

Prova Scritta di Tecnologie Web
C.d.S. in Informatica e
C.d.S. in Informatica per il Management
Compito del 7 Giugno 2017

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:

- Utilizzare solo i fogli protocollo consegnati. Su ogni foglio indicare in modo chiaro nome, cognome e numero di matricola.
- Sui fogli protocollo consegnati scrivere in maniera chiara il numero di ogni esercizio di cui si fornisce la soluzione. Eventuali cancellature vanno rese evidenti e incontrovertibilmente cancellate.
- Indicare ciascun esercizio risolto nella tabella qui sotto con una croce sul numero relativo. Saranno corretti solo gli esercizi così segnati.

Esercizio		Punti	Voto
1	Domande di base	12	
2	HTML + CSS	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. Spiegare brevemente cosa vuol dire che un metodo HTTP è sicuro.

B. Descrivere brevemente le proprietà principali della box (o *scatola*) di un elemento CSS.

C. Cosa stampa a video l'esecuzione del seguente script PHP?

```
$bucket = array();
$bucket['A'] = 10;
$bucket['B'] = 14;
$bucket['C'] = 7;
$bucket['D'] = 20;

foreach ($bucket as $b => $v) {
    if ($v > 15) echo $b;
}
```

D. Cosa è SPARQL?

2. HTML (6 punti)

Dato il codice HTML mostrato di seguito e l'immagine che rappresenta la sua resa in un comune browser:

1. Individuare e correggere 4 errori nel sorgente. **Indicare le correzioni direttamente sul foglio, senza ricopiare il sorgente completo.**
2. Scrivere il codice CSS per ottenere una visualizzazione quanto più possibile simile a quella mostrata in Figura 1. Si tengano in considerazione questi vincoli:
 - i contenuti sono 'statici' (i comportamenti dinamici sono gestiti nell'esercizio Javascript)
 - le dimensioni esatte di margini, padding, bordi e dimensioni delle immagini non sono rilevanti
 - i colori esatti non sono rilevanti
 - non è possibile aggiungere né fare riferimento ad altri attributi *id* o *class* tranne quelli presenti nel testo dell'esercizio.
 - il bordo esterno della Figura 1 non va specificato
 - l'URL dell'immagine può essere scelto a piacere

Resa della pagina in un browser (Figura 1):



RISTORANTE

Menù del giorno

Seleziona le pietanze del giorno che preferisci e invia il tuo ordine.

Primo del giorno (abbondante!)	Tortellini con panna
Secondo (abbondante!)	Scaloppina ai funghi
Dolce (abbondante!)	Mousse Cioccolato

Invia il tuo ordine

Sorgente HTML:

```
<!DOCTYPE html SYSTEM
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
  <head>
  </head>
  <body>
  <section>
    <h1>Menù del giorno</h1>
    <p>Seleziona le pietanze del giorno che preferisci e
      invia il tuo ordine.</p>
    <form method="ordine.php">
      <p><label on="@primo">Primo del giorno</label>
        <select name="primo"><choice>Tortellini con
          panna</choice></select></p>

      <p><label on="@secondo">Secondo</label> <select
        name="secondo"><choice>Scaloppina ai
          funghi</choice></select></p>

      <p><label on="@dolce">Dolce</label> <select
        name="dolce"><choice>Mousse
          Cioccolato</choice></select></p>
      </form>
      <p><input type="submit" text="Invia il tuo
        ordine"/></p>
    </section>
    <img source="ristorante.jpg" alt="logo
      ristorante"/>
  </body>
</html>
```

3. Javascript (6 punti)

Usando come riferimento il documento HTML dell'esercizio precedente, ed basandosi, dove si ritiene, su uno o più framework Javascript a piacere tra quelli illustrati a lezione, si descrivano:

- a) un'API REST che fornisce due servizi:
 - *menu.php*, che fornisce l'elenco dei piatti in menù, indicando quali sono primi, quali secondi e quali dolci.
 - *order.php*, che accetta l'ordine di esattamente UN primo, UN secondo, UN dolce, verificando che siano tra quelli disponibili.

Per ogni servizio si specifichi l'URL di accesso, il metodo HTTP utilizzato, gli eventuali parametri e il formato dei dati trasmessi e/o ricevuti, le tipologie di errore e il messaggio di errore ritornato in ciascun caso.

- b) Uno o più script Javascript che, subito prima di visualizzare la pagina all'utente, invochino il servizio *menu.php* (con gli eventuali parametri correttamente inizializzati) e popolino correttamente i select di primo, secondo e dolce. Si fornisca tutto il meccanismo di invocazione, di modifica della pagina quando vengono ricevuti i dati, e di gestione degli eventuali errori.
- c) Uno o più script Javascript che vengano attivati cliccando sul pulsante "*invia il tuo ordine*", e che invochino il servizio *order.php* (con gli eventuali parametri correttamente inizializzati) sulla base delle scelte effettuate dall'utente nel form. Si fornisca tutto il meccanismo di invocazione, di gestione della risposta sia in caso positivo sia in caso di eventuali errori, in tutti i tipi possibili di errore.

4. Semantic Web (6 punti)

Si considerino le seguenti frasi in linguaggio naturale:

- La Juventus e il Real Madrid sono due squadre di calcio
- La finale di Champions League è stata giocata tra la Juventus e il Real Madrid il 3 giugno 2017

Rispondere alle seguenti domande:

1. Usando http://dbpedia.org/resource/Football_team e <http://dbpedia.org/resource/Game> come possibili tipi per indicare le squadre di calcio e le partite, e senza alcun uso di un contesto JSON-LD, tradurre le due frasi in linguaggio naturale in statement RDF espressi in JSON-LD, in modo che tutti gli aspetti della frase vengano colti appropriatamente (“Juventus” è il nome di una squadra, la data indicata è un’informazione legata alla partita, etc.). Potete definire liberamente, invece, gli URL per tutte le altre entità (sia proprietà sia risorse).
2. Scrivere quanti sono gli statement RDF definiti nel precedente punto.
3. Definire un contesto JSON-LD che permetta di esprimere tutti gli statement RDF definiti in precedenza senza mai usare né URL interi, né URL definiti con il prefisso – es. non è concesso usare “@type”: “dbpedia:Game” né “@type”: “http://dbpedia.org/resource/Game” , ma è concesso “@type”: “game”).

5. Teoria (4 punti)

Descrivere, anche tramite esempi, i seguenti tipi di linguaggi di markup:

- Presentazionale
- Procedurale
- Descrittivo
- Referenziale