)

Rispondere correttamente a tre delle seguenti domande:

A. Spiegare brevemente se la seguente affermazione è vera o falsa: "REST è un protocollo che estende HTTP per garantire scalabilità e flessibilità."

B. Il seguente documento HTML è memorizzato su un server web. Quante richieste HTTP deve eseguire un browser per la sua visualizzazione completa, assumendo che le risorse dipendenti non abbiano dipendenze a loro volta?

```
<html>
<head>
  <link rel='stylesheet' href="style.css" type="text/css">
</head>
<body>
  <p>Content linking to a <a href="page.html">longer description</a>.</p>
  <p>And some images: 
  <canvas id="myCanvas" width="100" height="100"></canvas></p>
</body>
</html>
```

C. Qual è il colore assegnato al testo contenuto nell'elemento `<div class='it'><p>Lorem Ipsum</p></div>` dopo l'applicazione delle seguenti regole CSS?

```
p { color: yellow; }
div p { color: red; }
p.it { color: blue; }
```

D. Che risultato produce il seguente template XSLT applicato agli elementi "br" di una pagina HTML?

```
<xsl:template match="br">
  <xsl:element name="hr"/>
</xsl:template>
```

HTML (6 punti)

Scrivere il codice HTML 5 (e CSS) di un documento che rappresenti il contenuto visualizzato nell'immagine seguente:

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	✓

Effettua una prenotazione

Scegli il campo:

campo 1
 campo 2
 campo 3
 campo 4

Nome e cognome

Numero di telefono

commenti o richieste di informazioni

La figura precedente rappresenta un servizio che permette di prenotare campi da calcetto. La pagina è divisa in due parti: nella sezione in alto l'utente può consultare la disponibilità dei campi, nella parte inferiore è possibile effettuare una prenotazione.

Tale pagina deve avere le seguenti caratteristiche:

- per i valori dei campi di scelta di mese, giorno e orario è sufficiente mettere alcuni dei valori corrispondenti;
- il codice esadecimale del colore dello sfondo della riga di intestazione della tabella è 468284, quello della prima e della terza riga del corpo è 90C4DE, quello delle restanti è E6E6FA. Si noti anche che nella riga di intestazione il testo è di colore bianco;
- nell'ultima colonna della tabella, per i campi liberi viene utilizzata l'immagine "si.png", e per quelli occupati l'immagine "no.png". Entrambe le immagini sono contenute nella sottocartella "immagini";
- nella parte relativa alla prenotazione, si noti che l'opzione "campo 2" è disabilitata.
- nella parte relativa alla prenotazione, si noti l'allineamento dei campi e delle aree di testo e del testo "prenota".

Nello scrivere il codice si tengano in considerazione questi vincoli:

- nessun elemento deve contenere l'attributo id (di conseguenza non si può usare il relativo selettore nel codice CSS);
- nessun elemento deve contenere l'attributo style;
- l'elemento presentazionale b non può essere usato;
- Si richiede di scrivere tutto il codice CSS in un file separato, non in elementi <style>.

JavaScript (6 punti)

Si faccia riferimento alla figura dell'esercizio precedente, e utilizzando un framework Javascript a scelta tra JQuery e ExtJS, si implementino i seguenti servizi:

1. Al momento della visualizzazione della pagina, i selettori di mese, giorno e orario vengono automaticamente inizializzati alla data odierna e alla prossima ora (si ignorino i minuti). A tal fine, si ricordi che l'oggetto Date inizializzato senza parametri (e.g., new Date()) crea un oggetto di tipo Date del momento attuale, e che attraverso getMonth (restituisce un intero 0-11), getDay (restituisce un intero 1-31) e getHour (restituisce un intero 0-23) è possibile accedere ai valori corrispondenti.
2. Ad ogni selezione di data o orario, l'applicazione client-side si collega in GET asincrono al servizio <http://www.calchetto.it/campiDisponibili?date=DD/MM&hour=HH> dove MM è un mese (intero 1-12), DD è un giorno (intero 1-31) e HH è l'orario di inizio (00-23). In caso di richiesta corretta, il servizio restituirà un JSON come il seguente:

```
{
  "richiesta": { "data": "2012-07-24", "ora": "17"},
  "campo": [
    {"nome": "campo 1", "giocatori": 8, "caratteristiche": "AI", "prezzo": "40", "libero": true},
    {"nome": "campo 2", "giocatori": 10, "caratteristiche": "PCIS", "prezzo": "60", "libero": false},
    {"nome": "campo 3", "giocatori": 10, "caratteristiche": "PCIS", "prezzo": "60", "libero": true},
    {"nome": "campo 4", "giocatori": 14, "caratteristiche": "AI", "prezzo": "70", "libero": true}
  ]
}
```

dove A: Aperto, I: Illuminazione, P: Coperto (Pallone), S: Spogliatoi. I prezzi sono espressi in euro. Appena ricevuta la risposta corretta, l'applicazione client-side riempirà adeguatamente la tabella corrispondente (N.B.: il numero di righe della tabella è e deve rimanere variabile: non si assuma che ci sono esattamente quattro campi). In caso di richiesta scorretta viene visualizzato al posto della tabella un testo di errore significativo. Nella lista dei campi disponibili si disabilitino inoltre le voci relative ai campi occupati.

3. Ad ogni pressione del pulsante "Prenota", l'applicazione client-side si collega in POST asincrono al servizio <http://www.calchetto.it/prenotaCampo> spedendo un payload JSON come il seguente:

```
{
  "campo": "campo 1",
  "nome": "Mario Rossi",
  "telefono": "051 123456",
  "commenti": "Solo se c'è bel tempo",
  "data": "2012-07-24",
  "ora": "17"
}
```

Appena ricevuta la risposta corretta, l'applicazione visualizzerà un apposito messaggio di conferma. Si gestisca con un apposito messaggio di errore la spedizione di un valore nullo in un campo del form.

Semantic Web (6 punti)

Considerate le seguenti risorse e proprietà definite in Turtle mediante statement RDF (prefissi non dichiarati):

```
# La classe (o l'insieme) di tutte le person
foaf:Person a owl:Class .

# La classe (o l'insieme) di tutti gli articoli di conferenza
swc:Paper a owl:Class .

# La classe (o l'insieme) di tutte le organizzazioni (accademiche e non)
foaf:Organization a owl:Class .

# Collega gli articoli (soggetto dello statement RDF)
# alle persone (oggetto dello statement RDF) che li hanno creati
swrc:author a rdf:Property .

# Collega le persone (soggetto dello statement RDF)
# alle organizzazioni (oggetto dello statement RDF) a cui fanno parte
swrc:affiliation a rdf:Property .

# Permette di specificare il titolo di un articolo
swrc:title a rdf:Property .

# Permette di specificare il nome completo di una persona o di un'organizzazione
foaf:name a rdf:Property .
```

Rispondere alle seguenti domande:

1. In una sintassi a scelta tra RDF/XML e Turtle, tradurre la seguente frase in statement RDF, generando le opportune risorse ed usando **tutte** le classi e proprietà definite precedentemente:

Tom Heath ha scritto un articolo dal titolo "Linked Data - Welcome to the Data Network" e un'altro dal titolo "Revyu: Linking reviews and ratings into the Web of Data.". Le affiliazioni di Tom Heath per i due articoli sono rispettivamente Talis e Knowledge Media institute.
2. Considerando un grosso insieme di triple simili a quelle create al punto precedente, scrivere una query SPARQL che **restituisca** tutte le organizzazioni a cui Tom Heath è affiliato, senza ripetizioni e in ordine alfabetico.
3. Utilizzando la libreria *rdflib* di Python, scrivere un piccolo script che carichi il file *graph.ttl* contenente gli statement indicati all'inizio dell'esercizio e li salvi nel file *graph.xml* in formato RDF/XML.

Teoria (4 punti)

Rispondere ai seguenti quesiti su HTTP:

- spiegare le differenze tra "proxy", "gateway" e "tunnel"
- spiegare le caratteristiche principali dei meccanismi di *autenticazione*